SONDER HEFT 1/87

64'er

12,— DM
Einzelpreis: 12,— sfr; 95,— ÖS;
11 500,— Lire; 41,— Fmk; 15,— hfl

TIPS - TRICKS - LISTINGS

TIPS & TRICKS

HILFESTELLUNG FEHLER SCHNELL ERKANNT

WICHTIG SPEICHERADRESSEN IM C16 UND PLUS/4

ESSENZ ROM-ROUTINEN UND NULLSEITE DES C128

PRAXIS

TRACKER 64 KONTRA KOPIERER

AT-TEXT SPART SCHREIBMASCHINE

DATEI-VERWALTUNG LEICHT GEMACHT

SPIELE

C64-SPIELEGENERATOR

DAS SCHWERT DER SAMURAI

WAHLKAMPF IN GRUSINIEN

• MEHR ALS 30 PROGRAMME FÜR C64, C128, C16/C116 UND PLUS/4 •

WWW.HOMECOMPUTERWORLD.COM

Das RUN Geschenk-Abo!



Macht Spaß, ist nützlic
und kommt 12 mal pro
Jahr ins Haus. Ein
Geschenk mit bestem
Erinnerungswert.

CW PUBLIKATIONEN

Ihre starke Verbindung zur Computerwe

Am einfachsten bestellen Sie mit diesem Coupon. Ausfüllen, abtrennen und ausreichend frankiert im Kuvert ab die Post an:

CW-Publikationen
Vertrieb RUN
Postfach 40 04 29
D-8000 München 40

RUN Geschenk-Abo Bestellcoupon

Ich verschenke RUN zum Bezugspreis von DM 68.— (12 Ausgaben), inkl. MwSt. und Lieferung „frei Haus". Auslandsendpreis: DM 78.—, für die Schweiz sfr 74.— Luftpostversand auf Anfrage.

Adresse des Bestellers (Rechnungsanschrift):

Name Vorname

Straße, Nr./Postfach

PLZ Wohnort

Datum X Unterschrift

Adresse des Beschenkten:

Name Vorname

Straße, Nr./Postfach

PLZ Wohnort

Laufzeit des Geschenkabonnements:

☐ Das Abonnement soll sich nach einem Jahr automatisch um weitere 12 Monate verlängern, wenn es nicht acht Wochen vor Ablauf gekündigt wird.

☐ Das Abonnement soll auf ein Jahr (12 Ausgaben) begrenzt sein.

Garantie:

Ich weiß, daß ich diese Vereinbarung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs. Ich bestätige dies durch meine zweite Unterschrift.

Datum X Unterschrift

Gewünschte Zahlungsweise:

☐ Bequem und bargeldlos durch Bankabbuchung

Bankleitzahl (vom Scheck abschreiben) Geldinstitut

Kontonummer/Inhaber

☐ Gegen Rechnung, zahlbar sofort nach Erhalt

WWW.HOMECOMPUTERWORLD.COM

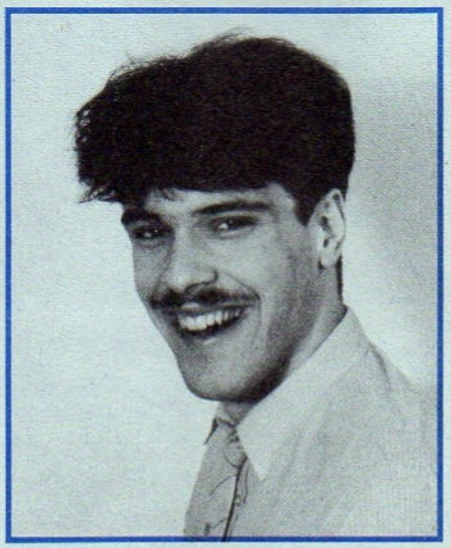
32 Listings - 23 Seiten Tips und Tricks

Er ist zu Ende: Unser großer Ideenwettbewerb „Gröprowaz II". 101 glückliche Gewinner erhalten Preise im Wert von 50 000 Mark. Die Auswahl zu treffen, welche Einsendungen einen der Preise erhalten, war schwierig. Neue und vor allem gute Ideen und Beiträge aktiver RUN-Leser heizten uns, von der Redaktion, kräftig ein. Täglich liefen zwei Kilometer Listings und Beschreibungen, entsprechend der doppelten Strecke zum Wirtshaus an der Ecke, durch unsere Hände. Spiele, Anwendungen und Utilities mußten unseren Anforderungen standhalten. Im ersten Durchgang wurde geprüft, ob die Programme fehlerfrei funktionieren. Fanden wir Errors in einem Beitrag, ging die Einsendung mit entsprechendem Vermerk (zum Beispiel: Syntax Error in 3450) an den Einsender zurück. Während dieses Testlaufs ordneten wir Programme nach Themen. Allein aus der Sparte Adreßverwaltungen traten 18 Programme den Wettstreit an. Ähnlich stapelten sich Beiträge unter den Bezeichnungen:

- *Vokabeltrainer,*
- *Mathetrainer,*
- *Basic-Erweiterungen,*
- *Sprite- und Zeichensatzeditoren*
- *sowie Musikprogramme.*

Im zweiten Durchgang drehten wir diese Beiträge erbarmungslos durch die Mangel. Sinn, Qualität, Aufmachung und Dokumentation müssen stimmen, damit ein Beitrag in der RUN veröffentlicht wird.

Nach diesem Test, uninteressante Beiträge eilten bereits den Autoren entgegen, waren die Brief- und Paketberge soweit abgebaut, daß man die Köpfe der zuständigen Redakteure wieder erkennen konnte. Alle verbliebenen und zum Veröffentlichen vorgesehenen Programme hatten ein Gesamtvolumen von sechs Waschkörben oder, auf einen Berg von zwei Metern Breite und drei Metern Länge gehäuft, eine Höhe von 85 Zentimetern. Zusammen ergab das 4,32mal den gelben Teil unseres Monats-Magazins. Da wir jedoch weder unseren Lesern noch unseren Autoren zumuten wollten, drei oder gar vier Monate zu warten, bevor diese Arbeiten zu lesen sind, mußte eine Extra-Publikation her. Aus diesem Grund befaßt sich unser erstes Sonderheft des Jahres 1987 auf 23 Seiten mit interessanten Tricks und Speicherstellen. 32 Listings reizen zum Abtippen. Und obwohl die meisten Programme nur ein bis zwei Seiten lang und damit in wenigen Minuten eingetippt sind, bieten wir alle Software dieses Heftes auf zwei Disketten an!



Gibt es Probleme beim Eintippen, stehen wir mit Rat und Tat zur Seite.

Dienstags, zwischen 16 und 18 Uhr, ist unser Telefon mit Rufnummer 0 89/3 60 86-2 28 besetzt. Dann geben wir Antworten und Lösungsvorschläge.

Gut Byte wünscht die Redaktion RUN

WWW.HOMECOMPUTERWORLD.COM

Inhalt

Checker für C64/16/Plus4 und C128
Tippfehler ohne Chance 6

Tips und Tricks

Errors unter der Lupe
Starke Medizin gegen Programmfehler 10

Was ist wo im 128er Basic-ROM?
Sprungadressen der Basic-Befehle 13

TEDdy im C16
So funktioniert der TExtDisplay-Chip 14

Im Kernal des C128
Wichtige Subroutinen des 128er-ROM 16

Z80-Programmiercodes im Überblick
Die Sprache des Zilog 80-Prozessors 20

Was ist wo in dem C16/116/Plus4?
Interessantes zwischen $100 und $1000 22

Von 0 auf 255: Nullseite des C16
So manipuliert man das Betriebssystem 28

SYS, PEEK und POKE im Commodore 64
Tricks und Kniffe mit dem 64er 31

Praxis-Listings

Geschwindigkeit ist Trumpf
Blitzschnell: „Power Basic" läßt alle stehen 33

Datasette als Datensafe
10mal schneller laden und speichern 40

Organisierte Datasetten: „Tape-Directory"
Inhaltsverzeichnisse von Datenkassetten erstellen 41

Dämonenhaft: DMON-Diskmonitor
Disketten auf die Bits geschaut 42

HELP hilft beim Fehler suchen
Simuliert den HELP-Befehl des C16/128 46

Floppy auf die Sprünge geholfen mit dem „Boot Sector Manager"
Anschalten-Laden-Starten 48

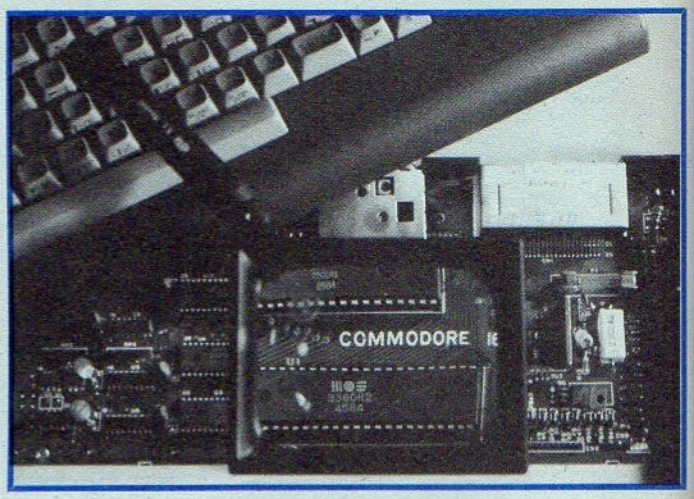
Auf den Track gekommen
Kopierprogramme austricksen 52

Diskette o.k.?
DIAN prüft Magnetscheiben 54

Y mal zwei
Grafiken auf der Folterbank 54

Adressen verwalten mit dem C64
Namen speichern, sortieren, drucken 56

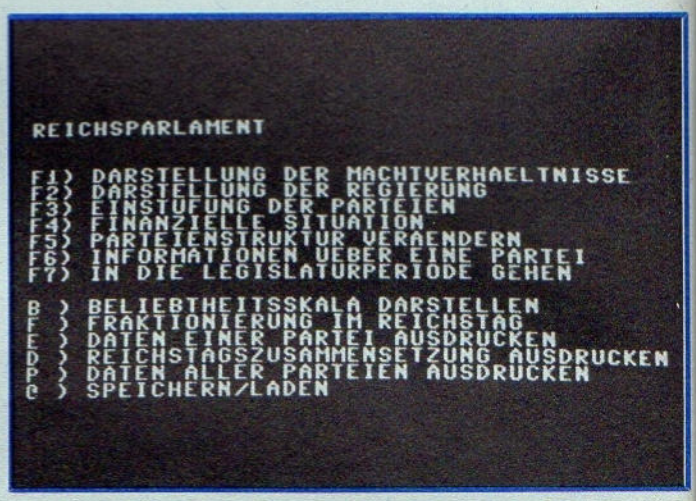
Schillernd Texten mit C16 und „AT-Text"
Tipp-Ex pensioniert 71



Jedes Bit genau durchcheckt

Auf 23 Seiten gibt es raffinierte Tricks; sind wichtige Speicherstellen der Commodore-Computer erläutert; stehen Antworten zu bislang ungeklärten Fragen. Wichtige PEEKs und POKEs unter der Lupe; ausführlich in ihrer Arbeitsweise und Aufgabe beschrieben, geben Einblick in die Betriebssysteme der C64, 128 und C16. **Das** *Nachschlagewerk bei kniffligen Fragen.*

ab Seite 10

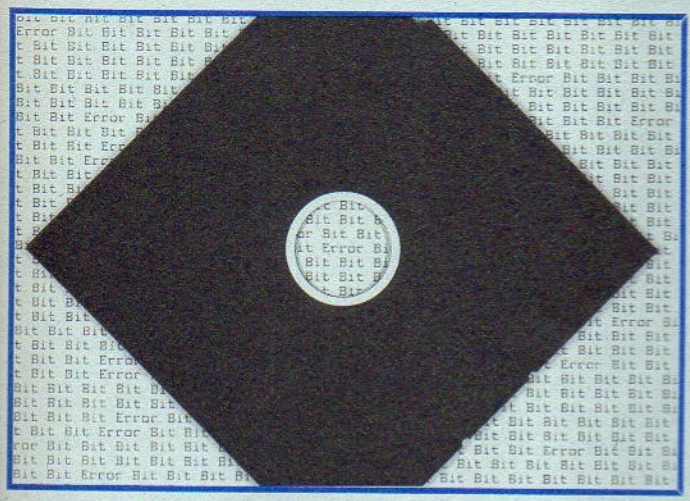


Die „Wende" in Grusinien oder: Gut geschmiert ist halb gewonnen

Bundeskanzler sein ist schwer. Das wissen unser Helmut und alle Leser, die das Spielelisting ab Seite 59 abtippen. Denn in diesem Programm geht es darum, mit eigenen Parteien die Macht im Land Grusinien an sich zu reißen und den obersten Posten, Reichskanzler, zu besetzen. Mißtrauensanträge, Schiebungen, Hetzkampagnen führen zum Erfolg.

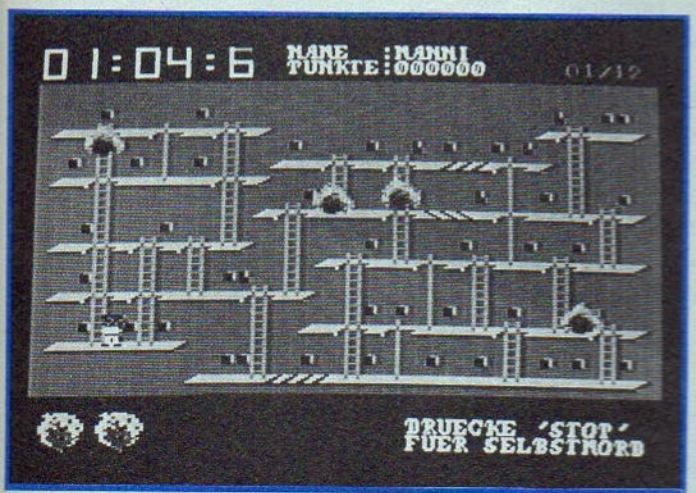
Seite 59

WWW.HOMECOMPUTERWORLD.COM



Im Gegensatz zu erfundenen Geistergeschichten ist der Disketten-Monitor DMON Wirklichkeit. Mit Hilfe seiner Befehle durchforstet man Directories, Programm- und Datenfiles. Diskettenschreib-/Lesefehler verlieren ihren Schrecken. Entwikkelt in Maschinensprache bearbeitet dieses Utility alle Aufgaben mit rasanter Geschwindigkeit. Dreieinhalb Seiten Listing
ab Seite 42

Dämonenhaft: DMON-Diskmonitor



Endlos neue Spielvariationen mit „Runman Construction Set" garantiert der eingebaute Spielegenerator. Jeder Spieler stellt seine eigenen Felder aus Treppen, Mauern, Bomben und Stangen zusammen. Schwierigkeitsstufen, Spielfiguren- und Gegnerzahlen sind frei definierbar. Ein komplett konstruiertes Spiel mit 8 Levels ist in diesem Super-Programm bereits enthalten.
Seite 88

Entwerfen und Spielen in einem: RCS

Guten Appetit: Commodore „verdaut" Karteikarten
Sortieren, Duplizieren, Editieren 75

Spielelistings

Die „Wende" in Grusinien
Gut geschmiert ist halb gewonnen 59

Krater sprengen ist nicht schwer ...
Auf der Flucht vor dem Imperium 79

Wasser im Commodore 64
„Schiffe versenken" mit dem Computer 81

Wer hat den richtigen Dreh: „Othello"
Denkspiel für C16/116/Plus4 85

Entwerfen und Spielen in einem: RCS
Spielegenerator für den 64er 88

„Explode" jagt den Commodore in die Luft
Basic-Kurs anhand eines Denkspieles 104

Die Erde als Abfalleimer
Müllbeseitigung der Zukunft 109

Bandwürmer im Computer
Operation „Wurmus": Bytes essen 111

Das Schwert der Samurai
Harte Karate-Kämpfe überstehen 112

Angriff auf Schlummerland
Flugzeugjagd mit einer Flak 124

Gimmy's entführen in NIM
Das bekannte Denkspiel gegen den C64 126

5 in eine Reihe
Intelligenter Algorythmus macht den C64 schlau 128

Umleitungen im Computer
Geschosse auf das Ziel leiten 130

C64 ist clever
Nur wenig besiegen den Computer in „Cleverness" 132

RUN an die Geschütze
2 Geschütze, 2 Spieler, 1 Sieger 134

Mit dem Kopf durch die Wand
„Break out" für den C16/116/Plus 4 136

FOG — Nebel des Grauens
Letzte Hoffnung: Rettung aus der Luft 138

Wer findet den Ausweg?
Dreidimensionale Labyrinthe durchqueren 139

Mit High-Speed über'n Bildschirm
Spielfeldpanik in 5 Levels 142

Sternbilder im Computer
Grafisch dargestellt: 30 Konstellationen 146

WWW.HOMECOMPUTERWORLD.COM

Eintippen ohne Fehler

In RUN werden zwei Arten von Programmen veröffentlicht: Basicprogramme und Maschinenprogramme.

Basicprogramme werden zur Vermeidung von Tippfehlern am besten mit dem **Korrektor** eingegeben. Nach jedem RETURN gibt er eine Prüfsumme aus, die mit der Prüfsumme im Listing übereinstimmen muß.

Die Basiclader zu den Korrektoren für den C64 und den C128 stehen auf Seite 7. Nach dem Einpoken der Daten aktivieren sie den Korrektor. Anschließend ist NEW einzugeben. Auf Wunsch speichern sie den fertigen Korrektor auch ab. Er kann dann mit LOAD "49152 KORREKTOR",8/NEW/SYS 49152 für den C64 und mit BLOAD "4864KORREKTOR128"/SYS 4864 für den C128 gestartet werden. Der C64-Korrektor arbeitet auch mit Simon's Basic. Dazu läßt man das erste Komma in Zeile 5000 weg. Simon's Basic muß vor dem Korrektor gestartet werden.
Der Basiclader des Korrektors für den C16/116/Plus/4 steht auf Seite 8. Er verfügt über keine eigene SAVE-Routine. Nach RUN kann man beim entsprechenden Hinweis den fertigen Korrektor abspeichern. Nochmaliges RUN startet das Programm.

Steuerzeichen sind bei den Basiclistings immer in Klartext übersetzt, so bedeutet [3LEFT], daß dreimal Cursor links gedrückt werden muß.

Maschinenprogramme können nicht direkt eingegeben werden. Dazu ist ein eigenes Programm nötig. Hier gibt es zwei gleichwertige Möglichkeiten: Basiclader oder Hexdump.

Ein **Basiclader** ist ein Basicprogramm, das ein Maschinenprogramm im Speicher unterbringt. Dazu werden die Zahlen, aus denen das Programm besteht, aus DATA-Zeilen ausgelesen und mit dem POKE-Befehl in den Speicher geschrieben.

Bei einem **Hexdump** liegen die gleichen Zahlen in hexadezimaler Form vor. Sie werden beim C64 mit einem speziellen Eingabeprogramm, dem **Checker** (Seite 9), im Speicher abgelegt.
RUN druckt Maschinenprogramme für den C64 als **kombinierte Hexdump-Basiclader** ab. Damit werden zwei Fliegen mit einer Klappe geschlagen: Dasselbe Listing kann einmal als Basiclader, einmal als Hexdump betrachtet werden. Der Leser kann sich so die Eingabemethode aussuchen, die ihm am meisten zusagt. Listing 1 zeigt ein solches Doppelprogramm. Es legt ein Beispielprogramm von 32 Byte Länge im Bereich ab 49152 ($C000) ab. Listing 2 und 3 zeigen, welche Teile vom Listing 1 jeweils wichtig sind, wenn man das Programm als Basiclader nutzt oder wenn man es als Hexdump verwendet.

Für den Basiclader ist alles von Interesse, außer den Angaben „C000" bis „C018" ganz links. Sie sind unbedingt wegzulassen! Übrig bleiben vier Basiczeilen mit den Nummern 10 bis 40, die Hexzahlen als DATAs enthalten. Abschließend ist noch der Teil mit den Nummern 60000—60090 in Listing 1 einzugeben. Er liest die DATAs und poket sie in den Speicher. Dieser Programmteil ist bis auf Zeile 60000 bei jedem Hexdump-Basiclader gleich. Er ist mit Prüfsummen für den Korrektor versehen. Bei den DATA-Zeilen wurde darauf verzichtet, da der Basiclader über ein eigenes Prüfsystem verfügt, das Eingabefehler weitgehend verhindert.

Bei Verwendung als Hexdump muß der Checker benutzt werden. Einzugeben sind dann nur die Hexzahlen (siehe Listing 3). Es fallen also die Basiczeilennummer, das Basicwort DATA dahinter und die Kommas zwischen den Hexzahlen weg. Die Hexadezimaladressen „C000" bis „C018" brauchen auch nicht eingetippt zu werden, da sie vom Checker geliefert werden. Sie sind nur zur Orientierung abgedruckt. Ebenso entfallen die Zeilen 60000—60090.

```
c000    10 data a9,00,8d,af,cf,8d,b0,cf, a9
c008    20 data 8d,b1,cf,8d,b4,cf,8d,b5, db
c010    30 data cf,85,5c,85,5d,a2,f8,8e, 46
c018    40 data b2,cf,08,68,8d,b3,cf,00, 2c

    60000 for a=49152 to 49176 step    4254
          8:s=0:c=0:for b=0 to 7:
          gosub 60040
    60010 s=s+h+1-c:c=s>255:s=255-s    5034
          and 255:poke a+b,h+1:next
          b:gosub 60040
    60020 f$="pruefsumme ok":if s<>    5699
          h+1 then f$="[rvs]
          pruefsummenfehler":f=f+1
    60030 gosub 60090:next a:print "   2267
          [down]"f"fehler":end
    60040 read h$:gosub 60050:h=1*     2715
          16:h$=mid$ (h$,2)
    60050 l=asc (h$+"x")-48:if l<0     3198
          or l>22 then 60080
    60060 if l<10 then return          1044
    60070 l=l-7:if l>9 then return     1334
    60080 f$="[rvs]eingabefehler":l=    2460
          0:f=f+1
    60090 print "zeile" peek (64)*     2726
          256+peek (63)f$:return
```

Listing 1: Beispiel für einen kombinierten Hexdump-Basiclader

```
10 data  a9,00,8d,af,cf,8d,b0,cf, a9
20 data  8d,b1,cf,8d,b4,cf,8d,b5, db
30 data  cf,85,5c,85,5d,a2,f8,8e, 46
40 data  b2,cf,08,68,8d,b3,cf,00, 2c
```

Listing 2: Beim Basiclader entfallen die Adressenangaben links

```
c000    a9 00 8d af cf 8d b0 cf  a9
c008    8d b1 cf 8d b4 cf 8d b5  db
c010    cf 85 5c 85 5d a2 f8 8e  46
c018    b2 cf 08 68 8d b3 cf 00  2c
```

Listing 3: Beim Hexdump fallen Zeilennummern, „Data" und Kommas weg

WWW.HOMECOMPUTERWORLD.COM

C64

```
 10 restore :for a=49152 to 49629:poke
    a,234:next :print "moment!
 20 f=0:read a:y=peek (63)+peek (64)*
    256:if a=0 then 200
 30 read b,c:for d=a to b:read a$:if
    len (a$)-2 then 90
 40 e=asc (a$)-48-(7 and a$>"@"):if a$<
    "0" or a$>"g" or a$>":" and a$<"a"
    then 90
 50 a$=mid$ (a$,2):e=e*16+asc (a$)-48-
    (7 and a$>"@")
 60 if a$<"0" or a$>"f" or a$>"9" and
    a$<"a" then 90
 70 f=f+e:poke d,e:next :if f=c then 20
 80 print "pruefsummenfehler: soll"c",
     ist"f:goto 100
 90 print "eingabefehler: "a$:y=0
100 z=peek (63)+peek (64)*256:a$=str$
    (y)+str$ (-z):if y=0 then a$=str$
    (z)
110 print "[down]ll"a$"[2up]";:poke
    198,1:poke 631,13:end
200 print "[down]fertig.":print "[down]
    soll das maschinenprogramm
     abgespeichert
210 input "[up]werden[2spaces]j[3left]"
    ;a$:if a$<>"j" then 300
220 sys 57812"@:49152 korrektor",8:poke
    193,0:poke 194,192:poke 174,222
230 poke 175,193:sys 62957:verify
    "49152 korrektor",8,1
300 a$="":input "[down]korrektor
     aktivieren? j[3left]";a$:if a$="j"
    then sys 49152
999 rem hauptprogramm
1000 data 49187,49208,2342,a9,ae,a2,c0,
     8d,02,03,8e,03,03,00,a0,1f,b9,e2
1001 data c1,99,00,04,88,10,f7
1002 data 49323,49387,9136,4c,31,ea,78,
     a9,2e,a2,c0,8d,14,03,8e,15,03,58
1003 data a9,00,8d,e1,c1,20,60,a5,8c,e1,
     c1,86,7a,84,7b,ac,e0,c1,f0,13,a0
1004 data 10,b9,e1,c1,99,f1,c1,a9,a0,99,
     e1,c1,88,d0,f2,8c,e0,c1,20,73,00
1005 data aa,f0,d3,88,84,3a,90,55
1006 data 49467,49629,19950,4c,96,a4,4c,
     08,af,20,6b,a9,20,79,a5,84,0b,a0
1007 data 00,84,08,84,45,84,46,84,09,c8,
     be,ff,01,f0,38,a5,08,30,0c,e0,20
1008 data f0,f2,e0,3a,f0,0c,e0,8f,f0,28,
     e0,22,d0,02,49,80,09,01,85,08,8a
1009 data a6,09,f0,06,0a,69,00,ca,d0,fa,
     18,65,45,85,45,90,02,e6,46,e6,09
1010 data a5,09,29,07,85,09,10,c2,46,08,
     90,21,a0,ff,8c,e0,c1,a2,14,20,b9
1011 data c1,c8,a9,bd,99,e2,c1,a9,be,c8,
     99,e2,c1,c8,a2,45,20,b9,c1,a9,bb
1012 data 99,e2,c1,4c,a4,a4,84,08,b5,00,
     85,63,b5,01,85,62,a2,90,38,20,49
1013 data bc,20,df,bd,a4,08,aa,c8,bd,00,
     01,f0,08,09,80,99,e2,c1,e8,d0,f2,60
1999 rem ein/ausschalter
2000 data 49152,49186,5131,ad,03,03,c9,
     c0,d0,09,ad,de,c1,ae,df,c1,4c,27
2001 data c0,8d,df,c1,ad,02,03,8d,de,c1,
     a9,a0,a2,1f,9d,e2,c1,ca,10,fa
2999 rem eingabesicherung
3000 data 49285,49322,4626,ad,e1,c1,d0,
     21,8d,20,d0,8d,21,d0,a5,da,30,0e
3001 data 09,80,85,da,a0,27,a9,20,99,28,
     04,88,10,fa,a6,d6,d0,05,a9,11,20,
     35,eb
4998 rem ende bzw. simon's anpassung
4999 rem duch weglassen des 1. kommas:
5000 data ,49293,49295,702,ea,ea,ea,
```

Korrektor für den C64, Disk Version

```
200 print "[down]fertig.":print "[down]
    soll das maschinenprogramm auf
     kassette
210 input "abgespeichert werden
    [2spaces]j[3left]";a$:if a$<>"j"
    then 300
220 sys 57812"49152 korrektor",1,1:poke
    193,0:poke 194,192:poke 174,222
230 poke 175,193:sys 62957:print "bitte
     die kassette zurueckspulen und
240 input "return druecken";a$:verify
    "49152 korrektor",1
```

Korrektor für den C64: Anpassungen für Datasette

C128

```
100 trap 640:bank 0:restore :print
    "moment...
110 data 1300,a9,00,8d,00,ff,ad,02,036
120 data 03,ae,03,03,c9,67,d0,0c,e0,8b5
130 data 13,d0,08,ad,35,14,ae,36,14,d96
140 data b0,14,8d,35,14,8e,36,14,a0,a89
150 data 27,a9,a0,99,37,14,88,10,fa,98e
160 data a9,67,a2,13,8d,02,03,8e,03,9b8
170 data 03,4c,53,fa,d8,20,24,c0,08,31a
180 data a0,27,b9,37,14,99,c0,07,88,d27
190 data 10,f7,a5,d7,30,17,a9,18,c5,9cc
200 data eb,f0,06,c5,e4,d0,02,c6,e4,388
210 data c5,e5,d0,02,c6,e5,a9,00,8d,aa9
220 data 61,03,28,4c,69,fa,78,a2,13,2f1
230 data 8e,15,03,a2,38,8e,14,03,58,7f7
240 data a2,ff,86,3c,20,93,4f,86,3d,d1a
250 data 84,3e,ad,37,14,c9,a0,f0,10,2d3
260 data a2,19,bd,37,14,9d,45,14,a9,315
270 data a0,9d,37,14,ca,10,f2,20,80,ee6
280 data 03,aa,f0,cc,90,03,4c,d9,4d,a50
290 data 20,a0,50,20,0a,43,84,0d,a0,014
300 data 00,84,09,84,0b,84,0c,84,0a,567
310 data 88,c8,20,c9,03,aa,f0,38,a5,b5a
320 data 09,30,0c,e0,20,f0,f1,e0,3a,9be
330 data f0,0c,e0,8f,f0,28,e0,22,d0,2b9
340 data 02,49,80,09,01,85,09,8a,a6,20e
350 data 0a,f0,06,0a,69,00,ca,d0,fa,625
360 data 18,65,0b,85,0b,90,02,e6,0c,d1b
370 data e6,0a,a5,0a,29,07,85,0a,10,bb4
380 data c1,46,09,90,17,a0,ff,a2,16,491
390 data 20,10,14,c8,a9,bd,99,37,14,965
400 data a2,0b,20,10,14,a9,bb,99,37,a6b
410 data 14,4c,ea,4d,84,09,b5,00,85,b79
420 data 65,b5,01,85,64,a2,90,38,20,e22
430 data 75,8c,20,44,8e,a4,09,aa,c8,ffb
440 data bd,00,01,f0,08,09,80,99,37,087
450 data 14,e8,d0,f2,60,4c4,37282
460 a=-1:for n=0 to 345:read a$:on len
    (a$) goto 470,490,510,530,550
470 print "data-fehler [rvs]"a$
480 print "zeile " peek (65)+peek (66)*
    256:end
490 b=dec (a$):poke a,b:p=p+b:a=a+1
500 k=xor ((k*2 and 4095)+int (k/2048),
    b):goto 570
510 if k=dec (a$) then k=0:goto 570
520 print "fehler: parity soll " mid$
    ( hex$ (k),2):goto 480
530 if a<0 then a=dec (a$):p=a:goto 570
540 print "pruefsumme fehlt!":end
550 if p<>val (a$) then print
    "pruefsummenfehler!":end
560 a=-1
570 next :if a>0 then 540
580 print "fertig.":print "[down]soll
     das maschinenprogramm
     abgespeichert
590 print "werden?":get key a$:if a$<>
    "j" then 610
600 bsave "4864korrektor128",b0,p4864
    to p5173:print ds$
610 print "[down]soll der korrektor
     gestartet":print "werden?
620 trap :get key a$:if a$="j" then sys
    4864
630 end
640 if er=14 then resume 480:else if er
    =30 then resume 660
650 print "eingabefehler: " err$ (er):
    help
660 trap :stop
```

Korrektor für den C128

WWW.HOMECOMPUTERWORLD.COM

```
100 trap 410:restore
110 data 0c,10,c1,07,9e,20,34,31,33,38
120 data 00,00,00,ad,e5,07,c9,18,d0,03
130 data ce,e5,07,ad,e8,07,c9,18,d0,03
140 data ce,e8,07,a9,00,8d,f1,07,4c,0e
150 data ce,a9,39,8d,02,03,a9,10,8d,03
160 data 03,a9,11,85,2c,60,a2,10,8e,15
170 data 03,a2,ff,86,3a,20,5a,88,86,3b
180 data 84,3c,ad,c0,0f,c9,a0,f0,10,a2
190 data 19,bd,c0,0f,9d,ce,0f,a9,a0,9d
200 data c0,0f,ca,10,f2,20,73,04,aa,f0
210 data d3,90,03,4c,25,87,20,3e,8e,20
220 data 53,89,84,0b,a0,00,84,08,84,45
230 data 84,46,84,09,88,c8,20,a5,04,aa
240 data f0,38,a5,08,30,0c,e0,20,f0,f1
250 data e0,3a,f0,0c,e0,8f,f0,28,e0,22
260 data d0,02,49,80,09,01,85,08,8a,a6
270 data 09,f0,06,0a,69,00,ca,d0,fa,18
280 data 65,45,85,45,90,02,e6,46,e6,09
290 data a5,09,29,07,85,09,10,c1,46,08
300 data 90,17,a0,ff,a2,14,20,db,10,c8
310 data a9,bd,99,c0,0f,a2,45,20,db,10
320 data a9,bb,99,c0,0f,4c,36,87,84,08
330 data b5,00,85,63,b5,01,85,62,a2,90
340 data 38,20,ce,a2,20,71,a4,a4,08,aa
350 data c8,bd,00,01,f0,08,09,80,99,c0
360 data 0f,e8,d0,f2,60,00,00,00
370 print "[clr]das programm sollte vor dem ersten
    startabgespeichert werden![down]
380 n=n+1:read a$:a=a+dec (a$):if len (a$)=2 then 380
390 print "data-fehler: [rvs]"a$:poke 1264,peek (63):
    poke 1265,peek (64):poke 1263,1
400 a=peek (65)+peek (66)*256-len (a$)-1:poke 1269,a
    and 255:poke 1270,a/256:help :end
410 if er=30 then print "break kann das programm
    zerstoeren!":end
420 if er=14 then resume 390
430 if er<>13 then print "zeilen-eingabefehler!":help
    :end
440 if n<>259 or a<>27332 then print
    "pruefsummenfehler, bitte pruefen!":end
450 restore :for a=4097 to 4354:read a$:poke a,dec
    (a$):next
460 print "das korrektorstarterprogramm kann jetzt
    abgespeichert werden.
470 poke 45,3:poke 46,17:clr
```

Korrektor für den C16/116/Plus/4

Checker für Maschinenprogramme

Der Checker dient zur Eingabe von Maschinenprogrammen für den C64. Er reduziert die Tipparbeit und verhindert Eingabefehler. Außerdem entfällt das langwierige Einlesen von Datazeilen.

Der Basiclader des Checkers steht auf Seite 9. Er poket nach dem Start mit RUN die Daten in den Speicher. Da er sich dabei selbst überschreibt, muß er unbedingt vorher abgespeichert werden! Nach dem Einlesen der Daten schreibt der Lader einen SAVE-Befehl auf den Bildschirm, so daß man nur Return zu drücken braucht, um den fertigen Checker abzuspeichern. Dieser kann dann mit LOAD "CHECKER",8 geladen und mit RUN gestartet werden. Will man den Checker mit der Datasette benutzen, muß man bei der Diskversion einige Zeilen ersetzen. Die nötigen Änderungen stehen im Listing unten.

Der Checker ist eine Art Texteditor für Maschinenprogramme, die in Form von Hexdumps abgedruckt sind. Jede einzelne Zeile (zu je acht Bytes) wird anhand einer miteinzugebenden Prüfsumme auf Richtigkeit getestet. Bei Tippfehlern ertönt ein akustisches Signal. Die hexadezimalen Daten werden mit den Tasten **0—9** und **A—F** eingegeben. Mit **RETURN** übernimmt man eine Zeile in den Speicher. Stimmt die vom Computer errechnete Prüfsumme mit der eingegebenen überein, ertönt ein hoher Ton. Bei einer fehlerhaften Zeile ist ein tiefer Ton zu hören, und der Cursor springt wieder an den Zeilenanfang.

Weitere Funktionen kommen folgenden Tasten zu:

Y: Muß vor dem Eintippen eines Programms als erstes betätigt werden! Der Checker fragt dann nach der Startadresse, unter der das Programm später abgespeichert werden soll. Der benötigte Wert ist jeweils den Programmbeschreibungen zu entnehmen.

Cursortasten: Der Cursor wird mit den Cursortasten gesteuert. Erreicht er den oberen oder unteren Bildschirmrand, wird entsprechend gescrollt.

INST/DEL: Löscht ein Zeichen oder fügt eines ein.

HOME: Setzt den Cursor an den Zeilenanfang.

CLR: Löscht eine Zeile und läßt sie zur Veränderung frei.

K: Löscht eine Zeile und zieht den Rest nach oben.

I: Fügt eine Leerzeile ein.

H: Setzt den Cursor an den Textanfang.

S: Speichert das Programm ab. Sollte sich im Text eine Leerstelle befinden, springt der Cursor dorthin. Saven ist erst möglich, wenn alle Leerstellen entfernt wurden.

L: Lädt ein abgespeichertes Programm, um es zum Beispiel zu ergänzen oder zu verändern.

P: Druckt das Programm aus.

Wird der Checker mit Runstop/Restore verlassen, so kann man ihn mit SYS 2073 ohne Datenverlust wieder aktivieren.

```
121 data 00,a0,02,20,bd,ff,68,86,93,66,9d,4c,5b,ff,ff  4592
127 data 2a,0f,99,00,04,c8,d0,a0,01,b1,b2,85,fd,c8,b1  3940
128 data b2,85,fe,38,4c,7d,f5,a9,00,20,fd,0e,ea,ea,20  4777
129 data f8,0f,20,d0,f7,20,17,f8,b0,58,20,af,f5,a5,b7  4369
130 data f0,07,20,ea,f7,90,09,b0,4a,20,2c,f7,f0,45,b0  3998
131 data 43,a5,90,29,10,d0,3d,85,c3,a9,40,85,c4,20,6a  4510
132 data 0f,b0,32,8a,69,07,aa,90,02,e6,af,86,1d,8a,29  4392
133 data 38,4a,4a,4a,a8,a5,af,38,e9,40,06,1d,2a,06,1d  4051
134 data 2a,85,1d,a2,00,e4,1d,f0,08,a9,ff,9d,00,3c,e8  3943
135 data d0,f4,b9,75,0d,9d,00,3c,4c,19,08,4c,92,09,30  4224
138 data c3,ff,4c,92,09,20,de,0d,b0,53,20,fd,0e,20,d0  3893
139 data f7,20,38,f8,b0,3b,20,8f,f6,a5,1b,85,ae,a5,1c  4410
140 data 38,e9,40,85,af,a5,fe,85,c2,a5,fd,85,c1,18,65  4675
141 data ae,85,ae,a5,af,65,c2,85,af,a9,01,20,6a,f7,b0  4004
142 data 13,a9,00,85,c1,a9,40,85,c2,a5,1c,85,af,a5,1b  4662
143 data 85,ae,20,67,f8,4c,19,08,20,08,85,1d,90,df,e6  4207
250 print "[clr,2down]save" chr$ (34)"checker" chr$     3209
    (34)"[home]";:end
```

Checker für den C64: Änderungen für Datasette

WWW.HOMECOMPUTERWORLD.COM

```
 1 data 0b,08,c1,07,9e,32,30,36,34,00,00,00,00,00,00 3850
 2 data a9,00,85,fd,85,fe,20,f8,0f,20,8a,09,a9,3f,8d 3822
 3 data 8a,02,a9,18,8d,05,dc,20,2f,08,20,a3,0a,4c,26 3590
 4 data 08,20,27,0a,a9,00,85,c6,a5,c6,f0,fc,20,27,0a 4017
 5 data ad,77,02,60,a9,00,8d,20,d0,a9,0e,8d,21,d0,a0 4143
 6 data 00,a2,d8,84,1b,84,1f,86,1c,86,20,a9,04,85,1e 4078
 7 data 84,1d,a9,00,91,1b,a9,a0,91,1d,c8,d0,f5,e6,1c 4525
 8 data e6,1e,e8,e0,dc,d0,ec,a2,17,a0,55,a9,06,91,1f 4396
 9 data c8,91,1f,c8,c8,c0,6c,90,f3,c8,c8,a9,05,91,1f 4432
10 data c8,91,1f,a5,1f,18,69,28,85,1f,90,02,e6,20,ca 4844
11 data d0,da,a0,05,a2,30,a9,b0,99,00,04,8a,09,80,99 4386
12 data 01,04,c8,c8,c8,e8,e0,38,d0,ed,a2,06,bd,ee,08 4595
13 data 99,00,04,c8,ca,d0,f6,60,a0,50,a9,04,84,1b,85 4237
14 data 1c,a9,17,85,02,a5,61,85,19,a5,62,85,1a,20,f5 4314
15 data 08,a5,19,18,69,08,85,19,90,02,e6,1a,a5,1b,18 4899
16 data 69,07,85,1b,90,02,e6,1c,c6,02,d0,e3,60,8d,95 4290
17 data 93,8b,88,83,a5,fd,18,65,19,48,a5,fe,65,1a,20 4634
18 data 66,09,68,20,66,09,20,83,09,a5,19,85,1d,a5,1a 4573
19 data 18,69,40,85,1e,a5,1a,85,1f,a5,19,0a,26,1f,0a 4295
20 data 26,1f,a9,3c,85,20,a5,19,4a,4a,4a,29,07,a8,b9 4282
21 data 98,09,a0,00,31,1f,f0,22,a0,00,84,0d,b1,1d,20 4096
22 data 66,09,20,83,09,a4,0d,c8,c0,08,d0,ef,20,2a,0b 3902
23 data 20,83,09,20,83,09,20,66,09,4c,65,09,a2,1c,a0 4452
24 data 00,a9,a0,91,1b,20,83,09,ca,d0,f4,60,48,4a,4a 4300
25 data 4a,4a,20,74,09,68,29,0f,4c,74,09,18,69,f6,90 4237
26 data 02,69,06,69,ba,29,bf,a0,00,91,1b,e6,1b,d0,02 4350
27 data e6,1c,60,a9,0f,8d,18,d4,20,30,0a,20,41,08,4c 4246
28 data bd,08,80,40,20,10,08,04,02,01,a9,50,a0,04,85 3956
29 data 1d,84,1e,a2,16,a0,27,84,20,a9,4f,85,1f,a4,1f 4304
30 data b1,1d,a4,20,91,1d,c6,1f,c6,20,10,f2,a5,1d,18 4375
31 data 69,28,85,1d,90,02,e6,1e,ca,d0,dc,a9,c0,85,1b 3722
32 data a9,07,85,1c,a5,61,18,69,b0,85,19,a5,62,69,00 4854
33 data 85,1a,4c,f5,08,a9,98,a0,07,85,1d,84,1e,a2,17 4543
34 data a9,27,85,1f,a9,4f,85,20,a4,1f,b1,1d,a4,20,91 4472
35 data 1d,c6,20,c6,1f,10,f2,a5,1d,38,e9,28,85,1d,b0 3834
36 data 02,c6,1e,ca,d0,dc,a9,50,85,1b,a9,04,85,1c,a5 4227
37 data 61,85,19,a5,62,85,1a,4c,f5,08,a4,63,b1,65,49 4858
38 data 05,91,65,60,a9,00,85,61,85,62,a9,00,85,63,85 4780
39 data 64,a9,55,85,65,a9,d8,85,66,60,c6,64,30,0c,a5 4656
40 data 65,38,e9,28,85,65,b0,02,c6,66,60,e6,64,a5,61 4274
41 data 38,e9,08,85,61,b0,0b,c6,62,10,07,a9,00,85,61 4236
42 data 85,62,60,4c,e6,09,e6,64,a5,64,c9,17,f0,0c,a5 4726
43 data 65,18,69,28,85,65,90,02,e6,66,60,c6,64,a5,61 4286
44 data 18,69,08,85,61,90,02,e6,62,a5,62,c9,3f,d0,0b 4477
45 data a5,61,c9,08,d0,05,a9,00,85,61,60,4c,a0,09,c9 4218
46 data 11,d0,03,4c,6e,0a,c9,91,d0,03,4c,45,0a,c9,1d 4396
47 data d0,03,4c,68,0b,c9,9d,d0,03,4c,3a,0b,c9,0d,d0 4362
48 data 03,4c,17,0c,c9,30,90,0f,c9,3a,90,08,c9,41,90 4538
49 data 07,c9,47,b0,03,4c,98,0b,c9,13,d0,03,4c,9b,0c 4505
50 data c9,93,d0,03,4c,95,0c,c9,14,d0,03,4c,e6,0c,c9 4925
51 data 94,d0,03,4c,bf,0c,c9,48,d0,06,20,30,0a,4c,bd 4095
52 data 08,c9,49,d0,03,4c,11,0d,c9,4b,d0,03,4c,85,0d 4351
53 data c9,53,d0,03,4c,0d,10,c9,4c,d0,03,4c,79,0f,c9 4657
54 data 59,d0,03,4c,68,10,c9,50,d0,03,4c,b2,10,60,a2 3757
55 data 08,a0,00,18,a9,00,71,1d,49,ff,c8,ca,d0,f8,60 4249
56 data c6,63,a5,63,10,03,e6,63,60,c9,02,f0,f3,c9,05 4668
57 data f0,ef,c9,08,f0,eb,c9,0b,f0,e7,c9,0e,f0,e3,c9 4988
58 data 11,f0,df,c9,14,f0,db,c9,19,d0,df,a9,16,85,63 4132
59 data 60,e6,63,a5,63,c9,1c,d0,03,c6,63,60,c9,02,f0 4487
60 data f1,c9,05,f0,ed,c9,08,f0,e9,c9,0b,f0,e5,c9,0e 4277
61 data f0,e1,c9,11,f0,dd,c9,14,f0,d9,c9,17,d0,af,a9 4728
62 data 1a,85,63,60,48,20,a0,0c,68,29,3f,09,80,a4,63 4225
63 data 91,1d,4c,68,0b,a0,00,b1,1b,29,7f,c9,30,b0,02 4250
64 data 69,3a,e9,30,4c,83,09,20,a8,0b,0a,0a,0a,0a,85 4663
65 data 0d,20,a8,0b,05,0d,60,a5,65,85,1b,a5,66,29,03 4437
66 data 09,04,85,1c,20,04,0c,20,e8,0b,b9,98,09,85,67 4313
67 data 49,ff,a0,00,31,1f,91,1f,60,a5,1e,38,e9,40,85 5117
68 data 1f,a5,1d,0a,26,1f,0a,26,1f,a9,3c,85,20,a5,1d 4184
69 data 4a,4a,4a,29,07,a8,60,a5,64,0a,0a,0a,18,65,61 4052
70 data 85,1d,a5,62,69,00,69,40,85,1e,60,20,c8,0b,a0 4544
71 data 00,84,02,20,b9,0b,20,83,09,a4,02,91,1d,c8,c0 4265
72 data 08,d0,ef,20,83,09,20,83,09,20,b9,0b,85,0d,20 4208
73 data 2a,0b,c5,0d,f0,06,20,61,0c,85,63,60,a5,67,a0 4036
74 data 00,11,1f,91,1f,a9,00,85,63,20,6e,0a,a9,a0,20 4156
75 data 6e,0c,a0,14,20,86,0c,4c,8f,0c,a9,3c,20,6e,0c 4032
76 data a0,50,20,86,0c,4c,8f,0c,8d,01,d4,a9,00,8d,00 4039
77 data d4,a9,11,8d,05,d4,a9,f1,8d,06,d4,a9,11,8d,04 4179
78 data d4,60,a2,00,ca,d0,fd,88,d0,f8,60,a9,00,8d,04 4094
79 data d4,60,20,c8,0b,20,bd,08,a9,00,85,63,60,a5,65 4092
80 data 85,1d,a5,66,29,03,09,04,85,1e,60,00,01,03,04 4516
81 data 06,07,09,0a,0c,0d,0f,10,12,13,15,16,1a,1b,20 4483
82 data a0,0c,a0,1b,b1,1d,c9,a0,d0,1b,a2,11,bc,ad,0c 4569
83 data c4,63,f0,0e,84,0d,ca,bc,ad,0c,b1,1d,a4,0d,91 3922
84 data 1d,d0,eb,a9,a0,91,1d,60,20,a0,0c,a5,63,f0,f8 4016
85 data a2,11,dd,ad,0c,f0,03,ca,10,f8,bc,ad,0c,b1,1d 4059
86 data ca,bc,ad,0c,91,1d,e8,e8,e0,12,d0,ef,a9,a0,a0 3655
87 data 1b,91,1d,4c,3a,0b,20,04,0c,a8,f0,85,1f,a9,7f 4069
88 data 85,20,a2,07,a9,0f,85,0d,8a,a8,b1,1f,a4,0d,91 3993
89 data 1f,c6,0d,ca,10,f3,a5,20,c5,1e,d0,06,a5,1f,c5 3850
90 data 1d,f0,0d,a5,1f,38,e9,08,85,1f,b0,d8,c6,20,d0 3808
91 data d4,20,e8,0b,a6,1f,b9,75,0d,85,0d,b9,7d,0d,a0 4347
92 data 00,31,1f,4a,08,85,02,b1,1f,25,0d,05,02,91,1f 4258
93 data e8,f0,08,28,7e,00,3c,08,4c,65,0d,20,bd,08,28 4453
94 data 60,00,80,c0,e0,f0,f8,fc,fe,ff,7f,3f,1f,0f,07 4399
95 data 03,01,20,04,0c,a0,07,a2,0f,84,0d,8a,a8,b1,1d 4110
96 data a4,0d,91,1d,ca,c6,0d,10,f3,a5,1d,18,69,08,85 4903
97 data 1d,90,02,e6,1e,a5,1d,c9,f8,d0,dc,a5,1e,c9,7f 4181
98 data d0,d6,20,04,0c,20,e8,0b,a2,ff,18,08,28,3e,00 4710
99 data 3c,08,ca,e4,1f,d0,f6,b9,75,0d,3d,00,3c,85,02 4452
100 data bd,00,3c,28,2a,39,7d,0d,05,02,9d,00,3c,4c,bd 4377
101 data 08,a2,00,bd,00,3c,c9,ff,d0,03,e8,d0,f6,86,02 4031
102 data a2,ff,e4,02,f0,3f,bd,00,3c,d0,05,ca,e0,ff,d0 3906
103 data f2,a6,02,bd,00,3c,85,02,a0,00,a5,02,39,98,09 4181
104 data f0,03,c8,10,f6,84,0d,a9,00,85,62,8a,0a,26,62 4749
105 data 0a,26,62,0a,26,62,05,0d,0a,26,62,0a,26,62,0a 3847
106 data 26,62,85,61,20,36,0a,38,60,a0,07,bd,00,3c,d9 3857
107 data 75,0d,f0,05,88,10,f8,30,bc,84,02,a9,00,85,1c 4023
108 data 8a,0a,26,1c,0a,26,1c,0a,26,1c,05,02,0a,26,1c 3836
109 data 0a,26,1c,0a,26,1c,85,1b,a5,1c,18,69,40,85,1c 4605
110 data 18,60,86,0d,85,1d,84,1e,a5,63,48,a5,64,48,a5 4253
111 data 65,48,a5,66,48,a0,27,a9,a0,99,00,04,88,10,f8 4569
112 data a0,00,b1,1d,f0,09,20,2a,0f,99,00,04,c8,d0,f3 4064
113 data a2,00,84,63,a9,06,99,00,d8,99,01,d8,c8,e8,e4 4167
114 data 0d,d0,f7,a9,d8,85,66,a9,00,85,64,85,65,85,20 4433
115 data 20,2f,08,48,29,60,d0,1a,68,c9,0d,f0,2d,c9,14 4204
116 data d0,ef,a5,20,f0,eb,c6,63,c6,20,a9,a0,a4,63,99 4769
117 data 00,04,10,de,a4,20,68,99,00,02,c4,0d,f0,d4,a4 4170
118 data 63,20,2a,0f,99,00,04,e6,20,e6,63,d0,c6,a4,20 4094
119 data a9,00,99,00,02,68,85,66,68,85,65,68,85,64,68 4809
120 data 85,63,60,48,a9,20,a0,0f,a2,0f,20,66,0e,98,a2 4109
121 data 00,a0,02,20,bd,ff,68,a8,a2,08,a9,08,20,ba,ff 4286
122 data 20,c0,ff,a9,01,4c,40,0f,46,49,4c,45,4e,41,4d 4114
123 data 45,20,00,48,4a,4a,4a,4a,4a,aa,68,29,1f,1d,38 4369
124 data 0f,60,00,a0,80,c0,40,e0,c0,e0,a9,08,20,b4,ff 3787
125 data a9,6f,20,96,ff,a0,27,a9,a0,99,00,04,a9,00,99 5164
126 data 00,d8,88,10,f3,a0,00,20,a5,ff,c9,0d,f0,09,20 4725
127 data 2a,0f,99,00,04,c8,d0,f0,ad,00,04,0d,01,04,29 3714
128 data 0f,48,20,ab,ff,68,60,a9,00,20,fd,0e,d0,6e,a2 3753
129 data 08,20,c6,ff,a9,00,85,90,85,1b,20,cf,ff,85,fd 3951
130 data 20,cf,ff,8d,fe,00,20,f8,0f,a9,3c,85,1c,a9,00 4487
131 data 85,02,85,1d,a9,40,85,1e,a5,90,d0,35,a0,00,20 4266
132 data cf,ff,91,1d,c8,c0,08,d0,f6,a0,00,a6,02,b1,1b 4079
133 data 1d,98,09,91,1b,e6,02,e8,e0,08,d0,06,a9,00,85 4160
134 data 02,e6,1b,a5,1d,18,69,08,85,1d,90,cf,e6,1e,a5 4369
135 data 1e,c9,80,d0,c7,20,cc,ff,a9,08,20,c3,ff,20,30 3883
136 data 0a,4c,92,09,a9,00,85,c6,a5,c6,f0,fc,d0,eb,a0 4216
137 data 00,98,99,00,3c,c8,d0,fa,60,20,cc,ff,a9,08,20 4361
138 data c3,ff,4c,92,09,20,de,0d,b0,53,a5,1b,48,a5,1c 4426
139 data 48,a9,01,20,fd,0e,85,1f,68,85,1c,68,85,1b,a5 4678
140 data 1f,d0,c5,a9,00,85,1d,a9,40,85,1e,a2,08,20,c9 4025
141 data ff,a5,fd,20,d2,ff,a5,fe,20,d2,ff,a5,1d,c5,1b 4263
142 data d0,06,a5,1e,c5,1c,f0,94,a0,00,b1,1d,20,d2,ff 4028
143 data c8,c0,08,d0,f6,a5,1d,18,69,08,85,1d,90,df,e6 4004
144 data 1e,d0,db,4c,bd,08,a2,04,a0,10,a9,88,20,66,0e 3781
145 data a9,0f,85,1b,a9,04,85,1c,20,b9,0b,8d,fe,00,20 4411
146 data b9,0b,8d,fd,00,4c,92,09,41,4e,46,41,4e,47,53 4239
147 data 41,44,52,45,53,53,45,20,00,18,69,f6,90,02,69 4622
148 data 06,69,3a,4c,d2,ff,48,4a,4a,4a,4a,20,98,10,68 4479
149 data 29,0f,4c,98,10,20,de,0d,b0,ae,a9,00,20,bd,ff 4373
150 data a9,04,aa,a0,00,20,ba,ff,20,c0,ff,a2,04,20,c9 3663
151 data ff,a0,06,a9,20,20,d2,ff,88,d0,f8,a2,07,a0,00 3591
152 data 98,20,a4,10,a9,20,20,d2,ff,c8,ca,10,f3,a0,05 4406
153 data b9,6c,11,20,d2,ff,88,10,f7,a9,0d,20,d2,ff,20 4961
154 data d2,ff,a5,fd,85,1f,a5,fe,85,20,a9,00,85,1d,a9 4394
155 data 40,85,1e,a5,1d,c5,1b,d0,06,a5,1e,c5,1c,f0,4e 4472
156 data a5,20,20,a4,10,a5,1f,20,a4,10,a9,20,20,d2,ff 4037
157 data 20,d2,ff,a0,00,b1,1d,20,a4,10,a9,20,20,d2,ff 4013
158 data c8,c0,08,d0,f1,a9,20,20,d2,ff,20,d2,ff,20,2a 4420
159 data 0b,20,a4,10,a9,0d,20,d2,ff,a5,1f,18,69,08,85 4333
160 data 1f,90,02,e6,20,a5,1d,18,69,08,85,1d,90,aa,e6 3638
161 data 1e,d0,a6,20,cc,ff,a9,04,4c,c3,ff,4d,55,53,4b 4150
162 data 48,43,0d,0d,0d,0d,0d,0d,0d,68,c1,00,00,00,00 3793
200 for z=1 to 162:f$=" ok":for b=0 to 14              2299
210 read h$:gosub 300:h=l*16:h$=mid$ (h$,2):gosub 300 3041
220 poke 2034+z*15+b,h+l                              1738
230 next :print "zeile"z;f$:next                      1716
240 poke 45,122:poke 46,17                            1639
250 print "[clr,2down]save" chr$ (34)"checker" chr$    3398
    (34)",8[home]";:end
300 l=asc (h$+"x")-48:if l<0 or l>22 then 330         3067
310 if l<10 then return                               1044
320 l=l-7:if l>9 then return                          1334
330 l=0:f$=" [rvs]eingabefehler":return               2647
```

Basiclader des Checkers für den C64, Diskversion

WWW.HOMECOMPUTERWORLD.COM

Errors unter der Lupe

Trifft der Commodore 16 auf einen Befehl, den er nicht versteht, gibt es einen Error. Die Nummer des Errors steht in Variable ER, der Text in String ER$(ER). Trat der Fehler in einem Programm auf, liest man aus Variable EL die Nummer der fehlerhaften Zeile. Um Errors zu verbessern, muß man wissen, was der C16 nicht verarbeiten konnte.

Benutzer des Commodore Basic V3.5 besitzen hervorragende Möglichkeiten, Fehler im Basic-Programm zu entdecken. Mit RUN behebt man Errors. Blitzschnell.

So funktioniert's!

Fehlernummer: 01
Fehlertext:
?Too Many Files-Error

Erklärung: Dieser Fehler tritt auf, wenn mehr als 10 Files zur gleichen Zeit geöffnet sind. Mindestens ein File ist mit CLOSE zu schließen. Wahrscheinlich durchläuft der C16 eine Endlosschleife, in der er nach dem zehnten Durchlauf, beim elften Versuch, ein File zu öffnen, mit dieser Fehlermeldung abbricht.

Fehlernummer: 02
Fehlertext: ?File Open Error

Erklärung: Es wurde versucht, ein bereits geöffnetes File nochmals zu öffnen. Entweder stehen zwei OPEN-Anweisungen, die sich auf dasselbe File beziehen, an verschiedenen Stellen im Programm, oder das Programm springt an die Stelle, an der das File eröffnet wurde, zurück.
(Siehe Fehler 1: ?Too Many Files).

Fehlernummer: 03
Fehlertext: ?File Not Open Error

Erklärung: Es wurde versucht, Daten aus einer Datei zu lesen oder in eine Datei zu schreiben, die vorher nicht mit OPEN geöffnet wurde. Entweder wurde der OPEN zu Beginn der Dateibehandlung vergessen oder man hat sich bei einer OPEN oder GET#/INPUT#/PRINT#-Anweisung bei der logischen Filenummer vertippt.

Fehlernummer: 04
Fehlertext: ?File Not Found Error

Erklärung: Es wurde versucht, eine Datei zu lesen, die nicht auf Kassette/Diskette ist. Der Grund liegt oftmals darin, daß man sich entweder beim Laden oder schon beim Speichern vertippt hat und der Name des Files auf dem Datenträger nicht dem eingegebenen Namen entspricht.

Fehlernummer: 05
Fehlertext:
?Device Not Present Error

Erklärung: An ein externes Gerät sollten Daten übermittelt oder davon eingelesen werden, obwohl dieses entweder abgeschaltet oder gar nicht angeschlossen ist. Tritt zum Beispiel auf, wenn man Programmme abspeichern oder laden will und die Diskettenstation ausgeschaltet ist. Läßt sich einfach durch Anschließen und Einschalten des gewünschten Gerätes beheben.

Fehlernummer: 06
Fehlertext: ?Not Input File Error

Erklärung: Öffnet man ein File mit dem Befehl OPEN var1,var2,var3 und möchte man aus dieser Datei lesen, so muß die Sekundäradresse var3 gleich 0 sein! Der Open Befehl ist entsprechend abzuändern, um diesen Fehler zu vermeiden (Beispiel: OPEN 1,2,0).

Fehlernummer: 07
Fehlertext:
?Not Output File Error

Erklärung: Öffnet man ein File mit dem Befehl OPEN var1,var2,var3 und möchte in diese Datei schreiben, so muß die Sekundäradresse var3 gleich 1 oder 2 sein! 2 schreibt bei Datasetten die Bandendemarkierung auf Kassette, wenn alle Daten abgespeichert sind. (OPEN 1,2,1/ OPEN 1,2,2).

Fehlernummer: 08
Fehlertext:
?Missing File Name Error

Erklärung: Es wurde versucht, ein File auf Diskette zu speichern oder von Diskette zu laden, ohne daß der Filename angegeben war. Bei Diskettenoperationen ist grundsätzlich der Filename mit anzugeben!

Fehlernummer: 09
Fehlertext:
?Illegal Device Number Error

Erklärung: Es wurde versucht, eine Gerätenummer anzusprechen, die für den angewandten Befehl falsch ist. So kann man nicht versuchen, mit 'LOAD„name",4' Daten vom Drucker zu laden. Die Device-Nummer ist in diesem Beispiel von 4 auf 8 zu ändern.

Bier ist weg Error

Fehlernummer: 10
Fehlertext:
?Next Without For-Error

Erklärung: Ein NEXT-Befehl wurde gefunden, ohne daß vorher eine FOR..TO..-Anweisung gegeben war. Oftmals tritt der Fehler auf, weil man sich in den Variablen geirrt hat. Beispiel:

```
10 For m=2 to 10
20 next n
```

Ergibt einen *?Next Without For Error* in Zeile 20. Basic wartet auf ein NEXT M, erhält aber ein NEXT N. Da keine For..TO..-Schleife mit N eröffnet ist, gibt es Fehler Nummer 10. Seltener ist, daß man mit einem GOTO oder GOSUB in ein Unterprogramm springt, daß in einem undefinierten NEXT endet. Beispiel:

```
10 GOTO 30
20 for i = 2 to 20
```


WWW.HOMECOMPUTERWORLD.COM

30 print i
40 next i
Ergibt *?Next Without For-Error in 40*, da Zeile 20 übersprungen und somit kein Schleifenanfang gegeben wird.

Fehlernummer: 11
Fehlertext: ?Syntax Error
Erklärung: Diese Fehlermeldung begleitet einen Basic-Programmierer auf Schritt und Tritt. Meist verursachen Tippfehler diesen Error. Manchmal ist er Ergebnis eines vergessenen Zeichens.
Beispiele:
10 print „hallo" a=a+1
„:" vergessen
10 print „hallo":a=a+1 RICHTIG
10 pint „hallo":a=a+1
Tippfehler

Fehlernummer: 12
Fehlertext:
?Return Without Gosub Error
Erklärung: Obwohl kein GOSUB-Sprung gemacht wurde, traf das Programm auf eine RETURN-Anweisung. Da es von nirgendwo herkam, wußte es nicht, wohin es zurückkehren sollte. Daher der Fehler. Meist liegt es daran, daß am Ende des Hauptprogrammes viele kleine Unterprogramme liegen. Wird das Basic-Programm nicht vor diesen Unterprogrammen mit GOTO oder STOP/ END abgefangen, arbeitet es eine Unterroutine ab, die mit RETURN endet — und schon ist der Fehler da.
Beispiel:

```
 10 input „zwei zahlen eingeben";
    a,b
 20 gosub 100
 30 ?„Danke"
100 if a=b then print „beide zahlen sind gleich": return
110 print „die beiden zahlen sind ungleich"
120 return
```

Dieses Programm schreibt zweimal dieselbe Aussage auf den Bildschirm bevor es mit *?Return Without Gosub-Error* abbricht.

Fehlernummer: 13
Fehlertext: ?Out Of Data Error
Erklärung: Mit dem Befehl READ wurde versucht, mehr Daten aus DATA's einzulesen, als vorhanden waren. Entweder war dies der Fall, weil man die DATA's erneut angesprochen hat: Dann hilft ein RESTORE, bevor man erneut Daten „READed". Meist hat man beim Eingeben der DATA-Zeilen eine oder mehrere Zahlen vergessen, so daß weniger Zahlen vorhanden sind, als versucht wird, einzulesen. Beispiel:
10 for i = 1to 10
20 read a: print a
30 next i
40 data 1,2,3,4,5,6,7,8,9
Fügen Sie in Zeile 40 ein Komma und die Zahl 10 an, gibt es keinen Fehler mehr! Der Fehler liegt daher nicht in Zeile 20 (wo vergeblich eingelesen wird), sondern in Zeile 40!

Fehlernummer: 14
Fehlertext:
?Illegal Quantity Error
Erklärung: Es wurde versucht, eine Basic-Funktion mit Werten zu nutzen, die diese nicht bearbeiten konnte. Beispiele dafür:
POKE 5000,500 oder POKE 5000,—1 (In Speicherstellen kann man nur Werte zwischen 0 und 255 schreiben!)
a$=„": print asc(a$)

C16 läuft über

Es ist schwierig, den ASCII-Wert eines Buchstabens zu ermitteln, den es nicht gibt . . . In solchen Fällen hilft:
print asc (a$+chr$(0))! print 15 AND 34000 (Es dürfen nur Zahlen zwischen —32768 und +32767 ANDiert werden!)
Es gibt noch viele weitere Beispiele, in denen dieser Error auftritt. Am besten ist es, die fehlerhafte Zeile zu listen und zu prüfen, ob eine Basic-Anweisung mit den zugewiesenen Variablen nicht zusammenarbeiten kann, weil diese außerhalb der Toleranz liegen. (Beispiel: print len (a$)).

Fehlernummer: 15
Fehlertext: ?Overflow Error
Erklärung: Der C16 kann keine Zahlen verarbeiten, die größer als 1.701411833E+38 oder kleiner als —1.701411833E+37 sind. Am besten prüft man die Formeln in der fehlerhaften Zeile neu durch.

Fehlernummer: 16
Fehlertext: Out Of Memory Error
Erklärung: tritt auf, wenn:
1.) Das Programm größer ist, als der freie Basic-Speicher. (Künstlich zu erzeugen, indem man das Basic-Speicherende herab oder den Basic-Speicheranfang herauf legt).
2.) Der Speicherplatz genügt nicht, um alle benötigten Variablen aufzunehmen. Hier hilft manchmal die FRE(0)-Funktion. Dieser Befehl löscht alle unbenutzten Variablen und schafft dadurch Speicherplatz.
3.) Mehr als 10 FOR. .TO-Schleifen wurden eröffnet, ohne daß die NEXT-Anweisung eine Schleife geschlossen hat.
5.) Zu viele Funktionen wurden mit DEF FN definiert.

Fehlernummer: 17
Fehlertext:
?Undef'd Statement Err
Erklärung: Eine Programmzeile sollte angesprungen werden, die nicht vorhanden war. Die Befehle:
RUN 100
GOTO 100
GOSUB 100
RESUME 100
verursachen einen *?Undef'd Statement Error*, wenn es im Programm keine Zeilennummer 100 gibt.

Fehlernummer: 18
Fehlertext: ?Bad Subscript Error
Erklärung: Es wurde versucht eine Index-Variable anzusprechen, die nicht definiert ist. Indexvariablen müssen bis zum 10. Feld (ab 0) nicht definiert werden, damit man fehlerfrei in sie schreibt oder daraus liest. Ab Feld 11 ist eine einmalige, vorherige DIMensionierung notwendig.
10 a(12)=15
führt zu *?Bad Subscript Error*;
10 dim a(50): a(12)=15
arbeitet ohne jedes Problem. Bei DIMensionierungen lernt man schnell Fehlernummer 19 kennen:

Fehlernummer: 19
Fehlertext:
?ReDIM'd Array Error
Erklärung: Es wurde zum zweiten Mal versucht, ein Indexfeld zu dimensionieren.

```
10 DIM a(30)
20 DIM a(55):
   rem Fehler: Doppel-DIM
```

Entweder müssen vorher alle Variablen mit CLR gelöscht werden, oder man unterläßt das zweite ▶

WWW.HOMECOMPUTERWORLD.COM

DIMensionieren. Aus diesem Grund ist es ratsam, DIM-Felder eher größer als notwendig zu dimensionieren, da sie nachträglich nicht erweitert werden können.

Fehlernummer: 20
Fehlertext: ?Division by Zero-Err
Erklärung: Mathematische Grundregel: Divisionen durch 0 sind nicht erlaubt. Bevor man durch eine Variable dividiert, muß man abfragen, ob diese 0 ist. Beispiel:

```
10 input „zahl eingeben";a
20 if a=0 then end
30 print „10 dividiert durch „;a;"
   ist „;a/10;" !"
```

Fehlernummer: 21
Fehlertext: ?Illegal Direct Error
Erklärung: Befehle wie INPUT und GET dürfen nicht im Direktmodus, also ohne Zeilennummer davor, eingegeben werden.
Beispiel:

```
10 input a: rem kein Fehler
```

„input a" ergibt den Fehler.

Fehlernummer: 22
Fehlertext:
?Type Mismatch Error
Erklärung: Die Unterschiede zwischen Strings und Zahlen wurden nicht berücksichtigt. Beispielsweise darf keinem String eine Zahlenvariable zugewiesen werden: a$=a.

Formular TOO Direct by Zero-Error???

Dafür gibt es Befehle:
a$=str$(a). Bei solchen Befehlen muß man sich genau an die Syntax halten und auf den kleinen Unterschied eines Dollarzeichens oder der Anführungsstriche achten. Beispiele:

```
a =chr$(15)   :rem Fehler
a$=asc(b$)    :rem Fehler
a =asc(15)    :rem Fehler
a=„hallo"     :rem Fehler
a$=24444      :rem Fehler
a$=„24444"    :rem Richtig!
```

Fehlernummer: 23
Fehlertext:
?String Too Long Error
Erklärung: Strings dürfen maximal 255 Zeichen lang sein. Beispiel:

```
10 a$=a$+„1"
20 a=a+1: print a
30 goto 10
```

Nachdem dieses Programm mit Fehlermeldung geendet hat, sollte man mal mit print a den Inhalt dieser Zählvariablen am Bildschirm ausgeben. Ist eine Basiczeile länger als 80 Zeichen, kommt es ebenfalls zu diesem Error. Dann muß man entweder die eine Zeile in zwei Zeilen teilen oder die Basic-Befehle-Kurzform benutzen.
? ist die Kurzform für print
i shift„N" für input
etc. . . . (siehe Handbuch)

Fehlernummer: 24
Fehlertext: ?File Data Error
Erklärung: Entweder sollten numerische Werte von Diskette/Kassette gelesen werden, obwohl dort Strings standen oder aber es sollten Strings von Diskette/Kassette eingelesen werden obwohl nur Zahlen vorhanden waren.

Fehlernummer: 25
Fehlertext:
?Formula Too Complex-Error
Erklärung: Eine Berechnung ist für den Computer zu kompliziert. Die Rechnung ist in Teilberechnungen aufzuspalten, die der Rechner bewältigt. (Unterprogramm mit GOSUB).

Fehlernummer: 26
Fehlertext:
?Can't Continue Error
Erklärung: CONT funktioniert, wenn

- — das Programm ohne Fehler abbrach.
- — in der Zwischenzeit kein Fehler aufgetreten ist
- — keine Basic-Zeile eingegeben oder verändert wurde
- — kein anderes Programm von Diskette oder Kassette geladen wurde.

Ansonsten gibt der C16 den Error aus.

Fehlernummer: 27
Fehlertext:
?Undef'd Function Error
Erklärung: Eine mathematische Funktion muß zuerst mit DEF FN definiert werden, bevor man mit FN darauf zugreifen kann. Ansonsten entsteht der *?Undef'd Function Error*. Häufig ist er die Folge eines Schreibfehlers.

Fehlernummer: 28
Fehlertext: ?Verify Error
Erklärung: Schreibt man Daten auf Diskette oder Kassette und prüft danach mit VERIFY, ob alles richtig abgespeichert wurde, kann dieser Fehler auftreten. Dann ist es ratsam, die Diskette/Kassette zu wechseln und einen neuen Datenträger einzulegen. Auf diesen speichert man die Daten nochmals. Ein erneuter VERIFY-Befehl sollte keinen Fehler zur Folge haben.

Fehlernummer: 29
Fehlertext: ?Load Error
Erklärung: Es war nicht möglich, die gewünschte Datei oder das gewünschte Programm von Diskette/Kassette einzulesen. Das liegt entweder daran, daß keine Diskette/Kassette eingelegt ist, die eingelegte Diskette/Kassette die geforderten Daten nicht enthält oder der Datenträger defekt ist. Defekte Datenträger prüft man mit Verify auf ihre Funktionen.

Der Lösung nahe

Fehlernummer: 30
Fehlertext: ?Break Error
Erklärung: Dieser „Fehler" ist kein Fehler. Er tritt nur auf, wenn man die STOP-Taste drückt, während ein Basic-Programm läuft oder wenn in diesem Programm der Befehl STOP steht. *?Break Error* kennzeichnet folglich einen künstlich herbeigeführten und gewünschten Fehler.

Fehlernummer: 31
Fehlertext: ?Can't Resume Error
Erklärung: Trifft der C16 auf eine RESUME-Anweisung, ohne daß ein Fehler auftrat, gibt er einen Fehler aus. In der Zeile vor dem RESUME sollte man einen GOTO auf die Zeile nach dem RESUME einfügen. Danach gibt es keinen *Can't Resume Error* mehr.

Fehlernummer: 777
Fehlertext: ?Out of RUN Error
Erklärung: Dieser Fehler tritt immer dann auf, wenn am Kiosk die RUN ausverkauft ist. Doch es gibt eine Lösung: In diesem Heft bis Seite 115 blättern, Abo-Karte rausnehmen, alles ausfüllen, 60-Pfennig-Briefmarke draufkleben und losschicken. Danach gibt es keine Probleme mehr mit diesem wirklich schwerwiegenden, grauenvollen Error.

WWW.HOMECOMPUTERWORLD.COM

Was ist wo im 128er Basic-ROM?

Viele Funktionen, die man in eigenen Maschinenprogrammen benötigt, gibt es bereits: Im Kernal- und Interpreter ROM des C128.

Basic 7.0 strotzt vor Befehlen. Dies ist für Basic- und Maschinenspracheprogrammierer vorteilhaft. In Basic nutzt man die gegebenen Anweisungen; in Maschinensprache die Unterroutinen der Basic-Befele im ROM. Dazu notwendig sind:
ROM-Sprungadressen der Befehle sowie die entsprechenden Basic-Tokens (beim C128 teilweise zweistellig).

Befehl	$ Token	dez	$ Adressen	dez
ABS	$ B6	/182	$ 85FE	/ 34302
APPEND	$ FE 0E	/014	$ 9ABB	/ 39611
ASC	$ C6	/198	$ 7FFD	/ 32765
AUTO	$ DC	/220	$ 585E	/ 22622
ATN	$ C1	/193	$ 8E2F	/ 36399
BACKUP	$ F6	/246	$ 9CDF	/ 40159
BANK	$ FE 02	/002	$ 68F8	/ 26872
BEGIN	$ FE 18	/024	$ 752A	/ 29994
BEND	$ FE 19	/025	$ 519C	/ 20892
BLOAD	$ FE 11	/017	$ 9B74	/ 39796
BOOT	$ FE 1B	/027	$ 704F	/ 28751
BOX	$ E1	/225	$ 60DD	/ 24797
BSAVE	$ FE 10	/016	$ 9B46	/ 39750
BUMP	$ CE 03	/003	$ 7D2F	/ 32015
CATALOG	$ FE 0C	/012	$ 9A05	/ 39429
CHAR	$ E0	/224	$ 65F3	/ 26099
CHR$	$ C7	/199	$ 7F45	/ 32581
CIRCLE	$ E2	/226	$ 64B4	/ 25780
CLOSE	$ A0	/160	$ 8B15	/ 35605
CLR	$ 9C	/156	$ 5105	/ 20741
CMD	$ 9D	/157	$ 542C	/ 21548
COLOR	$ E7	/231	$ 679D	/ 26525
COLLECT	$ F3	/243	$ 9C92	/ 40082
COLLISION	$ FE 17	/023	$ 6E7E	/ 28286
CONLAT	$ FE 13	/019	$ 9CC5	/ 40133
CONT	$ 9A	/154	$ 5954	/ 22868
COPY	$ F4	/244	$ 9CA9	/ 40105
COS	$ BE	/190	$ 8D85	/ 36229
DATA	$ 83	/131	$ 519C	/ 20892
DCLEAR	$ FE 15	/021	$ 9C85	/ 40069
DCLOSE	$ FE 0F	/015	$ 9AF0	/ 39664
DEC	$ D1	/209	$ 7A99	/ 31385
DEF	$ 96	/150	$ 7E80	/ 32384
DEL	$ D1	/209	$ 7A99	/ 31385
DELETE	$ F7	/247	$ 5CDA	/ 23770
DIM	$ 86	/134	$ 5764	/ 22372
DIRECTORY	$ EE	/238	$ 9A05	/ 39429
DLOAD	$ F0	/240	$ 9B25	/ 39717
DO	$ EB	/235	$ 5E21	/ 24097
DOPEN	$ FE 0C	/013	$ 9AA4	/ 39588
DRAW	$ E5	/229	$ 65BD	/ 26045
DSAVE	$ EF	/239	$ 9B0A	/ 39690
DVERIFY	$ FE 14	/020	$ 9B22	/ 39714
END	$ 80	/128	$ 4B1B	/ 19220
ENVELOPE	$ FE 0A	/010	$ 6DDB	/ 28123
ERR$	$ D3	/211	$ 7819	/ 30745
EXIT	$ ED	/237	$ 5E7B	/ 24187
EXP	$ BD	/189	$ 89AD	/ 35245
FILTER	$ FE 03	/003	$ 6D6B	/ 28011
FOR	$ 81	/129	$ 5C4C	/ 23628
FRE	$ B8	/184	$ 7A41	/ 31297
GET	$ A1	/161	$ 54FE	/ 21758
GOSUB	$ 8D	/141	$ 58B6	/ 22710
GOTO	$ 89	/137	$ 58C2	/ 22722
GRAPHIC	$ DE	/222	$ 68A5	/ 26789
GSHAPE	$ E3	/227	$ 63B3	/ 25523
HEADER	$ F1	/241	$ 9BAD	/ 39853
HELP	$ EA	/234	$ 586F	/ 22639
HEX$	$ D2	/210	$ 7B65	/ 31589
IF	$ 8B	/139	$ 51D2	/ 20946
INPUT	$ 85	/133	$ 554E	/ 21838
INPUT#	$ 84	/132	$ 5534	/ 21812
INSTR	$ D4	/212	$ 933D	/ 37693
INT	$ B5	/181	$ 8675	/ 34421
JOY	$ CF	/207	$ 7BE9	/ 31721
KEY	$ F9	/249	$ 5F48	/ 24392
LEFT$	$ C8	/200	$ 7F5C	/ 32604
LEN	$ C3	/195	$ 7AEE	/ 31470
LET	$ 88	/136	$ 52B2	/ 21122
LIST	$ 9B	/155	$ 4FEE	/ 20462
LOAD	$ 93	/147	$ 8AA8	/ 35493
LOCATE	$ E6	/230	$ 6723	/ 26403
LOG	$ BC	/188	$ 834D	/ 33613
LOOP	$ EC	/236	$ 5ED2	/ 24274
MID$	$ CA	/202	$ 7FA2	/ 32674
MONITOR	$ FA	/250	$ B000	/ 45056
MOVSPR	$ FE 06	/006	$ 69F5	/ 27125
NEW	$ A2	/162	$ 50E3	/ 20707
NEXT	$ 82	/130	$ 56E0	/ 22240
ON	$ 91	/145	$ 528F	/ 21135
OPEN	$ 9F	/159	$ 8B09	/ 35593

WWW.HOMECOMPUTERWORLD.COM

PAINT	$ DF	/223	$ 5FD1 / 24529
PEEK	$ C1	/194	$ 7AE8 / 31464
PEN	$ CE 04	/004	$ 7C8D / 31885
PLAY	$ FE 04	/004	$ 6B12 / 27410
POKE	$ 97	/151	$ 7B08 / 31496
POS	$ B9	/185	$ 7E56 / 32342
PRINT	$ 99	/153	$ 5446 / 21574
PRINT#	$ 98	/152	$ 5426 / 21542
PUDEF	$ DD	/221	$ 5D79 / 23939
RCLR	$ CD	/205	$ 7BBA / 31672
RDOT	$ D0	/208	$ 9488 / 38024
READ	$ 87	/135	$ 5595 / 21909
RECORD	$ FE 12	/018	$ 9C1D / 39965
REM	$ 8F	/143	$ 51AA / 20906
RENAME	$ F5	/245	$ 9CD1 / 40145
RENUMBER	$ F8	/248	$ 59E6 / 23014
RESTORE	$ 8C	/140	$ 59B8 / 22968
RESUME	$ D6	/214	$ 5DA4 / 23972
RETURN	$ 8E	/142	$ 516F / 20847
RGR	$ CC	/204	$ 7BA5 / 31653
RIGHT$	$ C9	/201	$ 7F90 / 32656
RND	$ BB	/187	$ 7DBA / 32186
RREG	$ FE 09	/009	$ 57A6 / 22438
RSPPOS	$ CE 05	/005	$ 7D2A / 32042
RSPRCOLOR	$ CE 07	/007	$ 7CF4 / 31988
RSPRITE	$ CE 06	/006	$ 7CB1 / 31921
RUN	$ 8A	/138	$ 5989 / 22921
RWINDOW	$ CE 09	/009	$ 7D8D / 32141
SAVE	$ 94	/148	$ 8A8E / 35470

SCALE	$ E9	/223	$ 672E / 26414
SCNCLR	$ E8	/232	$ 67E0 / 26592
SCRATCH	$ F2	/242	$ 9BE7 / 39911
SGN	$ B4	/180	$ 85DF / 34271
SIN	$ BF	/191	$ 8D8C / 36236
SLEEP	$ FE 0B	/011	$ 6906 / 26886
SOUND	$ DA	/218	$ 6F06 / 28422
STR$	$ C4	/196	$ 7F34 / 32564
SQR	$ BA	/186	$ 8931 / 35121
SPRCOLOR	$ FE 08	/008	$ 6EAA / 28330
SPRDEF	$ FE 1D	/029	$ 7084 / 28804
SPRITE	$ FE 07	/007	$ 697E / 27006
SSHAPE	$ E4	/228	$ 6251 / 25169
STOP	$ 90	/144	$ 4B12 / 19218
SYS	$ 9E	/158	$ 576E / 22382
TAN	$ C0	/192	$ 8DD5 / 36309
TEMPO	$ EF 05	/005	$ 6D02 / 27906
TRAP	$ D7	/215	$ 5D8F / 23951
TROFF	$ D9	/217	$ 57A0 / 22432
TRON	$ D8	/216	$ 579D / 22429
USR	$ B7	/183	$ 1218 / 04635
VAL	$ C5	/197	$ 7A6D / 31341
VERIFY	$ 95	/149	$ 8AA5 / 35496
VOL	$ DB	/219	$ 6EDF / 28383
WAIT	$ 92	/146	$ 695C / 26972
WIDTH	$ FE 1C	/028	$ 6ED0 / 28368
WINDOW	$ FE 1A	/026	$ 6FE6 / 28646
XOR	$ CE 08	/008	$ 7D74 / 32116

TEDdy im C16

Im C16 befindet sich ein neues, sehr leistungsfähiges Chip. Sein Name: TED (=TExt Display). Er verwaltet Speicher, Bildschirm, Sound, Tastatur und Joystick.

Um den C16 besser zu programmieren, ist man auf den Chip 7360 angewiesen. Er bearbeitet wichtige Funktionen. Manipuliert man seine Register (Basisadresse $FF00/65280), treten verblüffende Effekte auf. Damit unterstützt man zum Beispiel Window-Programmierung (ähnlich GEOS auf dem Commodore 64). Je nach Programm wird aus dem Commodore 16 sogar ein kleiner, programmierbarer Synthesizer.

Speicher	Funktion
$FF00/65280	Timer1— Low Byte
$FF01/65281	Timer1— High Byte
$FF02/65282	Timer2— Low Byte
$FF03/65283	Timer2— High Byte
$FF04/65284	Timer3— Low Byte
$FF05/65285	Timer3— High Byte
$FF06/65286	Hat mehrere Funktionen:
Bit 0—2	Beinhalten die vertikale Position des Bildschirminhalts
Bit 3	aus = Vertikal 24 Zeichen an = Vertikal 25 Zeichen
Bit 4	aus = Bildschirm aus an = Bildschirm an
Bit 5	aus = Bit-Map-Modus aus an = Bit-Map-Modus an
Bit 6	aus = Extended-Color-Modus aus an = Extended-Color-Modus an
Bit 7	„totes" Bit, immer 0
$FF07/65287	Hat mehrere Funktionen:
Bit 0—2	Beinhaltet die horizontale Pos. des Bildschirminhalts

WWW.HOMECOMPUTERWORLD.COM

Bit 3	aus = Horizontal 38 Zeichen an = Horizontal 40 Zeichen
Bit 4	aus = Multicolor-Modus aus an = Multicolor-Modus an
Bit 5	aus = Normale Funktion an = Horiz. Pos. „einfrieren"
Bit 6	aus = PAL -Norm an = NTSC-Norm
Bit 7	aus = Reverse (rvs) möglich an = Reverse abschalten
$FF08/65288	Tastatur
$FF09/65289	Hat mehrere Funktionen:
Bit 0	unbenutzt
Bit 1	Rasterinterruptflag
Bit 2	Light-Pen-Verarbeitung
Bit 3	Timer # 1 Unterlauf
Bit 4	Timer # 2 Unterlauf
Bit 5	unbenutzt
Bit 6	Timer # 3 Unterlauf
Bit 7	Interruptflag
$FF0A/65290	Hat mehrere Funktionen:
Bit 0	Achtes Bit aus $FF0B/65291
Bit 1	Rasterinterruptflag
Bit 2	Light-Pen-Verarbeitung
Bit 3	Timer # 1
Bit 4	Timer # 2
Bit 5	unbenutzt
Bit 6	Timer # 3
Bit 7	unbenutzt

Ist mindestens eines der Bits in Speicherstelle $FF0A/65290 gleich dem Bit in $FF09/65289, löst der TED einen Interrupt aus.

$FF0B/65291	Rasterzeilen /Bit 8 steht in $FF0A/65290
$FF0C/65292	Hardware Cursorposition
Bit 0	Bit 8 aus $FF0D/65293
Bit 1	Bit 9 aus $FF0D/65293
Bit 2—7	unbenutzt
$FF0D/65293	Hardware Cursorposition Die ersten 8 Bits. Überlauf nach $FF0C.
$FF0E/65294	Frequenz Stimme 1/ Bits 0 bis 7
$FF0F/65295	Frequenz Stimme 2/ Bits 0 bis 7
$FF10/65296	Frequenz Stimme 2
Bit 0	Bit 8 aus $FF0F/65295
Bit 1	Bit 9 aus $FF0F/65295
Bit 2—7	unbenutzt
$FF11/65297	Hat mehrere Funktionen:
Bit 0—3	Lautstärke (max. 15)
Bit 4	aus = Stimme 1 aus an = Stimme 1 an
Bit 5	aus = Stimme 2 aus an = Stimme 2 an
Bit 6	aus = Stimme 2: Rauschen aus an = Stimme 2: Rauschen an
Bit 7	aus = ohne Bedeutung an = Stimme 1 und 2: Einschaltzustand wiederherstellen
$FF12/65298	Hat mehrere Funktionen:
Bit 0	Frequenz Stimme 1 Bit 8
Bit 1	Frequenz Stimme 2 Bit 9
Bit 2	TED ROM/RAM Zugriff
Bit 3—5	Bit-Map-Basisadresse
Bit 6—7	Ohne Bedeutung
$FF13/65299	Hat mehrere Funktionen:
Bit 0	Hardwareuhr-Statusbit
Bit 1	Hardwareuhr setzen
Bit 2—7	Zeichensatz-Basisadresse
$FF14/65300	Farbspeicher:
Bit 0—2	Ohne Bedeutung
Bit 3—7	Farbspeicher-Basisadr.
$FF15/65301	Hintergrundfarbe 0:
Bit 0—3	Farbe
Bit 4—6	Helligkeit
Bit 7	unbenutzt
$FF16/65302	Hintergrundfarbe 1:
Bit 0—3	Farbe
Bit 4—6	Helligkeit
Bit 7	unbenutzt
$FF17/65303	Hintergrundfarbe 2:
Bit 0—3	Farbe
Bit 4—6	Helligkeit
Bit 7	unbenutzt
$FF18/65304	Hintergrundfarbe 3:
Bit 0—3	Farbe
Bit 4—6	Helligkeit
Bit 7	unbenutzt
$FF19/65305	Hintergrundfarbe 4:
Bit 0—3	Farbe
Bit 4—6	Helligkeit
Bit 7	unbenutzt
$FF1A/65306	Bit-Map-Wert
$FF1B/65307	Bit-Map-Wert
$FF1C/65308	Vertikale Position
Bit 0	Bit 8 aus $FF1D/65309
Bit 1—7	Ohne Bedeutung
$FF1D/65309	Vertikale Position Bits 0 bis 7
$FF1E/65310	Horizontale Position
$FF3E/65342	ROM Zugriff
$FF3F/65343	RAM Zugriff

Um Werte in ein bestimmtes Byte zu schreiben, benutzt man die Befehle AND und OR, je nachdem, ob man das Bit löschen oder setzen will. Die gePOKEten Werte dürfen nicht größer als 255 sein oder einen negativen Wert aufweisen. („Illegal Quantity Error" ist die Folge!)

Beispiel:

Um eine 25-Zeilen-Ausgabe zu erhalten, gibt man:

```
POKE 65286,PEEK(65286)OR 8
```

ein und setzt damit einzig und allein Bit 3 in diesem Register.

WWW.HOMECOMPUTERWORLD.COM

Im Kernal des C128

Buchstaben auf 40- oder 80-Zeichenbildschirmen ausgeben, Diskette booten, Tastaturwerte einlesen, und so weiter ... Für jede dieser Aufgaben gibt es im Kernal ROM eine Routine. Wer effizient programmieren will, nutzt diese ausgetesteten, platzsparenden „Subroutines" so oft als möglich.

Maschinenspracheunterprogramme des Kernal-ROM's steuern den Commodore 128. Nutzt man diese Routinen in Assembler, spart man sich viel Zeit, Arbeit und damit Energien.

Label: ACPTR
Adresse: $FFA5 / 65445
Register: A
Liest vom IEC-BUS ein Zeichen in den Akkumulator. Je nach Erfolg wird das Statusbyte in $90 gesetzt. Ist somit das Gegenstück zu CIOUT ($FFA8).

Label: BASIN
Adresse: $FFCF / 65487
Register: A
BASIN liest ein Zeichen aus der mit CHKIN definierten Datei oder der Tastatur in den Akkumulator.

Label: BOOTCALL
Adresse: $FF53 / 65363
Register: X
Bootet die eingelegte Diskette im Laufwerk mit Geräteadresse X-Register. Ist ein BOOT-File vorhanden, liest und startet es der C128, ansonsten erhält das Ursprungsprogramm die Kontrolle zurück.

Label: BSOUT
Adresse: $FFD2 / 65490
Register: A
Schreibt das im Akkumulator definierte Zeichen an das, mit CKOUT definierte Ausgabegerät. Ist das Ausgabegerät der Bildschirm, wird der Akkumulatorwert als ASCII-Zeichen angenommen und in den entsprechenden CBM-Code-Wert umgerechnet.

Label: BSOUT SCRN
Adresse: $C72D / 50989
Register: A
Gibt das im Akkumulator enthaltene Zeichen an der aktuellen Cursorposition des gerade aktiven Bildschirms aus. Der Vorteil gegenüber der „normalen" BSOUT-Printroutine: BSOUT SCRN ist schneller.

Label: C64MODE
Adresse: $FF4D / 65357
Register: keine
Diese Routine schaltet vom Commodore 128-Modus auf den C64-Modus um. Speicherplatz, Tastatur und Taktfrequenz werden dem C64-Modus angepaßt. Die Routine endet mit RESET. Somit endet die Kontrolle des Maschinenprogrammes über den Computer mit diesem JSR/JMP-Sprung.

Label: CHKIN
Adresse: $FFC6 / 65478
Register: X, Carry
Lenkt den Eingabekanal auf den vorher mit OPEN geöffneten Kanal um. Jedes weitere mit BASIN eingelesene Zeichen stammt aus dem neuen Kanal. Nach CLOS oder CLRCH ist automatisch die Tastatur wieder Eingabegerät. Tritt ein Fehler auf, so setzt CHKIN das Carry-Flag.

Label: CINIT
Adresse: $FF81 / 65409
Register: keine
Setzt den 40/80-Zeichenmodus entsprechend der Umschalttaste, initialisiert 40- und 80-Zeichencontroller sowie die Funktionstasten neu. Der Tastaturpuffer wird geleert und sämtliche Ein-/Ausgabekanäle werden geschlossen.

Label: CIOUT
Adresse: $FFA8 / 65448
Register: A
Sendet den Inhalt des Akkumulators an den IEC-BUS. Je nach Erfolg wird das Statusbyte in $90 gesetzt. Ist somit das Gegenstück zu ACPTR ($FFA5).

Label: CKOUT
Adresse: $FFC9 / 65481
Register: X, Carry
Schaltet vom bisherigen Ausgabekanal auf den vorher mit OPEN eröffneten, im X-Register angegebenen Kanal um. Sämtliche mit BSOUT ausgegebene Zeichen, werden an das definierte Ausgabegerät gesendet. Das Carry-Flag dient zur Fehlererkennung. Ist es gesetzt, trat ein Fehler auf.

Label: CLALL
Adresse: $FFE7 / 65511
Register: keine
Löscht sämtliche geöffneten Dateien, ohne diese zu schließen. Führt danach einen Sprung nach CLRCH durch.

Label: CLOSE
Adresse: FFC3 / 65475
Register: A, Carry
Schließt das File mit der im Akkumulator vorhandenen, logischen Filenummer. Sämtliche Daten dieser Datei werden aus der „OPEN FILE"-Tabelle gelöscht. Ist ein Fehler aufgetreten (zum Beispiel: zu schließendes File war nicht geöffnet), setzt CLOSE das Carry Flag (= 1).

Label: CLRCH
Adresse: $FFCC / 65484
Register: keine
Sämtliche Eingabe- und Ausgabeänderungen mit CHKIN und CKOUT werden rückgängig gemacht, ein UNTALK- und ein TALK-Befehl gesendet.

Label: CLRWIN
Adresse: $C142/49474
Register: keine
Löscht den Bildschirmbereich des aktuell definierten Fensters. Ist kein Fenster definiert, löscht CLRWIN den gesamten Bildschirm.

Label: CLQIR
Adresse: $C77D / 51069
Register: keine
Löscht sowohl den Hochkommaals auch die INSERT- und RVS-Modi des C128.

WWW.HOMECOMPUTERWORLD.COM

Label: CURHOM
Adresse: $C150 / 49752
Register: keine
Positioniert den Cursor in die linke, obere Ecke des aktuell definierten Fensters. Ist kein Fenster definiert, positioniert CURHOM den Cursor in die linke, obere Ecke des Bildschirms.

Label: DLCHR
Adresse: $FF62 / 65378
Register: keine
Kopiert die C128-Zeichensätze vom Character-Rom ins VDC (80 Zeichenkontroller) RAM. Je nach Stellung der ASCII/DIN-Taste handelt es sich um amerikanische oder um deutsche Zeichen.

Label: DMACALL
Adresse: $FF50 / 65360
Register: X
Ermöglicht DMA-Zugriffe. Im X-Register steht der Wert der neu einzuschaltenden Konfiguration.

Label: END
Adresse: $4DC6 / 19910
Register: keine
Schaltet den Commodore 128 vom Programmodus auf Direktmodus um. Beendet ein Basic-Programm ohne END oder BRK.

Label: GETCONF
Adresse: $FF6B / 65387
Register: X
Holt das dem X-Register entsprechende Konfigurationsbyte aus der Tabelle ab Adresse $F7F0. Der neue Wert steht im Akkumulator. Der Wert des X-Registers darf 16 nicht übersteigen.

Label: GETIN
Adresse: $FFE4 / 65508
Register: A
Liest ein Zeichen aus der definierten Eingabedatei (Tastatur oder RS232) in den Akkumulator.

Label: GETLIN
Adresse: $C258 / 49752
Register: keine
Diese Routine nimmt so lange Zeichen von der Tastatur entgegen und printed diese am Bildschirm, bis CR = RETURN betätigt wird.

Label: INDCMP
Adresse: $FF7A / 65402
Register: A,X,Y
Speicher: $02C8
Vergleicht den Wert einer beliebigen Speicherstelle mit dem Akkumulator. Die Register müssen genauso gesetzt werden, wie in INDSTA, mit der Ausnahme, daß Speicherstelle $02C8 auf den Nullseitenpointer zeigt. Das Ergebnis des Vergleiches steht in Speicherstelle $05.

Label: INDFET
Adresse: $FF74 / 65396
Register: A,X,Y
Liest aus einer beliebigen Speicherbank einen Wert. In der Nullseite muß der Pointer der Adresse stehen, die ausgelesen wird, der Akkumulator zeigt auf diese Nullseitenadresse, im X-Register steht der Konfigurationsindex und im Y-Register der Offset des Nullseiten-Pointers.

Label: INDSTA
Adresse: $FF77 / 65399
Register: A,X,Y
Speicher: $02B9
Im Gegensatz zur INDFET-Routine schreibt INDSTA den Wert des Akkumulators in eine beliebige Speicherstelle. X- und Y-Register haben dieselben Aufgaben; in Adresse $02B9 steht der Zeiger auf die Zero-Page-Adresse, die auf den gewünschten Speicher zeigt.

Label: IOBASE
Adresse: $FFF3 / 65523
Register: X,Y
Holt die Adresse des Ein-/Ausgabebereiches in die Register: X = Low-Byte, Y = High-Byte. Der Wert entspricht $D000 = Y: D0, X = 00.

Label: IOINIT
Adresse: $FF84 / 65412
Register: keine
Initialisiert sämtliche I/O-Leitungen. Drucker, Laufwerk, Modem etc. werden in den Anschaltzustand gesetzt.

Label: JMPFAR
Adresse: $FF71 / 65393
Register: keine
Speicher: $02 bis $09
Führt einen Sprung in beliebigen Speicher durch und führt ein dort vorhandenes Maschinenprogramm weiter. Die Parameter in Adressen $02 bis $09 gleichen den Parametern in JSRFAR.

Label: JSRFAR
Adresse: $FF6E / 65390
Register: keine
Speicher: $02 bis $09
Springt eine beliebige Speicherbank, unabhängig von der gerade eingeschalteten Konfiguration, an. Es steht in:
$02 — Konfigurationsindex
$03/$04 — Programmzähler (PC)
$05 — Prozessorstatus
$06 — Akkumulator
$07 — X-Register
$08 — Y-Register
$09 — Steckpointer (SP)
Wichtig: Bereich $C000 bis $FFFF (Kernal-ROM) muß eingeblendet sein, da das System ansonsten unkontrollierbare Funktionen ausführt.

Label: KEY
Adresse: $FF9F / 65439
Register: keine
Prüft die Tastatur ab. Ist eine Taste gedrückt, wird deren ASCII/DIN-Wert (je nach ASCII/DIN-Modus) ermittelt und in den Tastaturpuffer geschrieben und erhöht ihn um eins.

Label: LISTN
Adresse: $FFB1 / 65457
Register: A
Stellt das angesprochene, externe Gerät durch LISTEN auf Empfang. Im Akkumulator steht die Geräteadresse. Die Zeichen müssen über CIOUT ausgegeben werden, UNLISTEN schaltet das angesprochene Gerät ab.

Label: LKUPLA
Adresse: $FF59 / 65369
Register: A/ A,X,Y, Carry
Speicher: $90, $B8—$BA
Sucht nach der im Akkumulator enthaltenen logischen Filenummer. Ist das Carry gelöscht, stehen in:
Akkumulator: logische Filenummer
X-Register : Geräteadresse
Y-Register : Sekundäradresse
Ist das Carry-Flag gesetzt, war die gesuchte logische Filenummer nicht mit OPEN geöffnet.

Label: LKUPSA
Adresse: $FF5C / 65372
Register: Y/ A,X,Y,Carry
Speicher: $90, $B8—$BA
Sucht nach der im Y-Register enthaltenen Sekundäradresse. Ist das Carry-Flag gelöscht, steht im Akkumulator die logische Filenummer und in X die Gerätenummer. Ist das Carry gesetzt, wurde die Geräteadresse im Y-Register ▶

WWW.HOMECOMPUTERWORLD.COM

unter den geöffneten Kanälen nicht gefunden.

Label: LOADSP
Adresse: $FFD5 / 65493
Register: X,Y
Das mit SETNAM, SETLFS und SETBNK definierte File wird durch LOADSP an die Adresse: X-Register = Low-Byte, Y-Register = High-Byte geladen.

Label: MEMBOT
Adresse: $FF9C / 65436
Register: X,Y, Carry
Führt dieselbe Funktion aus wie MEMTOP, nur das sich die Register auf die Speicheruntergrenze anstatt der Speicherobergrenze beziehen.

Label: MEMTOP
Adresse: $FF99 / 65433
Register: X,Y, Carry
Ist das Carry-Flag gelöscht (=0), so übergibt diese Routine dem Register X den Low-Wert und dem Register Y den High-Wert der Speicherobergrenze. Ist das Carry-Flag gesetzt (=1), so schreibt diese Routine aus den Registern X den Low- und aus Y den High-Wert der Speicherobergrenze.

Label: OPEN
Adresse: $FFC0 / 65472
Register: keine
Öffnet die mit SETNAM, SETLFS und SETBNK definierte Datei. Erst nach diesem OPEN greifen CKOUT und CHKIN auf diese Datei zu.

Label: PFKEY
Adresse: $FF65 / 65381
Register: A,X,Y
Belegt die Funktionstasten F1—F10 sowie Shift-RUN/STOP und HELP mit beliebigen Zeichenketten. Im Akku steht die Zero-Page-Adresse, die den Pointer auf dem Funktionstastentext beinhaltet. Im X-Register steht die Funktionstastennummer (1 bis 12) und im Y-Register die Länge der Zeichenkette.

Label: PHOENIX
Adresse: $FF56 / 65366
Register: keine
Führt einen RESET durch. Der Computer startet, als ob er gerade neu eingeschaltet wurde. PHOENIX springt automatisch BOOTCALL an. Die Kontrolle über den Rechner wird an das Bootsystem übergeben.

Label: PLOT
Adresse: $FFF0 / 65520
Register: X,Y, Carry
Ist das Carry gesetzt (=1), steht im X-Register: Aktuelle Cursorspalte
Y-Register: Aktuelle Cursorzeile
Ist das Carry gelöscht (=0), wird der Cursor an die entsprechende X/Y-Position (siehe oben) gesetzt.

Label: PRIMM
Adresse: $FF7D / 65405
Register: keine
Sämtliche Zeichen, die im Speicher nach diesem Aufruf stehen, werden so lange als Text auf dem aktuellen Ausgabegerät ausgegeben, bis PRIMM ein „Null-Byte" ($00) findet.

Label: RAMTAS
Adresse: $FF87 / 65415
Register: keine
Software-RESET. Setzt Zero-Page, RS232- und Kassettenpuffer zurück.

Label: READST
Adresse: $FFB7 / 65463
Register: A
Nachdem READST angesprungen wurde, steht im Akkumulator das Statusbyte.

Label: RDTIM
Adresse: $FFE1 / 65502
Register: A,X,Y
Liest die Systemuhr aus. Im Akkumulator steht der Stundenwert, im X-Register steht der Minutenwert und im Y-Register die Sekunden.

Label: RESET
Adresse: $E000 / 57344
Register: keine
Versetzt den Commodore in den Einschaltmodus zurück. Die Kontrolle erhält das Betriebssystem.

Label: RESTOR
Adresse: $FF8A / 65418
Register: keine
Initialisiert sämtliche Systemvektoren in Adressen $0314 bis $0332. Danach springt RESTOR die Unterroutine VECTOR mit gelöschtem Carry an.

Label: SAVESP
Adresse: $FFD8 / 65496
Register: A,X,Y
Speichert einen beliebigen Speicherbereich unter den mit SETNAM, SETBNK und SETLFS definierten Bedingungen auf Kassette oder Diskette ab. Die Anfangsadresse des abzuspeichernden Speicherbereichs muß in eine beliebige Zero-Page-Adresse geschrieben werden. Im Akkumulator steht die Speicherstelle dieser Adresse. Die Endadresse des angesprochenen Speicherbereichs errechnet sich aus: X-Register = Low-Byte und Y-Register = High-Byte.

Label: SCRORG
Adresse: $FFED / 65517
Register: A,X,Y
SCRORG schreibt die Werte des aktuellen Fensters in die Register:
Akkumulator: Maximale Spaltenzahl
X-Register : Anzahl der Spalten
Y-Register : Anzahl der Zeilen

Label: SECND
Adresse: $FF93 / 65427
Register: A
Der Wert des Akkumulators wird nach LISTEN als Sekundäradresse auf dem IEC-Bus ausgegeben.

Label: SETBNK
Adresse: $FF68 / 65384
Register: A,Y
Lädt oder eröffnet man eine Datei, muß vorher diese Routine angesprungen werden. Im Y-Register befindet sich der Konfigurationsindex des Dateinamens, im Akkumulator der des aktuellen Speicherbereiches.

Label: SETLFS
Adresse: $FFBA / 65466
Register: A,X,Y
Eröffnet Files mit der Konfiguration
Akkumulator = logische Filenummer
X-Register = Geräteadresse
Y-Register = Sekundäradresse
Die Registerwerte werden durch SETFLS nach $B8 bis $BA gespeichert.

Label: SETMSG
Adresse: $FF90 / 65424
Register: A
Speichert den Akkumulator in $9D. Ist das höchste Bit in $9D gesetzt, gibt der Commodore 128 Systemmeldungen aus, ist das höchste Bit = 0, werden Systemmeldungen verhindert.

Label: SETNAM
Adresse: $FFBD / 65469
Register: A,X,Y
Diese Routine muß angesprungen werden, bevor man eine Datei öffnet. Im Akkumulator steht die Anzahl der Zeichen des Filenamens,

WWW.HOMECOMPUTERWORLD.COM

im X-Register steht der Low-Wert und im Y-Register der High-Wert der Adresse, ab der der Filename steht. Mit SETBNK muß man die notwendigen Pointer der Speicherkonfiguration setzen.

Label: SETTIM
Adresse: $FFDB / 65499
Register: A,X,Y
Setzt die IRQ-gesteuerte Systemuhr ab Adresse $A0.
Akkumulator = Stunden
X-Register = Minuten
Y-Register = Sekunden.

Label: SETTMO
Adresse: $FFA2 / 65442
Register: A
Setzt man Bit 7 des Akkumulators, bevor man SETTMO anspringt, wird der Timeout der IEEE-Routinen ermöglicht. Ansonsten sperrt der nach $0A0E geschriebene Akkumulator IEEE-Timeouts.

Label: STOP
Adresse: $FFE1 / 65505
Register: Zero
Prüft die RUN/STOP-Taste. Wurde die Taste betätigt, ist das Zero-Flag gesetzt (= 1), ansonsten löscht Unterprogramm STOP das Zero-Flag. Bei gesetztem Zero-Flag wird ein Sprung in CLRCH durchgeführt.

Label: SWAPPER
Adresse: $FF5F / 65375
Register: keine
Schaltet zwischen 40-Zeichen- auf 80-Zeichen- und 80-Zeichen- auf 40-Zeichen-Darstellung um. SWAPPER tauscht die Speicherstellen $E0 bis $FA mit $0A40 bis $0A5A.

Label: TALK
Adresse: $FFB4 / 65460
Register: A
Schaltet das unter der im Akkumulator vorhandenen Geräteadresse angeschlossene Gerät zum Abspielen seiner Informationen auf. Ende = UNTLK.

Label: TKSA
Adresse: $FF96 / 65430
Register: A
Sendet den IEC-BUS nach erfolgtem TALK die Sekundäradresse im Akkumulator.

Label: UDTIM
Adresse: $FFEA / 65514
Register: keine
Inkrementiert die IRQ-gesteuerte Systemuhr um eine Einheit.

Label: UNLSN
Adresse: $FFAE/65454
Register: keine
Sendet dem bisher angesprochenen, externen Gerät Abschaltbefehl (Unlisten).

Label: UNTLK
Adresse: $FFAB / 65451
Register: keine
Schaltet das bisher angesprochene externe Gerät ab (Untalk).

Label: VECTOR
Adresse: $FF8D / 65421
Register: X,Y, Carry
Ist das Carry-Flag gesetzt, kopiert VECTOR die Systemvektoren ab Adresse $0314 in den Bereich: LOW = X-Register, High = Y-Register auf die Systemvektoren ab Adresse $0314.

dezimal	hexdez	Label
19910	$4DC6	END
57344	$E000	RESET
49474	$C142	CLRWIN
49482	$C150	CURHOM
49752	$C258	GETLIN
50989	$C72D	BSOUT SCRN
51069	$C77D	CLQIR
65357	$FF4D	C64MODE
65350	$FF50	DMACALL
65353	$FF53	BOOTCALL
65366	$FF56	PHOENIX
65369	$FF59	LKUPLA
65372	$FF5C	LKUPSA
65375	$FF5F	SWAPPER
65378	$FF62	DLCHR
65381	$FF65	PFKEY
65384	$FF68	SETBNK
65387	$FF6B	GETCONF
65390	$FF6E	JSRFAR
65393	$FF71	JMPFAR
65396	$FF74	INDFET
65399	$FF77	INDSTA
65402	$FF7A	INDCMP
65405	$FF7D	PRIMM
65409	$FF81	CINIT
65412	$FF84	IOINIT
65415	$FF87	RAMTAS
65421	$FF8D	VECTOR
65424	$FF90	SETMSG
65427	$FF93	SECND
65430	$FF96	TKSA
65433	$FF99	MEMTOP
65436	$FF9C	MEMBOT
65439	$FF9F	KEY
65442	$FFA2	SETTMO
65445	$FFA5	ACPTR
65448	$FFA8	CIOUT
65451	$FFAB	UNTLK
65454	$FFAE	UNLSN
65457	$FFB1	LISTN
65460	$FFB4	TALK
65463	$FFB7	READST
65466	$FFBA	SETLFS
65469	$FFBD	SETNAM
65472	$FFC0	OPEN
65475	$FFC3	CLOSE
65478	$FFC6	CHKIN
65481	$FFC9	CKOUT
65484	$FFCC	CLRCH
65487	$FFCF	BASIN
65490	$FFD2	BSOUT
65493	$FFD5	LOADSP
65496	$FFD8	SAVESP
65499	$FFDB	SETTIM
65502	$FFDE	RDTIM
65505	$FFE1	STOP
65508	$FFE4	GETIN
65511	$FFE7	CLALL
65514	$FFEA	UDTIM
65517	$FFED	SCRORG
65520	$FFF0	PLOT
65523	$FFF3	IOBASE

uf

Die Unterroutinen numerisch geordnet



WWW.HOMECOMPUTERWORLD.COM

Z80-Programmiercodes im Überblick

Bisher wurden Z80-Maschinenprogramme vernachlässigt. Grund: Der C64 besitzt eine 6502-Zentraleinheit und Zusatzkarten mit Z80-Prozessor sind teuer. Doch seit Commodore den 128'er, einen Computer mit Z80-Prozessor, anbietet, interessieren sich immer mehr Computerbenutzer dafür, den Z80-Programmiercode zu erlernen. Die Befehle des Z80:

Die Programmiersprache Assembler (Maschinencode) ist abhängig vom verwendeten Mikroprozessor. Folglich sind die Befehlswörter der Prozessoren 6502 (Motorola) und Z80 (Zilog) verschieden.

ADC Akkumulator,Operand
Addiert Operand unter Berücksichtigung des Carry-Bits zum Akkumulator und setzt, entsprechend dem Ergebnis, die Prozessorregister.

ADC HL,Operand
Ähnlich ADC Akkumulator, Operand, nur daß der Akkumulator durch das 16-Bit-Register HL ersetzt ist.

ADD Akkumulator,Operand
Ähnlich ADC Akkumulator, Operand, jedoch wird das Carry-Bit nicht berücksichtigt.

ADD HL,Operand
Ähnlich ADC HL, Operand, jedoch wird das Carry-Bit nicht berücksichtigt.

AND Operand
Logische UND-Verknüpfung zwischen Akkumulator und Operanden. Ergebnis steht im Akkumulator, Register sind dem Ergebnis entsprechend verändert.

BIT Zahl,Operand
Das Bit mit Nummer *Zahl* des Operanden wird auf 1 = an oder 0 = aus geprüft. Bei 1 ist das Z-Register 0, bei 0 = 1.

CALL Speicher
Ähnelt dem Gosub-Befehl in Basic. CALL spricht das Unterprogramm ab $Speicher an.

CALL Bedingung,Speicher
Springt nur dann das Unterprogramm in $Speicher an, wenn die gestellte Bedingung (Flags S-, Z-, A- und C-Tests möglich), erfüllt ist.

CCF
Komplementiert das Carry-Bit. (0 wird zu 1, 1 zu 0).

CP Operand
Vergleicht (ComPare) den Wert des Operanden mit dem Wert des Akkumulators und setzt die Flags entsprechend dem Ergebnis.

CPD
Vergleicht den Akkumulator mit der Speicherstelle des 16-Bit-Registers. Sind beide Werte gleich (das Ergebnis der Subtraktion 0), ist das Z-Flag gesetzt (= 1). Ist der Akkuwert größer oder gleich dem Operanden-Inhalt, ist S = 0. Ansonsten steht in S = 1, von HL und BC werden der Wert 1 abgezogen.

CPDR
Wiederholt die CPD-Anweisung, bis entweder Speicherstelle mit Adresse HL gleich dem Akkumulator-Wert oder BC = 0 ist.

CPI
Anstatt von den Registern HL und BC den Wert 1 zu subtrahieren, erhöht der Z80-Prozessor den Wert dieser 16-Bit-Register um eins. Ansonsten ist CPI funktionsgleich zu CPD.

CPIR
Wiederholt so lange die CPI-Anweisung, bis entweder Speicherstelle mit Adresse HL gleich dem Akkumulator-Wert oder BC = 0 ist.

CPL
Invertiert die Bits des Akkumulators (aus 0 wird 1, aus 1 wird 0).

DAA
Korrigiert falsche BCD-Werte im Akkumulator und Kontrollflags.

DEC Operand
Dekrementiert den Operanden (zieht vom Wert des Operanden 1 ab). Verändert die Kontrollregister nach Ergebnis.

DI
Der Z80-Prozessor ignoriert alle weiteren Interrupts bis EI.

DJNZ Zahl
Dekrementiert B um eins und vergleicht den neuen Wert mit 0. Ist er größer, arbeitet das Programm an der aktuellen Adresse + Zahl weiter.

EI
Ermöglicht mit DI abgeschaltete Interruptsprünge.

EX Operand1,Operand2
Schreibt den Wert des Operanden 1 in den Operanden 2 und den des Operanden 2 in den Operanden 1.

EXX
Die Low-Bytes der 16-Bit-Register BC, DE und HL werden mit den High-Bytes dieser Register vertauscht.

HALT
Das Programm wartet auf den nächsten Interrupt. Wurde vorher ein DI gegeben, „stürzt" der Rechner ab.

IM Zahl
Gibt den Interruptmodus an:
Zahl = 0: Wurde ein Interrupt gegeben, steht im Datenbus der auszuführende Befehl.
Zahl = 1: In Adresse $38 wird eine spezielle Interrupt-Routine ausgeführt.
Zahl = 2: Der Z80-Prozessor arbeitet ein Unterprogramm an der Adresse ab, deren Low-Byte im Datenbus und deren High-Byte im I-Register steht.

IN Zahl, (C)
Register C gibt den Port an, aus dem ein Byte in Register Zahl transferiert wird.

INC Operand
Erhöht den Wert des Operanden um 1.

IND
Ein Byte aus Port Nummer C-Register wird in die Speicherstelle unter HL transferiert. Die Werte in

WWW.HOMECOMPUTERWORLD.COM

HL und B werden um eins dekrementiert.

INDR
Wiederholt die IND-Anweisung so lange, bis B = 0 ist.

INI
Ein Byte aus Port Nummer C-Register wird in die Speicherstelle unter HL transferiert. Die Werte in HL und B werden um eins erhöht.

INIR
Wiederholt die INI-Anweisung so lange, bis B = 0 ist.

JP Speicher
Ähnelt dem Basic-Befehl GOTO. Der Z80 arbeitet ab Speicherstelle (Speicher) weiter.

JP Bedingung, Speicher
Ähnelt dem Z80-Befehl CALL Bedingung, Speicher, nur daß aus dem Unterprogramm keine Rückkehr mit RET stattfindet.

JP Operand
Ab Adresse des Operanden arbeitet das Programm weiter.

JR Zahl
Relativer Sprung: Springt zur in PC + Zahl festgelegten Adresse.

JR Bedingung, Zahl
Ähnelt JP Bedingung, Speicher, nur daß die Sprungadresse aus PC + Zahl ermittelt wird.

LD Operand1,Operand2
Transferiert den Inhalt des Operanden 2 in den Operanden 1.

LDD
Transferiert den Inhalt der durch HL bestimmten Speicherstelle in die durch DE bestimmte Speicherstelle. Danach dekrementiert LDD Werte des Register HL, DE und BC.

LDDR
Wiederholt die Z80-Anweisung LDD so lange, bis BC = 0 ist.

LDI
Transferiert den Inhalt der durch HL bestimmten Speicherstelle in die durch DE bestimmte Speicherstelle. Danach inkrementiert LDD die Werte des Register HL, DE und BC.

LDIR
Wiederholt die Z80-Anweisung LDI so lange, bis BC = 0 ist.

HNEG
Multipliziert den Akkumulator-Wert mit (—1) = Vorzeichenwechsel.

NOP
Wartet 2 Taktzyklen lang.

OR Operand
Logische ODER-Verknüpfung des Akkumulators und einem Operanden. Das Ergebnis steht im Akkumulator.

OTDR
Ähnelt der Z80-Anweisung INDR, nur daß der betreffende Wert ausgegeben (statt eingelesen) wird.

OTIR
Ähnelt der Z80-Anweisung INIR, nur daß der betreffende Wert ausgegeben (statt eingelesen) wird.

OUT (C),Zahl
Register C gibt den Port an, in den ein Byte aus Register Zahl transferiert wird.

OUT (Zahl),Akkumulator
Transferiert den Wert des Akkumulators an Port Zahl.

OUTD
Ein Byte wird in Port Nummer C-Register aus der Speicherstelle unter HL transferiert. Die Werte in HL und B werden um eins dekrementiert.

OUTI
Ähnelt OUTD, jedoch werden Werte aus HL und B inkrementiert.

POP Operand
Lädt das Registerpaar des Operanden vom Stapel.

PUSH Operand
Legt das Registerpaar des Operanden auf den Stapel.

RES Zahl,Operand
Schreibt an die Zahlte Stelle des Operanden eine 0.

RET
Die mit CALL angesprungene Routine ist beendet und das Programm arbeitet an der Speicherstelle nach dem CALL-Befehl weiter.

RET Bedingung
Beendet das Unterprogramm nur dann, wenn Bedingung erfüllt ist.

RETI
Rücksprung aus einer Interrupt-Routine (RTI).

RETN
Rücksprung aus einer MNI-Routine. (RUN/STOP-RESTORE)

RL Operand
Rotiert alle Bits des Operanden um eine Stelle nach links. Das 7. Bit wandert ins Carry, im 0. Bit steht der ehemalige Carry-Wert.

RLA
Ähnelt dem RL Operand-Befehl, jedoch beeinflußt er nur das Carry-Kontrollregister.

RLC Operand
Ähnelt RL Operand, jedoch: Bit 0 wird auf 0 gesetzt.

RLCA
Ähnelt RLC Operand, jedoch wird nur das Carry-Bit verändert.

RLD
Führt eine RL-Rotation im BCD-Modus durch.

RR Operand
Ähnelt RL Operand, jedoch wird nach rechts rotiert.

RRA
Entsprechend.

RRD
Führt eine RR-Rotation im BCD-Modus durch.

RST Speicher
Startet Unterprogramm ab Adresse Speicher.

SBC Akkumulator,Operand
Subtrahiert Operand unter Berücksichtigung des Carry-Bits zum Akkumulator und setzt, entsprechend dem Ergebnis, die Prozessorregister.

SBC HL,Operand
Ähnlich SBC Akkumulator,Operand, nur daß der Akkumulator durch das 16-Bit-Register HL ersetzt ist.

SCF
Setzt Carry-Kontrollflag auf 1 (= an).

SET Zahl,Operand
Schreibt an die Zahlte Stelle des Operanden den Wert 1 (= an).

SLA Operand
Schiebt die Bits des Operanden um eins nach links, Bit 7 wird ins Carry geschrieben, Bit 0 wird 0.

SRA Operand
Schiebt die Bits des Operanden um eins nach rechts, Bit 7 bleibt erhalten, Bit 0 kommt ins Carry.

SRL Operand
Schiebt die Bits des Operanden um eins nach rechts, Bit 7 wird 0, Bit 0 kommt ins Carry.

SUB Operand
Zieht vom Akkumulator den Wert des Operanden ab. Das Ergebnis steht im Akkumulator.

XOR Operand
Logische Exklusiv-Oder-Verknüpfung des Akkumulators mit Operanden. Ergebnis Akkumulator.

WWW.HOMECOMPUTERWORLD.COM

C16-116-Plus4: Das steht drin

Man muß die Subroutines des C16/Plus4-Betriebssystems kennen, um diesen Computer effizient zu programmieren.

Wissen ist Macht! Getreu diesem Motto gibt RUN Einblick in die wichtigsten Speicherstellen zwischen $0100 (=257) und $FF00 (=65280). Speichergrenzen verschieben, mathematische Formeln (SIN/TAN) berechnen: Alles steuert man mit diesen Adressen in denen Unterroutinen und Zwischenspeicher des Betriebssystems liegen.

mw

Label	hexadezimal	dezimal
FBUFFR	$0100—$010F	00256—00271
Hilfszwischenspeicher: 15 Bytes		
SAVEA	$0110—$0110	00272—00272
Hilfszwischenspeicher für		
SAVEY	$0111—$0111	00273—00273
SAVE und		
SAVEX	$0112—$0112	00274—00274
RESTORE		
COLKEY	$0113—$0122	00275—00289
Farb-/Luminanztabelle		
HZW0	$0123—$0123	00290—00290
Hilfszwischenspeicher		
SYSSTK	$0124—$01FF	00291—00511
Systemstack: 220 Bytes		
BUF	$0200—$0258	00512—00600
Basic/ Monitor-Eingabepuffer		
OLDLIN	$0259—$025A	00601—00602
Basic-Zwischenspeicher 1		
OLDTXT	$025B—$025C	00603—00604
Basic-Zwischenspeicher 2		
!!!!!!	$025D—$02AC	00605—00684
Dieser Bereich wird vom Disketten-		
organisationssystem (DOS) genutzt:		
XCNT	$025D—$025D	00605—00605
Diskettenbetriebssystem-Schleifenzwischen-		
speicher		
FNBUFR	$025E—$026D	00606—00621
Speicher für Filenamen		
DOSF1L	$026E—$026E	00622—00622
Anzahl der Zeichen des Filenamens # 1		
DOSDS1	$026F—$026F	00623—00623
Laufwerk # 1 — Nummer		
DOSF1A	$0270—$0271	00624—00625
Adresse des Filenamen-Feldes # 1		
DOSF2L	$0272—$0272	00626—00626
Anzahl der Zeichen des Filenamens # 2		
DOSDS2	$0273—$0273	00627—00626
Laufwerk # 2 — Nummer		
DOSF2A	$0274—$0275	00628—00629
Adresse des Filenamen-Feldes # 2		
DOSLA	$0276—$0276	00630—00630
Logische Diskettensystem-Adresse		
DOSFA	$0277—$0277	00631—00631
Geräteadresse		
DOSSA	$0278—$0278	00632—00632
Sekundäradresse		
DOSDID	$0279—$027A	00633—00634
ID-Nummer der Diskette		
DIDCHK	$027B—$027B	00635—00635
ID-Flag		
DOSSTR	$027C-$027C	00636—00636
DOS-Schreibzwischenspeicher		
DOSSPC	$027D—$02AC	00637—00684
Arbeitsspeicher des Diskettensystems		
XPOS	$02AD—$02AE	00685—00686
X-Achse des Grafik-Cursors		
YPOS	$02AF—$02B0	00687—00688
Y-Achse des Grafik-Cursors		
XDEST	$02B1—$02B2	00689—00690
Hilfsregister des Grafik-Cursors X		
YDEST	$02B3—$02B4	00691—00692
Hilfsregister des Grafik-Cursors Y		
XABS	$02B5—$02B6	00693—00694
Hilfsregister		
YABS	$02B7—$02B8	00695—00696
XSGN	$02B9—$02BA	00697—00698
Grafik —		
YSGN	$02BB—$02BC	00699—00700
Routinen.		
FCT1	$02BD—$02BE	00701—00702
Dienen großteils		
FCT2	$02BF—$02C0	00703—00704
als Variablen		
ERRVAL	$02C1—$02C2	00705—00706
für Grafikbefehle		
LESSER	$02C3—$02C3	00707—00707
und		
GREATR	$02C4—$02C4	00708—00708
deren Werte.		
ANGSGN	$02C5—$02C5	00709—00709
Zum Beispiel beinhaltet Winkelrichtung		
(Vorzeichen),		

WWW.HOMECOMPUTERWORLD.COM

SINVAL $02C6—$02C7 00710—00711
Sinuswert der Zahl,
COSVAL $02C8—$02C9 00712—00713
und Cosinuswert der Zahl.
ANGCNT $02CA—$02CB 00714—00715
Zwischenspeicher für Winkelberechnungen.
!!!!! $02CC—$02E0 00716—00736
Dieser Speicherbereich wird mehrfach, von verschiedenen Routinen genutzt. Die Bedeutungen:
HZW1 $02CC—$02CC 00716—00716
Hilfszwischenspeicher
BNR $02CD—$02CD 00717—00717
Anfangspointer
ENR $02CE—$02CE 00718—00718
Endpointer
DOLR $02CF—$02CF 00719—00719
Dollarflag
FLAG $02D0—$02DO 00720—00720
Kommaflag
SWE $02D1—$02D1 00721—00721
Zählspeicher
USGN $02D2—$02D2 00722—00722
Vorzeichen Exponent
UEXP $02D3—$02D3 00723—00723
Exponentenzeiger
VN $02D4—$02D4 00724—00724
Anzahl der Zeichen vor dem Komma bei Ausgabe mit PRINT USING.
CHSN $02D5—$02D5 00725—00725
Fehler-Flag
VF $02D6—$02D6 00726—00726
Anzahl der Felder vor dem Komma bei Ausgabe mit PRINT USING.
NF $02D7—$02D7 00727—00727
Anzahl der Felder nach dem Komma bei Ausgabe mit PRINT USING.
POSP $02D8—$02D8 00728—00728
Vorzeichenflag
FESP $02D9—$02D9 00729—00729
Flag für Exponenten
ETOF $02DA—$02DA 00730—00730
Hilfsspeicher
CFORM $02DB—$02DB 00731—00731
Zeichenzähler
SNO $02DC—$02DC 00732—00732
Zeichennummerzwischenspeicher
BLFD $02DD—$02DD 00733—00733
Flag: Ist Zeichen „ " oder „ *"?
BEGFD $02DE—$02DE 00734—00734
Feldanfangszeiger (PRINT USING)
LFOR $02DF—$02DF 00735—00735
Formatlänge (PRINT USING)
ENDFD $02E0—02E0 00736—00736
Feldendezeiger (PRINT USING)

Weitere Bedeutungen dieses Speicherbereiches:
XCENTR $02CC—$02CD 00716—00717
Bestimmen
YCENTR $02CE—$02CF 00718—00719
und
XDIST1 $02D0—$02D1 00720—00721
steuern
YDIST1 $02D2—$02D3 00722—00723
den
XDIST2 $02D4—$02D5 00724—00725
Grafikcursor
YDIST2 $02D6—$02D7 00726—00727
am
HZW3 $02D8—$02D9 00728—00729
Bildschirm
COLCNT $02DA—$02DA 00730—00730
Horizontale Zeichenzahl/Stelle
ROWCNT $02DB—$02DB 00731—00731
Vertikale Zeichenzahl/Stelle
STRCNT $02DC—$02DC 00732—00732
Zeichenzähler allg.

Weitere Bedeutungen des Speicherbereiches von $02CC bis $02E3:
XCORD1 $02CC—$02CD 00716—00717
Zwischen-
YCORD1 $02CE—$02CF 00718—00719
speicher
BOXANG $02D0—$02D1 00720—00721
zur Steuerung
XCOUNT $02D2—$02D3 00722—00723
verschiedener
YCOUNT $02D4—$02D5 00724—00725
Basic 3.5-
BXLENG $02D6—$02D7 00726—00727
Grafik-
XCORD2 $02D8—$02D9 00728—00729
befehle
YCORD2 $02DA—$02DB 00730—00731
(Variablen).
Weitere Bedeutungen dieser Stellen:
XCIRCL $02CC—$02CD 00716—00717
CIRCLE-Funktion: Der Zentrumspunkt, **X-Achsenwert.**
YCIRCL $02CE—$02CF 00718—00719
CIRCLE-Funktion: Der Zentrumspunkt, **Y-Achsenwert.**
XRADUS $02D0—$02D1 00720—00721
Wert des Radius in X-Achsenrichtung
YRADUS $02D2—$02D3 00722—00723
Wert des Radius in Y-Achsenrichtung
ROTANG $02D4—$02D5 00724—00725
Rotationswinkel
HZW4 $02D6—$02D7 00726—00727
Hilfsspeicher
ANGBEG $02D8—$02D9 00728—00729
Winkel von . . .
ANGEND $02DA—$02DB 00730—00731
Winkel bis . . .
XRCOS $02DC—$02DD 00732—00733
Cosinus-Wert des X-Radius
YRSIN $02DE—$02DF 00734—00735

WWW.HOMECOMPUTERWORLD.COM

Sinus-Wert des Y-Radius
XRSIN $02E0—$02E1 00736—00737
Sinus-Wert des X-Radius
YRCOS $02E2—$02E3 00738—00739
Cosinus-Wert des Y-Radius
Die letzte Version, welche Bedeutungen dieser Speicherbereich hat:
HZW5 $02CC—$02CC 00716—00716
Hilfsspeicher
KEYLEN $02CD—$02CD 00717—00717
Tastatureingabe: Anzahl
KEYNXT $02CE—$02CCE 00718—00718
Aktuelles Eingabezeichen
STRSZ $02CF—$02CF 00719—00719
Länge des aktuellen Strings
GETTYP $02D0—$02D0 00720—00720
Variable ändern
STRPTR $02D1—$02D1 00721—00721
Anzeige: Welcher String
OLDBYT $02D2—$02D2 00722—00722
Zwischenspeicher-Grafikausgabe
NEWBT $02D3—$02D3 00723—00723
Nummer des neuen Strings
HZW6 $02D4—$02D4 00724—00724
Hilfszwischenspeicher
XSIZE $02D5—$02D6 00725—00726
Shape: Horizontales Maximum (Größe)
YSIZE $02D7—$02D8 00727—00728
Shape: Vertikales Maximum (Größe)
XSAVE $02D9—$02DA 00729—00730
Hilfszwischenspeicher
STRADR $02DB—$02DC 00731—00732
Shape: Stringvariable des Shapes
BITIDX $02DD—$02DD 00733—00733
Zwischenspeicher: Bitindex
SAVSIZ $02DE—$02E1 00734—00737
Zwischenspeicher
HZW7 $02E2—$02E3 00738—00739
Zwischenspeicher
CHRPAG $02E4—$02E4 00740—00740
High Byte der Character-ROM Adresse
BITCNT $02E5—$02E5 00741—00741
GSHAPE-Befehl: Zwischenspeicher
SCALEM $02E6—$02E6 00742—00742
SCALE-Flag
WIDTH $02E7—$02E7 00743—00743
WIDTH-Flag
FILFLG $02E8—$02E8 00744—00744
BOX-Flag
BITMSK $02E9—$02E9 00745—00745
Zwischenspeicher
NUMCNT $02EA—$02EA 00746—00746
Zwischenspeicher
TRCFLG $02EB—$02EB 00747—00747
TRACE-Flag: $00 bedeutet aus, $80 bedeutet aktiviert
T3 $02EC—$02EC 00748—00748
Zwischen-
T4 $02ED—$02EE 00749—00750
speicher
VTEMP3 $02EF—$02EF 00751—00751
für Grafikbefehle
VTEMP4 $02F0—$02F0 00752—00752
Parameteranzahl des Grafikbefehls
VTEMP5 $02F1—$02F1 00753—00753
Flag: 1=relativ, 0=absolut
ADRAY1 $02F2—$02F3 00754—00755
Pointer auf die Umwandlungsroutine: Fließkomma in Integer
ADRAY2 $02F4—$02F5 00756—00757
Pointer auf die Umwandlungsroutine: Integer in Fließkomma
USRSPC $02F6—$02FD 00758—00765
Freiraum für eigene Pointer
BNKVEC $02FE—$02FF 00766—00767
Cartridge-Pointer
IERROR $0300—$0301 00768—00769
Zeigt auf die Fehlermelderoutine
IMAIN $0302—$0303 00770—00771
Zeigt auf die Basic-Warmstartroutine
ICRNCH $0304—$0305 00772—00773
Zeigt auf die Token-Definierungsroutine
IQPLOP $0306—$0307 00774—00775
Zeigt auf Keywort erzeugen-Routine
IGONE $0308—$0309 00776—00777
Zeigt auf die Ausgabe-Routine
IEVAL $030A—$030B 00778—00779
Zeigt auf Symbol Evaluation-Routine
IESCLK $030C—$030D 00780—00781
Zeigt auf User-Tokendefinition
IESCPR $030E—$030F 00782—00783
Zeigt auf Keywort editieren-Routine
IESCEX $0310—$0311 00784—00785
Zeigt auf Token bearbeiten-Routine
ITIME $0312—$0313 00786—00787
Zeigt auf Interrupt Uhr steuern-Routine
CINV $0314—$0315 00788—00789
Zeigt auf Interrupt-Routine
CBINV $0316—$0317 00790—00791
Zeigt auf Break Interrupt-Routine
IOPEN $0318—$0319 00792—00793
Zeigt auf OPEN-Routine
ICLOSE $031A—$031B 00794—00795
Zeigt auf CLOSE-Routine
ICHKIN $031C—$031D 00796—00797
Zeigt auf Kanal für Eingabe öffnen-Routine
ICKOUT $031E—$031F 00798—00799
Zeigt auf Kanal für Ausgabe öffnen-Routine
ICLRCH $0320—$0321 00800—00801
Zeigt auf I/O-Init-Routine
IBASIN $0322—$0323 00802—00803
Zeigt auf Zeichen einlesen-Routine
IBSOUT $0324—$0325 00804—00805
Zeigt auf Zeichen ausgeben-Routine
ISTOP $0326—$0327 00806—00807
Zeigt auf STOP-Taste prüfen-Routine
IGETIN $0328—$0329 00808—00809
Zeigt auf GETIN-Routine

WWW.HOMECOMPUTERWORLD.COM

ICLALL $032A—$032B 00810—00811
Zeigt auf alle Files schließen-Routine
USRCMD $032C—$032D 00812—00813
Zeigt auf BRK-Routine
ILOAD $032E—$032F 00814—00815
Zeigt auf die LOAD-Routine
ISAVE $0330—$0331 00816—00817
Zeigt auf die SAVE-Routine
USRSP1 $0332—$0332 00818—00818
Hilfsspeicher
TAPBUF $0333—$03F2 00819—01010
Kassetten READ/WRITE-Puffer
WRLEN $03F3—$03F4 01011—01012
Zeichencounter bei WRITE
RDCNT $03F5—$03F6 01013—01014
Zeichencounter bei READ
INPQUE $03F7—$0436 01015—01078
RS232-Schnittstellenpuffer
ESTAKL $0437—$0454 01079—01108
Systemzwischenspeicher/Arbeitsbereich Low
ESTAKH $0455—$0472 01109—01138
Systemzwischenspeicher/Arbeitsbereich High
CHRGET $473—$0478 01139—01144
Zeichen von Basic-Text einlesen-Routine
CHRGOT $0479—$0484 01145—01156
letztes, mit CHRGET eingelesenes Zeichen nochmals einlesen.
QNUM $0485—$0493 01157—01171
Zwischenspeicher für RS232-Handling
INDSUB $0494—$04A1 01172—01185
ROM/RAM Umschaltroutine
ZERO $04A2—$04A4 01186—01188
Basic-Hilfsspeicher
INDTEXT $04A5—$04AF 01189—01199
Textpointer
INDIN1 $04B0—$04BA 01200—01210
Index # 0 und # 1
INDIN2 $04BB—$04C5 01211—01221
Index # 2
INDST1 $04C6—$04D0 01222—01232
String # 1
INDLOW $04D1—$04DB 01233—01243
Hilfsspeicher
INDFMO $04DC—$04E6 01244—01254
Fließkommaakkumulator-Speicher
PUFILL $04E7—$04E7 01255—01255
Hilfs-
PUCOMA $04E8—$04E8 01256—01256
speicher
PUDOT $04E9—$04E9 01257—01257
der
PUMONY $04EA—$04EA 01258—01258
PRINT USING-Anweisung
TMPDES $04EB—$04EE 01259—01262
Variable der INSTR-Routine
ERRNUM $04EF—$04EF 01263—01263
Der Nummerncode des zuletzt aufgetretenen Fehlers
ERRLIN $04F0—$04F1 01264—01265
Zeilennummer, in welcher der letzte Fehler auftrat
TRAPNO $04F2—$04F3 01266—01267
Zeilennummer, zu der das Programm verzweigt, wenn ein Fehler auftritt
TMPTRP $04F4—$04F4 01268—01268
Fehlernummer kurzzeitig zwischengespeichert
ERRTXT $04F5—$04F6 01269—01270
Pointer auf Adresse, in der die Fehlermeldung (Text) steht
OLDSTK $04F7—$04F7 01271—01271
Hilfsspeicher zur Fehlerbehandlung
TMPTXT $04F8—$04F9 01272—01273
zuletzt benutzter Basic-Text
TMPLIN $04FA—$04FB 01274—01275
zuletzt benutzte Basic-Zeile
MTIMLO $04FC—$04FD 01276—01277
Interrupt-Uhr Hilfsspeicher Low
MTIMHI $04FE—$04FF 01278—01279
Interrupt-Uhr Hilfsspeicher High
USRPOK $0500—$0502 01280—01282
Basic-Hilfsspeicher
RNDX $0503—$0507 01283—01287
Zufallszahl
DEJAVU $0508—$0508 01288—01288
Flag für Kalt-/Warmstart
LAT $0509—$0512 01289—01298
In diesem Speicherbereich befindet sich die Tabelle der logischen Filenummern aller gerade geöffneten Dateien.
FAT $0513—$051C 01299—01308
In diesem Speicherbereich befindet sich die Tabelle der Gerätenummern aller gerade geöffneten Dateien.
SAT $051D—$0526 01309—01318
In diesem Speicherbereich befindet sich die Tabelle der Sekundäradressen aller gerade geöffneten Dateien.
KEYD $0527—$0530 01319—01328
Tastaturpuffer: Maximal 10 Zeichen.
MEMSTR $0531—$0532 01329—01330
Zeiger auf Basic-Anfang
MSIZ $0533—$0534 01331—01332
Zeiger auf Basic-Ende
TIMOUT $0535—$0535 01333—01333
IEEE Timeout-Flag
FILEND $0536—$0536 01334—01334
Ist das Ende eines Files erreicht, so steht in dieser Speicherstelle der Wert 01, ansonsten steht dort 00.
CTALLY $0537—$0537 01335—01335
Anzahl der Zeichen im Puffer, die noch ausgelesen werden
CBUFVA $0538—$0538 01336—01336
Anzahl aller Zeichen im Puffer
TPTR $0539—$0539 01337—01337
Pointer: Zeigt auf das nächste aktuelle Zeichen im Puffer
FLTYPE $053A—$053A 01338—01338

WWW.HOMECOMPUTERWORLD.COM

Gibt den File-Typ des gerade bearbeiteten Programmes der Datasette an

COLOR	$053B—$053B	01339—01339

Attribut-Byte der COLOR-Anweisung

FLASH	$053C—$053C	01340—01340

Flag für FLASH.

USRFRE	$053D—$053D	01341—01341

Frei nutzbar für Anwender

HIBASE	$053E—$053E	01342—01342

Zwischenspeicher der

XMAX	$053F—$053F	01343—01343

Bildschirmausgabe

RPTFLG	$0540—$0540	01344—01344

Gibt an, welche Tasten mit automatischer REPEAT-Funktion ausgestattet sind: $00=üblich, $40=keine, $80=alle

KOUNT	$0541—$0541	01345—01345

Counter für die

DELAY	$0542—$0542	01346—01346

REPEAT-Funktion

SHFLAG	$0543—$0543	01347—01347

Gibt an, ob SHIFT in Funktion ist.

LSTSHF	$0544—$0544	01348—01348

Letztes Zeichen in SHIFT-Mode

KEYLOG	$0545—$0546	01349—01350

Zeigt auf die KEYLOG-Routine

MODE	$0547—$0547	01351—01351

Flag: Gibt an, ob sich der Computer im Text- oder Grafikmodus befindet.

AUTODN	$0548—$0548	01352—01352

Scroll-Flag: bei 0=aktiviert

LINTMP	$0549—$0549	01353—01353

Basic-Zwischenspeicher

ROLFLG	$054A—$054A	01354—01354
FORMAT	$054B—$054B	01355—01355

TEDMON-Zwischenspeicher

MSAL	$054C—$054E	01356—01358

Zwischenspeicher

WRAP	$054F—$054F	01359—01359

des

TMPC	$0550—$0550	01360—01360

CPU 7501-

DIFF	$0551—$0551	01361—01361

Mikroprozessors

PCH	$0552—$0552	01362—01362

Program Counter High

PCL	$0553—$0553	01363—01363

Program Counter Low

FLGS	$0554—$0554	01364—01364

Flags: Zero, negativ, Carry, etc.

ACC	$0555—$0555	01365—01365

Akkumulator

XR	$0556—$0556	01366—01366

X-Register

YR	$0557—$0557	01367—01367

Y-Register

SP	$0558—$0558	01368—01368

Stackpointer

INVL	$0559—$0559	01369—01369

Weitere Prozessor-

INVH	$055A—$055A	01370—01370

hilfsregister

CMPFLG	$055B—$055B	01371—01371

Zwischenspeicher Vergleichsroutinen

BAD	$055C—$055C	01372—01372

Fehlerflag

KYNDX	$055D—$055D	01373—01373

Nummer der aktuellen Funktionstaste

KEYIDX	$055E—$055E	01374—01374

Zeiger auf den Text der aktuellen Funktionstaste

KEYBUF	$055F—$0566	01375—01382

Tabelle der Funktionstasten: Zeigt die Länge der einzelnen Texte an.

PKYBUF	$0567—$05E6	01383—01510

Texte der einzelnen Funktionstasten

KDATA	:$05E7—$05E7	01511—01511
KDYCMD	:$05E8—$05E8	01512—01512
KDYNUM	:$05E9—$05E9	01513—01513
KDYPRS	:$05EA—$05EA	01514—01514
KDYTYP	:$05EB—$05EB	01515—01515
SAVRAM	:$05EC—$06EB	01516—01771

Dieser Speicherbereich (eine Speicherseite) wird von BANKING-Routinen belegt und genutzt.

PAT	$05EC—$05EF	01516—01519

Adreßtabelle

LNGJMP	$05F0—$05F1	01520—01521

Long Jump Adresse

FETARG	$05F2—$05F2	01522—01522

Long Jump Akkumulator

FETXRG	$05F3—$05F3	01523—01523

Long Jump X-Register

FEXTSRG	$05F4—$05F4	01524—01524

Long Jump Status Register

AREAS	$05F5—$065D	01525—01629

RAM-Speicherbereich der BANKING-Routinen

ASPECH	$065E—$06EB	01630—01771

RAM-Freier Bereich

STKTOP	$06EC—$07AF	01772—01967

Basic-Pseudo Stack

WROUT	$07B0—$07B0	01968—01968

Aktuelles Byte, das auf Datasette geschrieben wird

PARITY	$07B1—$07B1	01969—01969

Zwischenspeicher bei Berechnungen (Parity)

TT1	$07B2—$07B2	01970—01970

Zwischenspeicher für

TT2	$07B3—$07B3	01971—01971

die HEADER-Routine

USRFR2	$07B4—$07B4	01972—01972

Frei für Benutzersoftware

RDBITS	$07B5—$07B5	01973—01973

Indexspeicher für READBYTE-Routine

ERRSP	$07B6—$07B6	01974—01974

Fehler Stack Pointer

FPERRS	$07B7—$07B7	01975—01975

Nummer des ersten Fehlers

DSAMP1	$07B8—$07B9	01976—01977

Zeitkonstante

WWW.HOMECOMPUTERWORLD.COM

DSAMP2	$07BA—$07BB	01978—01979
Zeitkonstante		
ZCELL	$07BC—$07BD	01980—01981
Zeitkonstante		
SRECOV	$07BE—$07BE	01982—01982
STOPKEY-Flag		
DRECOV	$07BF—$07BF	01983—01983
DROPKEY-Flag		
TRSAVE	$07C0—$07C3	01984—01987
RDBLOCK-Flag		
RDETMP	$07C4—$07C4	01988—01988
Zwischenspeicher		
LDRSCN	$07C5—$07C5	01989—01989
Zeitspeicher		
CDERRM	$07C6—$07C6	01990—01990
Fehler bei der letzten Kassettenoperation		
VSAVE	$07C7—$07C7	01991—01991
Zwischenspeicher des VERIFY-Befehls		
T1PIPE	$07C8—$07CB	01992—01995
T1-Zwischenspeicher		
ENEXT	$07CC—$07CC	01996—01996
Flag für Lesefehler		
UOUTQ	$07CD—$07CD	01997—01997
Aktuell zu sendendes Byte		
UOUTFG	$07CE—$07CE	01998—01998
Pufferflag für UOUTQ: 00=leer, 01=voll		
SOUTQ	$07CF—$07CF	01999—01999
Systembyte, das gesendet wird		
SOUNFG	$07D0—$07D0	02000—02000
Pufferflag für SOUTQ: 00=leer, 01=voll		
INQFPT	$07D1—$07D1	02001—02001
Pointer auf RS232-Pufferanfang		
INQRPT	$07D2—$07D2	02002—02002
Pointer auf RS232-Pufferende		
INQCNT	$07D3—$07D3	02003—02003
Anzahl der Zeichen im RS232-Puffer		
ASTAT	$07D4—$07D4	02004—02004
Statusspeicherstelle von ACIA		
AINTMP	$07D5—$07D5	02005—02005
Zwischenspeicher der INPUT-Routine		
ALSTOP	$07D6—$07D6	02006—02006
Flag für STOP		
ARSTOP	$07D7—$07D7	02007—02007
Flag für CTRL-S		
APRES	$07D8—$07D8	02008—02008
ACIA-Flag		
KLUDES	$07D9—$07AE	02009—02020
Zwischenspeicher des Systems		
SCBOT	$07E5—$07E5	02021—02021
Unterer Rand des Bildschirms		
SCTOP	$07E6—$07E6	02022—02022
Oberer Rand des Bildschirms		
SCLF	$07E7—$07E7	02023—02023
Linker Rand des Bildschirms		
SCRT	$07E8—$07E8	02024—02024
Rechter Rand des Bildschirms		
SCRDIS	$07E9—$07E9	02025—02025
Scroll-Flag: $00=ein, $80=aus		
INSFLG	$07EA—$07EA	02026—02026
INSERT-Flag: $00=kein Einfügen, $80=automatisches Einfügen		
LSTCHR	$07EB—$07EB	02027—02027
ESC-Flag: ESC-Code=27		
LOGSCR	$07EC—$07EC	02028—02028
Zwischenspeicher der		
TCOLOR	$07ED—$07ED	02029—02029
Bildschirmroutinen		
BITABL	$07EE—$07F1	02030—02033
Umrechnungstabelle		
SAREG	$07F2—$07F2	02034—02034
Akkumulator bei SYS-Befehl		
SXREG	$07F3—$07F3	02035—02035
X-Register bei SYS-Befehl		
SYREG	$07F4—$07F4	02036—02036
Y-Register bei SYS-Befehl		
SPREG	$07F5—$07F5	02037—02037
Status Register bei SYS-Befehl		
LSTX	$07F6—$07F6	02038—02038
Indexregister bei Tastaturabfragen		
STPDSB	$07F7—$07F7	02039—02039
CTRL-S Flag: Steht hier $00, so wirkt CTRL-S, steht $06 in STPDSB, so ist die CTRL-S Funktion gesperrt.		
RAMROM	$07F8—$07F8	02040—02040
Steht hier $00, liest TEDMON aus dem ROM, bei $01 liest er aus dem RAM		
COLSW	$07F9—$07F9	02041—02041
Flag der Farb/Luminanz-Tabelle		
FFRMSK	$07FA—$07FA	02042—02042
ROM-Maske der Bildschirmsteuerung		
VMBMSK	$07FB—$07FB	02043—02043
VM-Maske der Bildschirmsteuerung		
LSEM	$07FC—$07FC	02044—02044
Steuert den Datasettenmotor		
PALCNT	$07FD—$07FD	02045—02045
PAL-Register		
USRFR2	$07FE—$07FF	02046—02047
Frei nutzbarer Speicher		
TEDATR	$0800—$0BFF	02048—03071
Farbspeicher im Textmodus		
TEDSCN	$0C00—$0FFF	03072—04095
Bildschirmspeicher im Textmodus		
BASBGN	$1000—$. . ..	04096—.....
Basic-Start im Textmodus		
BMLUM	$1800—$1BFF	06144—07167
Luminanztabelle im Grafikmodus		
BMCOLR	$1C00—$1FFF	07168—08191
Farbtabelle im Grafikmodus		
GRBASE0	$2000—.....	08192—.....
Grafikbildschirm (nur Grafikmodus)		
GRABSE1	$4000—.....	16384—.....
Basic-Start im Grafikmodus		
CHRBAS	$D000	53248
Zeichensatz-ROM		
TED	$FF00	65280
Beginn der TED (Chip 7360=Text-Display)-Register		

WWW.HOMECOMPUTERWORLD.COM

Von 0 auf 255: Nullseite des C16

Wichtige Subroutinen des Basic-Interpreters und des Diskettenbetriebssystemes nutzen intensiv Speicherstellen im Bereich von 0 bis 255. Ändert man diese Speicherstellen sinnvoll ab, erzielt man mit wenig Einsatz in Basic und Assembler viel Wirkung.

Basic-Speicheranfang, zuletzt bearbeitete Programmzeile oder zuletzt benutzter String: In der Nullseite des C16 ist alles gespeichert. Denn dort stehen alle Zwischenspeicher, die der Computer besonders oft nutzt und entsprechend wichtig sind. Man muß nur wissen, welche Unterprogramme auf den Speicher zugreifen . . .

nw

Label	Adresse	Funktion
PDIR	$00—$00	Datenrichtungsregister
	000—000	des Prozessors 7501.
PORT	$01—$01	Input/Output-Register
	001—001	des 7501
SRCHTK	$02—$02	Schleifen-
	002—002	flag
ZPVEC1	$03—$04	Neue Startadresse bei
	003—004	RENUMBER-Anweisung
ZPVEC2	$05—$06	Schrittweite der RE-
	005—006	NUMBER-Anweisung
CHARAC	$07—$07	Beinhaltet gesuchtes
	007—007	Zeichen (=Character)
ENDCHR	$08—$08	Flag für Hochkomma-
	008—008	modus
TRMPOS	$09—$09	Wert des letzten
	009—009	TABulatorschrittes
VERCK	$0A—$0A	Flag: Bei 0 = LOAD,
	010—010	bei 1 = VERIFY
COUNT	$0B—$0B	Eingabepufferzeiger/
	011—011	Zahl der Elemente
DIMFLG	$0C—$0C	Flag bei Normal-
	012—012	DIMensionierungen
VALTYP	$0D—$0D	Datentypflag: $FF=
	013—013	String, $00=Numerisch
INTFLG	$0E—$0E	Datentypflag: $80=
	014—014	INT, $00=Fließkomma
DORES	$0F—$0F	Flag für DATA, LIST,
	015—015	Garbage Collection
SUBFLG	$10—$10	Flag für FN (x)-
	016—016	Anweisungen
INPFLG	$11—$11	Flag: $00=INPUT,
	017—017	$40=GET, $98=READ
TANSGN	$12—$12	Vorzeichen bei Basic-
	018—018	Funktion ATN.
CANNL	$13—$13	Flag:
	019—019	INPUT-Prompt
LINNUM	$14—$15	64 K-Adresse für
	020—021	internen Gebrauch
TEMPPT	$16—$16	Pointer aus temp.
	022—022	Stringstapel
LASTPT	$17—$18	Zuletzt benutzter
	023—024	temp. Stringvektor
TEMPST	$19—$21	Stapel für temp.
	025—033	Strings
INDEX1	$22—$23	Hilfs-
	034—035	pointer 1
INDEX2	$24—$25	Hilfs-
	036—037	pointer 2
RESHO	$26/038	Speicherbereich
RSMOH	$27/039	für
RESMO	$28/040	Produkt
RESLO	$29/041	bei
	$2A/042	Multiplikationen
TXTTAB	$2B—$2C	Pointer auf den
	043—044	Basic-Anfang
VARTAB	$2D—$2E	Pointer auf den
	045—046	Variablen-Anfang
ARYTAB	$2F—$30	Pointer auf
	047—048	Array-Anfang
STREND	$31—$32	Pointer auf
	049—050	Array-Ende+1
FRETOP	$33—$34	Pointer auf
	051—052	String-Speicher
FRESPC	$35—$36	Hilfspointer
	053—054	zur Stringverarbeitung
MEMSIZ	$37—$38	Pointer auf
	055—056	Basic-Speicher Ende
CURLIN	$39—$3A	Gerade bearbeitete
	057—058	Basic-Zeilennummer
TXTPTR	$3B—$3C	Pointer auf
	059—060	Textausgabe
FNDPNT	$3D—$3E	Pointer für CONT=
	061—062	Zeile bei Abbruch
DATLIN	$3F—$40	Zeilennummer der
	063—064	aktuellen DATA-Zeile
DATPTR	$41—$42	Adresse des aktuellen
	065—066	DATA-Zeichens

WWW.HOMECOMPUTERWORLD.COM

INPPTR	$43—$44	Vektor für
	067—068	INPUT
VARNAM	$45—$46	Gerade benutzte
	069—070	Variable
VARPNT	$47—$48	Adresse der aktuellen
	071—072	Variable
FORPNT	$49—$4A	FOR-NEXT Schleifen
	073—074	Variablenpointer
OPPTR	$4B—$4C	Hilfs-
	075—076	pointer
OPMASK	$4D—$4D	Zwischenspeicher für
	077—077	Zeichenvergleiche
DEFPNT	$4E—$4F	078—079 Diverse
DSCPNT	$50—$51	080—081 Hilfs-
	$52—$52	082—082 pointer
HELPER	$53—$53	083—083 Grafik
JMPER	$54—$54	084—084 Pointer
SIZE	$55—$55	085—085 für versch.
OLDOV	$56—$56	086—086 Funktionen
TEMPF1	$57—$57	087—087 Bereich
HIGHDS	$58—$59	088—089 für
HIGHTR	$5A—$5B	090—091 ver-
	$5C-$5C	092—092 schiedene
LOWDS	$5D—$5E	093—094 nume-
LOWTR	$5F—$5F	095—095 rische
EXPSGN	$60—$60	096—096 Operationen
FACEXP	$61—$61	Exponent des
	097—097	Fließkommaacc. 1
FACH0	$62—$62	098—098 Mantisse
FACMOH	$63—$63	099—099 des
FACMO	$64—$64	100—100 Fließkomma-
FACLO	$65—$65	101—101 acc. 1
FACSGN	$66—$66	Vorzeichen des
	102—102	Fließkommaacc. 1
SGNFLG	$67—$67	Flag für
	103—103	Polynomberechnung
BITS	$68—$68	Überlauf des
	104—104	Fließkommaacc. 1
ARGEXP	$69—$69	Exponent des
	105—105	Fließkommaacc. 2
ARGHO	$6A—$6A	106—106 Mantisse
ARGMOH	$6B—$6B	107—107 des
ARGMO	$6C—$6C	108—108 Fließkomma-
ARGLO	$6D—$6D	109—109 acc. 2
ARGSGN	$6E—$6E	Vorzeichen des
	110—110	Fließkommaacc. 2
ARISGN	$6F—$6F	Vorzeichenvergleich
	111—111	zwischen Acc1 und Acc2
FACOV	$70—$70	Low-Stelle des
	112—112	Fließkommaacc. 1
FBUFPT	$71—$72	Pointer für Kassetten-
	113—114	pufferlänge
AUTINC	$73—$74	Zeilennummerinc.
	115—116	bei AUTO-Befehl
MVDFLG	$75—$75	Flag für
	117—117	Grafik
KEYNUM	$76—$76	118—118 Ein- und
KEYSIZ	$77—$77	119—119 Ausgabe
SYNTMP	$78—$78	120—120 Speicher
DSDESC	$79—$7B	Beinhaltet
	121—123	Variable DS$
TOS	$7C—$7D	Hilfspointer
	124—125	für Basic
TMPTON	$7E—$7F	126—126 Hilfs-
VOICNO	$80—$80	128—128 speicher
RUNMOD	$81—$81	129—129 für
POINT	$82—$82	130—130 ver-
GRAPHM	$83—$83	131—131 schiedene
COLSEL	$84—$84	132—132 Sound-
MC1	$85—$85	133—133 an-
FG	$86—$86	134—134 wei-
SCXMAX	$87—$87	135—135 sungen
SCYMAX	$88—$88	136—136 des
LTFLAG	$89—$89	137—137 C16-
RTFLAG	$8A—$8A	138—138 Basics.
STOPNB	$8B—$8B	139—139 Steue-
GRAPNT	$8C—$8D	140—141 rung
VTEMP1	$8E—$8E	142—142 vom
VTEMP2	$8F—$8F	142—143 TED.
STATUS	$90—$90	Status-
	144—144	pointer für ST
STKEY	$91—$91	Flag für
	145—145	RVS/STOP-Taste
SPVERR	$92—$92	146—146 Hilfs-
VERFCK	$93—$93	147—147 speicher
C3PO	$94—$94	Flag für
	148—148	Ausgabe ser. Bus
BSOUR	$95—$95	Speichert Zeichen
	149—149	von/für ser. Bus
XSAV	$96—$96	Hilfs-
	150—150	speicher
LDTNT	$97—$97	Beinhaltet die Zahl
	151—151	geöffneter Dateien
DFLTN	$98—$98	Nummer des
	152—152	Eingabegerätes
DELTO	$99—$99	Nummer des
	153—153	Ausgabegerätes
MSGFLG	$9A—$9A	Flag: $80 = Direktmode
	154—154	$00 = Programmode
SAL	$9B—$9B	Kassetten-
	155—155	puffer- und
SAH	$9C—$9C	Scroll-
	156—156	pointer
EAL	$9D—$9D	Pointer
	157—157	auf
EAH	$9E—$9E	Basic-
	158—158	Programmende
T1	$9F—$A0	Zwischen-
	159—160	speicher 1
T2	$A1—$A2	Zwischen-
	161—162	speicher 2
TIME	$A3—$A5	Echtzeit-
	163—165	uhr
R2D2	$A6—$A6	Register 1 für
	166—166	seriellen Bus
TPBYTE	$A7—$A7	Register für

WWW.HOMECOMPUTERWORLD.COM

	167—167	Kassettenroutine
BSOUR1	$A8—$A8	Register 2 für
	168—168	seriellen Bus
FPVERR	$A9—$A9	Zwischenspeicher
	169—169	für Farbvektor
DCOUNT	$AA—$AA	Register 2 für
	170—170	Kassettenroutine
FNLEN	$AB—$AB	Zeichenlänge des
	171—171	akt. Filenamens
LA	$AC—$AC	Aktuelle
	172—172	logische Filenummer
SA	$AD—$AD	Aktuelle
	173—173	Sekundäradresse
FA	$AE—$AE	Aktuelle
	174—174	Geräteadresse
FNADR	$AF—$B0	Pointer
	175—176	auf den Filenamen
ERRSUM	$B1—$B1	Hilfsspeicher der
	177—177	Fehlerroutine
STAL	$B2—$B2	Input-/
	178—178	Output
STAH	$B3—$B3	Start-
	179—179	adresse
MEMUSS	$B4—$B5	Pointer der aktuellen
	180—181	Ladeadresse
TAPEBS	$B6—$B7	Pointer der akt.
	182—183	Ladeendadresse
TEMP2	$B8—$B9	Hilfs-
	184—185	speicher
WERBAS	$BA—$BB	Pointer auf
	186—187	Kassettenpufferzeichen
IMPARM	$BC—$BD	Hilfs-
	188—189	speicher
FETPTR	$BE—$BF	Register benutzt von
	190—191	der LONG-FETCH Routine
SEDSAL	$C0—$C1	Scroll-
	192—193	register
RVS	$C2—$C2	RVS-
	194—194	flag
INDX	$C3—$C3	INPUT-
	195—195	Pointer
LSXP	$C4—$C4	X-/Y-
	196—196	Position
LSTP	$C5—$C5	des Cursors
	197—197	bei INPUT
SFDX	$C6—$C6	gedrückte Taste
	198—198	($40=keine)
CRSW	$C7—$C7	Eingabe Flag:
	199—199	Bildschirm/ Taste
PNT	$C8—$C9	Pointer auf
	200—201	Bildschirmzeile
PNTR	$CA—$CA	Position des Cursors
	202—202	in der Bildschirmzeile

QTSW	$CB—$CB	Flag für Hochkomma-
	203—203	mode ($00=aus, $0F=ein)
SEDT1	$CC—$CC	Länge der aktuellen
	204—204	Bildschirmzeile
TBLX	$CD—$CD	Cursor-
	205—205	zeile
DATAX	$CE—$CE	Zuletzt transferiertes
	206—206	Zeichen
INSTR	$CF—$CF	Insertmode:
	207—207	letztes Zeichen
	$D0—$D7	208—215 Frei für
	$D8—$E8	216—232 andere Zwecke.
CIRSEG	$E9—$E9	Arbeits-
	233—233	speicher
USER	$EA—$EB	Bildschirm-
	234—235	editor
KEYTAB	$EC—$ED	Tabulator
	236—237	speicher
TMPKEY	$EE—$EE	Arbeits-
	238—238	speicher
NDX	$EF—$EF	Zeichenzahl im
	239—239	Tastaturpuffer
STPFLG	$F0—$F0	CTRL-S Flag:
	240—240	ungleich 0: Prg. hält
TO	$F1—$F2	Hilfsregister für
	241—242	TEDMON
CHRPTR	$F3—$F3	Hilfs-
	243—243	speicher
BUFEND	$F4—$F4	Hilfs-
	244—244	speicher
CHKSOM	$F5—$F5	Prüfsummen-
	245—245	register
LENGTH	$F6—$F6	Hilfs-
	246—246	speicher
PASS	$F7—$F7	Hilfs-
	247—247	speicher
TYPE	$F8—$F8	Zwischen-
	248—248	speicher
USEKDY	$F9—$F9	Zwischen-
	249—249	speicher
XSTOP	$FA—$FA	Register für
	250—250	X- und STOP-Taste
CURBNK	$FB—$FB	Aktuelle Speicher-
	251—251	konfiguration
XON	$FC—$FC	Hilfsregister
	252—252	für RS232
XOFF	$FD—$FD	Hilfsregister
	253—253	für RS232
SEDT2	$FE—$FE	Register für
	254—254	Editor
LOFBUF	$FF—$FF	Zwischen-
	255—255	speicher

In der darauffolgenden Speicherseite von $0124 bis $01FF = dezimal 291 bis 512 legt der Prozessor seine Werte für JSR = Jump to Subroutine ab. Falls man von Basic aus in diesen Speicherbereich schreibt, stürzt der Computer mit hoher Wahrscheinlichkeit ab. Dieser Bereich ist dadurch für externe Zugriffe des Programmierers verboten. Seite 22 ist die genaue Dokumentation weiterer interessanter Speicherbereiche. Der TED (TExt-Display Chip) ist ab Seite 14 beschrieben.

WWW.HOMECOMPUTERWORLD.COM

SYS, PEEK und POKE im Commodore 64

Vielen Anwendern erscheint der 64er als ein Buch mit sieben Siegeln. Möchte man beispielsweise die Hintergrund- und Rahmenfarbe ändern, sind POKE-Anweisungen unumgänglich. Jeder POKE spricht eine bestimmte Speicherstelle an. POKE x,y schreibt in Speicherstelle x den Wert y. Die Farbspeicherstellen liegen in 53280 und 53281. Folglich muß man diese beiden Zahlen kennen, sollen Hintergrundfarben geändert werden. Ähnlich diesem Beispiel gibt es noch Dutzende weiterer POKE-Zahlen, die man kennen muß, um den Commodore 64 auszunutzen.

Speicherstelle: 43/44 /Basic-Anfang

Die beiden Basic-Speicheranfangszeiger geben an, in welchem Bereich freier Speicherplatz liegt.

FRE (0) = 3333

Da die Zahl größer als 256 ist, benötigt der 64er zwei Zahlen, die ihm diesen Wert mitteilen. Er errechnet sich aus Low-Byte + High-Byte * 256. Beispiel: In Speicher 43 steht hexadezimal 8. (Kurz $08 geschrieben). Der Inhalt des Speichers 44 beträgt $20 (hexadezimal 20). Der Wert, der sich daraus ergibt, ist 2 * 1024 + 8 = $ 2008. Wichtig sind diese Speicherstellen, wenn man ein Maschinenspracheprogramm in den Bereich von 2048 bis 4000 legen will und der Basic-Anfang deshalb 4001 betragen muß. (Ansonsten überschreibt ein Basic-Programm die Maschinenspracheroutine.) Teilt man 4001 in seine High- und Low-Bytezahlen auf, schreibt diese Werte in Speicherstellen 43/44 und POKEt in 4002 den Wert 0, dann beginnen Basic-Programme ab Adresse 4002. Das Maschinenspracheprogramm ist gegen Überschreiben durch Basic-Programme oder Variable geschützt.

Im Handbuch des C64 werden die Betriebssystemroutinen und Nullseitenspeicherstellen entweder ignoriert oder nur unzureichend beschrieben. RUN zeigt ausführlich, was die integrierten Maschinenroutinen können und wie man sie aus Basic bedient.

Speicherstelle: 45/46 /Basic-Ende

Dort steht (in Low- und High-Byte aufgespalten) die Endadresse des Basic-Speichers. Dieser Wert darf unter keinen Umständen unter dem Wert der Speicher-Anfangsadresse (43/44) liegen! Ein Unterschied von 1000 oder mehr Bytes ist sinnvoll. Die Werte nach Einschalten sind:
(43/44): 2048
(45/46): 40959

Wo ist das File?

Speicherstellen: 144/182 /Floppy

Vor allem zur Fehlersuche werden PEEK (144) und PEEK (182) genutzt. Normalerweise steht in 144 der Wert 0. Er zeigt dem Commodore-Betriebssystem, daß der letzte Schreib-/Leseversuch des Laufwerks erfolgreich war. Steht in 144 eine andere Zahl, gab es mit der Diskettenstation Probleme. In Speicherstelle 182 steht, wie oft die Floppy versuchte, auf die Diskette zuzugreifen.

Speicherstelle: 203 /Tastatur

Um herauszufinden, welchen Tastaturcode zum Beispiel die „Cursor links"-Taste besitzt, gibt man mit Befehl PRINT PEEK(203) die aktuell gedrückte Taste am Bildschirm aus.
(10 PRINT PEEK(203): GOTO 10)
Drückt man eine beliebige Taste, erscheint deren Wert am Bildschirm. Spieleprogrammierer nutzen PEEK(203) aus, um auf eine bestimmte Taste zu warten und, falls diese nicht betätigt wurde, das Programm weiterzuführen.

Repeat auf „P"

Beispiel:
10 Ist 203 gleich Leertaste?
20 Wenn ja, dann Schuß abfeuern
30 Wenn nein, dann fremde Raumschiffe weiterbewegen.
40 Wieder zu Zeile 10 springen

Speicherstelle: 650 /Tastenrepeat

Beispiel:
„ = = = = = = = = = = = = = = = = =
Um den oben gezeigten String einzugeben, muß man 15 mal die „ = "-Taste betätigen — vorausgesetzt, Speicherstelle 650 steht auf 0; denn POKEt man dort den Wert 128 ein, erhalten alle Tasten des Tastaturfeldes eine REPEAT-Funktion.

Speicherstelle: 198/631 bis 640 Tastaturpuffer

Mit dem Tastaturpuffer lassen sich einige Kunststücke vollführen. Da er bis zu 10 Zeichen aufnimmt, die er als normale Eingabe ausführt, nachdem der Inhalt des Pufferzeigers (198) entsprechend verändert wurde, ist es ein leichtes, den Commdore 64 wie von Geisterhand Befehle auf den Bildschirm schreiben zu lassen.
Beispiel:

```
10 POKE 631,ASC (?)
   = ASCII-Wert „?"
20 POKE 632,ASC (")
   =34
30 POKE 633,ASC (H)
   = ASCII-Wert „H"
40 POKE 634,ASC (A)
   = ASCII-Wert „A"
50 POKE 635,ASC (L)
   = ASCII-Wert „L"
```


WWW.HOMECOMPUTERWORLD.COM

60 POKE 636,ASC (L)
= ASCII-Wert „L"
70 POKE 637,ASC (O)
= ASCII-Wert „O"
80 POKE 638,13
= Wert für (RETURN)
(Wert 13 muß immer am Abschluß stehen, damit die Anweisung sofort ausgeführt wird).
90 POKE 198,Anzahl der eingepokten Zeichen, in diesem Fall 8.
Diese Befehlssequenz printed „HALLO" auf den Bildschirm.

Speicherstelle: 774 /LIST-Schutz
Neugierigen Nasen erleben nach POKE 774,27 oder POKE 774,0 ihr blaues Wunder, falls sie ein Basic-Programm LISTen wollen. POKE 774,26 schaltet den LIST-Schutz wieder ab.

Speicherstellen: 808,818,819
SAVE-Schutz
Beispiel:
10000 POKE 808,225
10010 POKE 818,253
10020 POKE 819,253
10030 RETURN

Schreibt man diese Unterroutine in ein eigenes Basic-Programm und startet diese mit GOSUB 10000 gleich zum Programmbeginn, ist es nicht mehr möglich, das File abzuspeichern. Die erweiterte Programmversion daher unbedingt vor dem ersten RUN abspeichern!

Künstliche Errors

Weitere POKE's und SYS': POKE 120, 2:
Danach gibt es nur noch eins: C64 ausschalten.

POKE 776, 1: Ein Zeichenwirrwarr am Bildschirm entsteht.
POKE 777, 1: Einzige Chance: RESET.
SYS 55296 : Wie POKE 788,13.
SYS 57344 : Erzeugt Overflow Errors
SYS 64738 : System-RESET (Kaltstart)
SYS 64764 : RESET (Warmstart)

Diese Unterroutine ist oftmals die letzte Möglichkeit, einen zerPOKEten Commodore 64 wieder auf die Beine zu stellen.

SYS 45998 : Undef'd Function Error
SYS 46000 : „? Error" ausgeben
SYS 46001 : „Syntax Error" printen
SYS 46002 : Ausgabe eines Illegal Direct Errors.

Vorsicht bei SIMON's Basic. Diese Erweiterung verändert mehrere Nullseitenspeicherstellen und versetzt den Vektor der Fehlerausgabe. Dadurch sind die meisten der hier gezeigten POKE's bei SIMON's Basic unwirksam oder haben unvorhersehbare Reaktionen (Absturz) zur Folge.

F. Bastian

RUN-O-THEK — Abtippen überflüssig

Für alle, die sich mühsames Abtippen, aufwendiges Testen und umständliche Fehlersuche wegen Falscheingaben ersparen wollen, gibt es eine Lösung: RUN-O-THEK.

Alle Programme dieser Ausgabe sind auf zwei Disketten **unter** der **Bestellnummer 713** erhältlich. Und das **für** nur **37,50 Mark** für **beide Disks.** Einfach mit der Zahlkarte von Seite 3 auf dem Postamt einzahlen oder einen ausgefüllten und unterschriebenen Euro-Scheck an

CW-Publikationen
RUN-O-THEK
Postfach 40 04 29
8000 München 40

abschicken und schon haben Sie sich viel Zeit und Ärger erspart. Bitte vergessen Sie in keinem Fall die Adresse und Bestellnummer anzugeben, da sonst Ihr Auftrag nicht ordnungsgemäß erledigt werden kann.

WWW.HOMECOMPUTERWORLD.COM

Geschwindigkeit ist Trumpf

System: C64, Laufwerk
Typ: Sprache
Sprache: Assembler (ASM)

„Power Basic" erweitert den Commodore 64 um Grafik, Disketten- und Soundbefehle. Seine Arbeitsgeschwindigkeit deklassiert Programme wie Simons Basic oder ExBasic Level II.

Die Erweiterung beinhaltet Befehle zu den wichtigsten Bereichen:
— Grafik
— Sound
— Sprites
— Floppy
— Textdarstellung

Das Programm ist als kombinierter Hexdump-Basiclader abgedruckt. Wie man mit dieser Art Listings umgeht, steht auf Seite 8. Gibt man das Programm als Basiclader ein, so speichert man die Daten ab und startet den Lader mit RUN. Danach gibt man:

```
POKE 43, 0: POKE 44,128
POKE 45, 97: POKE 46,146
SAVE„NAME",8
```

ein, um das fertige Maschinenprogramm zu speichern. Benutzt man den Checker, so ist als Startadresse 8000 einzugeben. Um die Erweiterung zu starten, gibt man SYS 32768 ein.

Floppy-Speeder integriert

POWER BASIC beinhaltet einen Floppy-Speeder, der bis zu sechsmal schnelleres LOAD und VERIFY von Diskette ermöglicht. Ein Vergleich: Um 37K Programm (150 Blocks) zu laden, braucht das Original-DOS 1 Minute 36 Sekunden; POWER BASIC hat das Programm schon nach 17 Sekunden im Speicher. Der Speeder verzichtet auf die Ausgabe von 'SEARCHING' und 'LOADING', da der Bildschirm sowieso abgeschaltet wird. Beim Verifizieren von Programmen gibt das Programm weiterhin die Meldung 'OK' oder '?VERIFY ERROR' aus. Man muß jedoch darauf achten, daß beim Laden oder Verifizieren von Programmen ein eventuell am seriellen Port angeschlossener Drucker ausgeschaltet ist, da dieser sonst den Datentransfer von der Floppy zum Computer beeinträchtigt. Hat man vergessen, den Drucker vorher auszuschalten, so merkt man das daran, daß sich der Schreib-/Lesekopf keine Spur weiterbewegt oder daß der Bildschirm schwarz bleibt und die Floppy weiterläuft, obwohl sich der Computer schon längst hätte zurückmelden müssen (bei dem größtmöglichen Programm [37K] nach weniger als 20 Sekunden). In so einem Fall einfach den Drucker ausschalten; das Programm wird dann korrekt geladen. Beim Speichern von POWER BASIC Programmen wäre es sinnvoll, an den Programmnamen ein '.PB' anzuhängen, um Programme, die nur mit POWER BASIC laufen, von anderen Programmen zu unterscheiden.

$0800—$9bff —	Basicspeicher
$9c00—$9f85 —	Befehlserkennung
$a000—$bfff —	Basic im RAM
$c000—$caae —	Befehle von POWER BASIC
$cb00—$cbff —	Tabelle von POWER BASIC
$cc00—$cfff —	Farbram des Grafikbildschirms
$d000—$d0ff —	Tabelle von TURBOLOAD (im RAM)
$d100—$d4b5 —	TURBOLOAD (im RAM)
$e000—$ffff —	Bitmap des Grafikbildschirms (im RAM)

Tabelle 1: Speicheraufteilung nach dem Start von POWER BASIC

FRE (x)
Dieser Befehl wurde so verbessert, daß jetzt die korrekte Anzahl freier Bytes angezeigt wird.

GOSUB x
Die Zeilennummer x kann auch ein mathematischer Ausdruck sein (zum Beispiel GOSUB 100+2*INT(x/3)).

$0001 —	Prozessorregister
$0034 —	Stringspeicher
$0036 —	Hilfszeiger
$0038 —	Speichergrenze
$0304,$0305 —	Umwandlung in Tokens
$0306,$0307 —	Umwandlung in Schlüsselwörter
$0308,$0309 —	neue Befehle
$0314,$0315 —	Interrupt (IRQ) — nur bei WINDOW 1
$0316,$0317 —	Break Interrupt
$0318,$0319 —	nicht maskierbarer Interrupt (NMI)
$0330,$0331 —	Load-Vektor

Tabelle 2: Von POWER BASIC veränderte Speicherstellen

GOTO x
x darf, wie bei GOSUB, Variablen und Ausdrücke enthalten.

RESTORE x
Es kann eine Zeilennummer (auch ein mathematischer Ausdruck) angegeben werden, wo der Zeiger für READ stehen soll. Dies ist besonders bei der Programmierung von Musikstücken interessant, wenn eine Melodie öfters vorkommt, die Daten für diese Meldodie aber nur einmal im Programm vorkommen sollen.

AT x,y [wobei x=0—39;y=0—24]
Der Cursor wird an die x-te Spalte und die y-te Zeile am Bildschirm positioniert. Ein folgendes PRINT ▶

WWW.HOMECOMPUTERWORLD.COM

oder INPUT erfolgt an dieser Position.

TCOL a,b,c [wobei a,b,c=0—15]
Die Farben des Textbildschirms werden gesetzt, wobei a die Hintergrund-, b die Rahmen- und c die Zeichenfarbe ist (Farben auf Seite 61 im Commodore Handbuch).

OLD
Retten eines nach NEW oder einem Reset gelöschten Programms. Nach einem Reset kann POWER BASIC mit SYS 39936 neu gestartet werden.

GRAFIK:

HIRES
Umschalten auf Grafikmodus. Der Grafikbildschirm liegt von 57344 [$E000] bis 65535 [$FFFF] unter dem ROM.

TEXT
Umschalten auf Textmodus.

CLEAR
Löschen des Grafikbildschirms.

GCOL a,b [wobei a,b=0—15]
Setzen der Grafikfarben, wobei a die Punkt- und b die Hintergrundfarbe ist. Die Rahmenfarbe im Grafikmodus kann mit TCOL geändert werden, die Hintergrundfarbe jedoch nicht. Das Farbram liegt von 52224 [$CC00] bis 53247 [$CFFF] im Speicher.

PLOT x,y
[wobei x=0—319;y=0—199]
Zeichnet einen Grafikpunkt an die x-te Spalte und die y-te Zeile des Grafikbildschirms.

CPLOT x,y
Löscht einen Grafikpunkt.

LINE x1,y1,x2,y2 [wobei x1,x2=0—319;y1,y2=0—199]
Zeichnet eine Linie vom Punkt mit den Koordinaten x1,y1 zum Punkt mit den Koordinaten x2,y2. Bei diesem Befehl wurde besonders auf die Geschwindigkeit Wert gelegt, um auch komplexe Grafiken (wie zum Beispiel Vektor-Grafiken) darstellen zu können.

CLINE x1,y1,x2,y2
Löscht eine Linie vom Punkt x1,y1 zum Punkt x2,y2.

BOX x1,y1,x2,y2
Zeichnet einen Kasten, wobei x1,y1 und x2,y2 zwei Eckpunkte des Kastens sind, die sich schräg gegenüberliegen.

CBOX x1,y1,x2,y2
Löscht einen Kasten, der durch die Punkte x1,y1 und x2,y2 festgelegt ist.

BLOCK x1,y1,x2,y2
Zeichnet einen ausgefüllten Kasten der durch die Eckpunkte x1,y1 und x2,y2 festgelegt ist.

CBLOCK x1,y1,x2,y2
Löscht einen ausgefüllten Kasten mit den Eckpunkten x1,y1 und x2,y2 wieder.

WINDOW a
a=1: Textfenster wird eingeblendet
a=0: Textfenster wird ausgeblendet
Mit WINDOW 1 schaltet man das Textfenster ein, wobei direkt auf Grafikmodus umgeschaltet wird (ein vorheriges HIRES kann entfallen). Die Auflösung mit Textfenster beträgt 320×151 Punkte (x=0—319;y=0—150), wobei jedoch die y-Koordinate eines Punktes auch weiterhin Werte bis 199 annehmen kann. Die Grafik unter dem Textfenster bleibt erhalten. Mit WINDOW 0 wird das Textfenster ausgeschaltet und auf Textmodus umgeschaltet. Folgt direkt hinter WINDOW 0 ein HIRES, so bleibt der Grafikmodus eingeschaltet.

SOUND:

VOL a [wobei a=0—255]
Setzt die Lautstärke aller drei Stimmen (0=stumm. . . 15=laut). Werte, die größer als 15 sind, setzen den Filtermodus (vergleiche Handbuch, Seite 156, Reg. 24).

ADSR nr,a,d,s,r
[wobei nr=1—3;a,d,s,r=0—15]
Bestimmt den Klang eines Tones, wobei nr für Stimme, a für Attack (Anschlag), d für Decay (Abschwellen), s für Sustain (Halten) und r für Release (Ausklingen) steht. Werte für die einzelnen Stufen:
Attack: 0=hart. . . 15=weich
Decay: 0=hart. . . 15=weich
Sustain: 0=stumm. . . 15=laut
Release: 0=schnell. . . 15=langsam

FRE (x)
— korrekte Anzahl freier Bytes
GOSUB x
— x kann Variablen enthalten
GOTO x
— x kann Variablen enthalten
RESTORE x
— Angabe einer Zeile erlaubt
AT x,y
— positioniert den Cursor
TCOL a,b,c
— setzt die Farben im Textmodus
OLD
— rettet ein gelöschtes Programm
HIRES
— schaltet auf Grafikmodus
TEXT
— schaltet auf Textmodus
CLEAR
— löscht den Grafikbildschirm
INVERS
— invertiert den Grafikbildschirm
GCOL a,b
— setzt die Grafikfarben
PLOT x,y
— zeichnet einen Punkt
CPLOT x,y
— löscht einen Punkt
LINE x1,y1,x2,y2
— zeichnet eine Linie
CLINE x1,y1,x2,y2
— löscht eine Linie
BOX x1,y1,x2,y2
— zeichnet einen Kasten
CBOX x1,y1,x2,y2
— löscht einen Kasten
BLOCK x1,y1,x2,y2
— zeichnet einen vollen Kasten
CBLOCK x1,y1,x2,y2
— löscht einen vollen Kasten
WINDOW a
— Textfenster an/aus
VOL
— setzt die Lautstärke
ADSR nr,a,d,s,r
— setzt die Klangwerte
WAVE nr,a
— setzt die Wellenform
PLAY nr,a
— spielt einen Ton
WIDTH nr,a
— setzt das Tastverhältnis
ERR
— zeigt den Floppystatus an
DIR
— zeigt das Directory an
DISK „Befehl"
— sendet einen Befehl zur Floppy
BLOAD „Name"[,adr]
— lädt Daten von der Floppy
FORMAT „Name,ID"
— formatiert eine Diskette
SPOS nr,x,y
— positioniert ein Sprite
SPRITE nr,a,b,c,d,e,f
— setzt das Spriteaussehen
SOFF a
— schaltet Sprites aus
MULTI a,b
— setzt die Multicolorfarben

Tabelle 3: Alle neuen Befehle auf einen Blick

WWW.HOMECOMPUTERWORLD.COM

WAVE nr,a
[wobei nr=1—3;a=0—255]
Bestimmt die Wellenform der Stimme nr, wobei a=129 Rauschen, a=65 Rechteck-, a=33 Sägezahn- und a=17 Dreieckschwingung bedeutet.
PLAY nr,a
[wobei nr=1—3;a=0—65535]
Bestimmt die Höhe a eines Tones der Stimme nr. Die Werte für die einzelnen Notenwerte stehen im Handbuch auf Seite 158f unter 'Parameter').
WIDTH nr,a
[wobei nr=1—3;a=0—4095]
Bestimmt das Tastverhältnis a der Stimme nr (nur bei der Wellenform Rechteck).

SPRITES:

SPOS nr,x,y [wobei nr=0—7;x=0—511;y=0—255]
Positioniert das Sprite nr auf dem Bildschirm und schaltet es ein. Die Variablen x und y stehen für die Position am Bildschirm, wobei natürlich ein Sprite mit x=511 und y=255 nicht sichtbar wäre, da es damit außerhalb des sichtbaren Feldes auf dem Bildschirm läge.
SPRITE nr,a,b,c,d,e,f [wobei nr=0—7;a=0—255;b,d,e,f,=0—1;c=0—15]
Definiert das Aussehen eines Sprites:
a ist der Block, in dem sich die Spritedaten befinden (zum Beispiel für 704 — >704=11*64, also a=11)
b ist die Art des Sprites (0=normal, 1=multicolor)
c ist die Farbe des Sprites
d ist die Priorität des Sprites (0=vor dem Hintergrund, 1=hinter dem Hintergrund)
e ist die Vergrößerung in x-Richtung (0=normal, 1=vergrößert).
SOFF nr [wobei nr=0—8]
Schaltet das Sprite nr aus. SOFF 8 schaltet alle Sprites aus.
MULTI a,b [wobei a,b=0—15]
Setzt die Multicolorfarben aller Sprites.

FLOPPY:

Diskettenoperationen im Grafikmodus sollten grundsätzlich nur bei ausgeschaltetem Textfenster erfolgen, weil es sonst zu Störungen in der Befehlsausführung kommen kann!
ERR
Zeigt den Floppystatus in der Form 'Fehlernummer, Fehlermeldung, Spur, Sektor' an.
DIR
Listet das Inhaltsverzeichnis (Directory) einer Diskette auf dem Bildschirm, ohne ein vorhandenes Basicprogramm zu zerstören. Durch Drücken der <CTRL> Taste kann das Listen unterbrochen, durch Loslassen der <CTRL> Taste wieder fortgesetzt werden.
DISK „Befehl"
Sendet einen Befehl an die Floppy (ersetzt OPEN 1,8,15,„Befehl": CLOSE 1).
BLOAD „Name"[,Adresse]
Lädt ein Maschinenspracheprogramm oder ein Grafikbild mit bis zu sechsfacher Geschwindigkeit, ohne die Basiczeiger zu verändern (=> kann in Programmen verwendet werden). Durch Angeben einer Adresse wird das Programm an die angegebene Stelle geladen. So ist es möglich, selbsterstellte Bilder in den Grafikspeicher von POWER BASIC (57344) zu laden und anzuzeigen. Dadurch, daß BLOAD schneller als die Originalroutine lädt und in Basicprogrammen verwendet werden kann, wäre ein Hires-Adventure mit POWER BASIC denkbar. Liegt ein Ladefehler vor, wie zum Beispiel FILE NOT FOUND, so verzichtet BLOAD auf eine Fehlermeldung. Die Art des Fehlers kann jedoch über die Statusvariable ST abgerufen werden:
ST=0 : Ok (kein Fehler)
ST=66 : FILE NOT FOUND
ST=—128: DEVICE NOT PRESENT
In einem Programm wäre folgendes denkbar: BLOAD „Hallo":IF ST=—128 THEN ?"Bitte schalten Sie die Floppystation ein!"
FORMAT „Name,ID"
Formatiert eine Diskette in zirka 24 Sekunden! Der Name der Diskette darf maximal 16 Zeichen lang sein; die ID besteht aus zwei Zeichen. Der 'Bump' am Anfang des Formatierens wurde auf ein Minimum reduziert, um einem Verstellen des Schreib-/Lesekopfes vorzubeugen. Die Formatierroutine springt direkt nach der Befehlsübertragung an die Floppy wieder ins Basic, was ein Weiterarbeiten mit dem Computer ermöglicht, während die Floppy formatiert.
WICHTIG: Soll hinter dem Basic-Befehl THEN ein neuer POWER BASIC-Befehl folgen, so muß vorher ein Doppelpunkt stehen (zum Beispiel IF a THEN:HIRES)!

T. Starke

```
8000    10 data 78,a9,34,85,01,a9,73,85, c5
8008    20 data fb,a9,80,85,fc,a0,00,84, 25
8010    30 data fd,a9,d1,85,fe,a2,04,b1, 4e
8018    40 data fb,91,fd,c8,d0,f9,e6,fc, 5f
8020    50 data e6,fe,ca,d0,f2,a9,37,85, dd
8028    60 data 01,58,a9,29,85,fb,a9,84, d8
8030    70 data 85,fc,84,fd,a9,9c,85,fe, a5
8038    80 data a2,04,b1,fb,91,fd,c8,d0, e0
8040    90 data f9,e6,fc,e6,fe,ca,d0,f2, 3a
8048   100 data a9,af,85,fb,a9,87,85,fc, 2f
8050   110 data 84,fd,a9,c0,85,fe,a2,0b, 8f
8058   120 data b1,fb,91,fd,c8,d0,f9,e6, 55
8060   130 data fc,e6,fe,ca,d0,f2,4c,00, 74
8068   140 data 9c,ea,ea,ea,ea,ea,ea,ea, b2
8070   150 data ea,ea,ea,a9,36,85,01,58, 9c
8078   160 data 84,90,a5,ba,d0,03,4c,13, e5
8080   170 data f7,c9,03,f0,f9,b0,03,4c, 42
8088   180 data 33,f5,a4,b7,d0,06,4c,10, 31
8090   190 data f7,4c,04,f7,a6,b9,a9,60, ee
8098   200 data 85,b9,20,d5,f3,a5,ba,20, ff
80a0   210 data 09,ed,a5,b9,20,c7,ed,20, 2f
80a8   220 data 13,ee,85,ae,a5,90,4a,4a, 12
80b0   230 data b0,df,20,13,ee,85,af,8a, 6c
80b8   240 data d0,08,a5,c3,85,ae,a5,c4, 61
80c0   250 data 85,af,a9,fd,25,90,85,90, 0d
80c8   260 data a0,00,b1,bb,c9,24,d0,03, 06
80d0   270 data 4c,01,f5,ad,20,d0,85,b6, b2
80d8   280 data 8c,20,d0,a9,0b,8d,11,d0, 52
80e0   290 data 78,a9,34,85,01,20,ff,d3, 8b
80e8   300 data a0,36,84,01,58,a0,1b,8c, 34
80f0   310 data 11,d0,a4,b6,8c,20,d0,a6, c5
80f8   320 data ae,a4,af,60,a9,0b,8d,00, 83
```


WWW.HOMECOMPUTERWORLD.COM

```
8100   330 data dd,2c,00,dd,10,fb,a9,03, 90
8108   340 data 8d,00,dd,a2,07,ca,d0,fd, d8
8110   350 data a0,04,ad,00,dd,2a,2a,26, fe
8118   360 data bd,6a,26,bd,ea,88,d0,f2, fc
8120   370 data a5,bd,49,ff,60,84,fb,a9, 60
8128   380 data 36,85,01,a2,01,20,92,03, 81
8130   390 data e0,01,f0,4b,a9,34,85,01, 7b
8138   400 data a2,02,a5,fb,d0,02,a2,04, b5
8140   410 data ad,00,d0,d0,07,ee,01,d0, f8
8148   420 data ad,01,d0,2c,a9,00,85,fc, 80
8150   430 data a5,93,f0,10,bd,00,d0,d1, ac
8158   440 data ae,f0,0e,a9,10,05,90,85, 3a
8160   450 data 90,4c,78,03,bd,00,d0,91, b3
8168   460 data ae,e6,ae,d0,02,e6,af,e8, 8a
8170   470 data e4,fc,d0,dc,a9,01,85,fb, 0d
8178   480 data ad,00,d0,d0,aa,18,60,a9, f5
8180   490 data 1d,38,60,20,94,01,c9,01, 7f
8188   500 data f0,f8,a9,0b,8d,00,dd,2c, d3
8190   510 data 00,dd,10,fb,a9,03,8d,00, 6c
8198   520 data dd,a0,06,88,d0,fd,ad,00, 3b
81a0   530 data dd,2a,2a,26,bd,6a,26,bd, 72
81a8   540 data ea,ea,ad,00,dd,2a,2a,26, 63
81b0   550 data bd,6a,26,bd,ea,ea,ad,00, 69
81b8   560 data dd,2a,2a,26,bd,6a,26,bd, 72
81c0   570 data ea,ea,ad,00,dd,2a,2a,26, 63
81c8   580 data bd,6a,26,bd,a9,34,85,01, b5
81d0   590 data a5,bd,49,ff,9d,00,d0,a9, f6
81d8   600 data 36,85,01,e8,d0,ac,60,a9, a6
81e0   610 data 05,85,31,a5,12,85,16,45, 6c
81e8   620 data 13,45,06,45,07,85,1a,a5, 87
81f0   630 data 13,85,17,a5,06,85,18,a5, f6
81f8   640 data 07,85,19,20,34,f9,a2,c8, 91
8200   650 data 20,56,f5,50,fe,b8,ad,01, 60
8208   660 data 1c,d9,24,00,f0,06,ca,d0, 4a
8210   670 data ef,4c,51,f5,c8,c0,08,d0, 3f
8218   680 data ea,20,56,f5,50,fe,b8,ad, 88
8220   690 data 01,1c,91,30,c8,d0,f5,50, e2
8228   700 data fe,b8,ad,01,1c,99,ba,01, 2d
8230   710 data c8,c0,46,d0,f2,20,e0,f8, 37
8238   720 data 20,e9,f5,c5,3a,f0,03,4c, 6a
8240   730 data 02,f5,a2,00,8a,20,cc,03, e2
8248   740 data bd,00,05,85,3b,2c,00,18, 34
8250   750 data 10,fb,a9,10,8d,00,18,2c, 27
8258   760 data 00,18,30,fb,98,26,3b,2a, a1
8260   770 data 2a,26,3b,2a,2a,8d,00,18, 9b
8268   780 data 98,26,3b,2a,2a,26,3b,2a, 98
8270   790 data 2a,8d,00,18,98,26,3b,2a, 08
8278   800 data 2a,26,3b,2a,2a,8d,00,18, 9b
8280   810 data 98,26,3b,2a,2a,26,3b,2a, 98
8288   820 data 2a,8d,00,18,e8,d0,b9,ad, a9
8290   830 data 00,1c,09,08,8d,00,1c,ad, e1
8298   840 data 00,05,c5,18,f0,03,4c,9e, 42
82a0   850 data fd,85,06,ad,01,05,85,07, 4b
82a8   860 data 4c,04,03,85,3b,2c,00,18, be
82b0   870 data 10,fb,a9,10,8d,00,18,2c, 27
82b8   880 data 00,18,30,fb,a0,04,a9,00, 62
82c0   890 data 26,3b,2a,2a,26,3b,2a,2a, d6
82c8   900 data 8d,00,18,88,d0,f0,60,a5, b8
82d0   910 data 18,85,06,8d,00,05,a5,19, 94
82d8   920 data 85,07,8d,01,05,a9,e0,85, c0
82e0   930 data 00,a5,00,30,fc,c9,01,d0, 90
82e8   940 data 0f,ad,00,05,f0,0f,85,06, bd
82f0   950 data ad,01,05,85,07,4c,fe,03, e2
82f8   960 data a9,01,20,cc,03,4c,22,eb, eb
8300   970 data a0,55,b9,ff,ce,99,ff,cf, 69
8308   980 data 88,d0,f7,a0,40,b9,be,d3, 81
8310   990 data 99,ff,ce,88,d0,f7,a0,15, 44
8318  1000 data b9,a0,d4,99,3f,cf,88,d0, 7c
8320  1010 data f7,20,00,cf,a0,55,b9,ff, 0c
8328  1020 data cf,99,ff,ce,88,d0,f7,4c, c9
8330  1030 data 0b,01,a0,b2,b9,ff,d0,99, e9
8338  1040 data 0a,01,88,d0,f7,a0,ba,b9, 19
8340  1050 data b1,d1,99,33,03,88,d0,f7, 9a
8348  1060 data a9,36,85,01,58,20,40,cf, 9f
8350  1070 data a9,57,20,dd,ed,a9,07,20, c1
8358  1080 data dd,ed,a9,1c,20,dd,ed,a9, 04
8360  1090 data 01,20,dd,ed,a9,0f,20,dd, ae
8368  1100 data ed,20,fe,ed,78,a9,34,85, 5c
8370  1110 data 01,60,a0,90,b9,ff,ce,99, a0
8378  1120 data ff,cf,88,d0,f7,a0,90,b9, 16
8380  1130 data 25,d4,99,ff,ce,88,d0,f7, 0a
8388  1140 data 20,00,cf,a0,90,b9,ff,cf, 55
8390  1150 data 99,ff,ce,88,d0,f7,4c,34, d1
8398  1160 data 03,a9,36,85,01,58,a9,6c, f2
83a0  1170 data 85,fb,a9,d2,85,fc,84,fd, 72
83a8  1180 data a9,03,85,fe,a2,09,20,7b, 6b
83b0  1190 data cf,a9,57,20,dd,ed,a5,fd, f5
83b8  1200 data 20,dd,ed,a5,fe,20,dd,ed, 58
83c0  1210 data a9,23,20,dd,ed,a0,00,78, 9e
83c8  1220 data a9,34,85,01,b1,fb,48,a9, 4e
83d0  1230 data 36,85,01,58,68,20,dd,ed, 92
83d8  1240 data c8,c0,23,d0,ea,20,fe,ed, 35
83e0  1250 data a5,fb,69,23,85,fb,90,03, 07
83e8  1260 data e6,fc,18,a5,fd,69,23,85, 8f
83f0  1270 data fd,90,03,e6,fe,18,ca,d0, 69
83f8  1280 data b5,20,7b,cf,a9,45,20,dd, e8
8400  1290 data ed,a9,f0,20,dd,ed,a9,03, a9
8408  1300 data 20,dd,ed,20,fb,ed,78,a9, ed
8410  1310 data 34,85,01,60,a9,08,20,0c, 05
8418  1320 data ed,a9,6f,20,b9,ed,a9,4d, ba
8420  1330 data 20,dd,ed,a9,2d,20,dd,ed, 84
8428  1340 data 60,a9,60,8d,04,03,a9,9d, 97
8430  1350 data 8d,05,03,a9,25,8d,06,03, 7e
8438  1360 data a9,9d,8d,07,03,a9,00,8d, 60
8440  1370 data 08,03,a9,9d,8d,09,03,a9, ef
8448  1380 data 9c,85,34,85,36,85,38,a9, 07
8450  1390 data b4,8d,30,03,a9,9c,8d,31, bd
8458  1400 data 03,a9,be,8d,18,03,a9,9c, ad
8460  1410 data 8d,19,03,a9,d5,8d,16,03, 29
8468  1420 data a9,9c,8d,17,03,a9,a0,a2, db
8470  1430 data 21,a0,00,85,fb,84,fa,b1, bc
8478  1440 data fa,91,fa,c8,d0,f9,e6,fb, 5c
8480  1450 data ca,d0,f4,a9,91,8d,24,a0, cd
8488  1460 data a9,9c,8d,25,a0,a9,aa,8d, 88
8490  1470 data 9c,b3,a9,9c,8d,9d,b3,a9, f0
8498  1480 data 20,8d,a0,a8,a9,ae,8d,a1, 72
84a0  1490 data a8,a9,9c,8d,a2,a8,a9,36, 7b
84a8  1500 data 85,01,b9,2f,9f,20,d2,ff, 5e
84b0  1510 data c8,c0,4d,d0,f5,20,44,a6, f8
84b8  1520 data 4c,74,a4,d0,03,4c,1d,a8, d9
84c0  1530 data 20,8a,ad,20,f7,b7,20,13, 70
84c8  1540 data a6,38,a5,5f,e9,01,a4,60, de
84d0  1550 data 4c,24,a8,38,4c,49,bc,20, 36
84d8  1560 data 8a,ad,4c,f7,b7,85,93,78, 80
84e0  1570 data a9,34,85,01,4c,8d,d3,48, 42
84e8  1580 data 8a,48,98,48,a9,7f,8d,0d, 3b
84f0  1590 data dd,ac,0d,dd,30,1f,20,bc, d7
84f8  1600 data f6,20,e1,ff,d0,17,20,a3, ed
8500  1610 data fd,20,18,e5,a9,36,85,01, 06
8508  1620 data a9,31,8d,14,03,a9,ea,8d, a7
8510  1630 data 15,03,6c,02,a0,4c,72,fe, 43
8518  1640 data 00,00,00,00,00,00,00,00, 00
8520  1650 data 00,00,00,00,00,00,00,00, 00
8528  1660 data 00,20,73,00,c9,e0,b0,09, e3
8530  1670 data 20,79,00,20,ed,a7,4c,ae, 6b
8538  1680 data a7,18,29,1f,0a,a8,b9,f0, c4
8540  1690 data 9e,85,a7,b9,ef,9e,85,a6, 37
8548  1700 data 20,73,00,6c,a6,00,10,30, c8
8550  1710 data c9,ff,f0,2c,24,0f,30,28, ab
8558  1720 data 8d,f9,cb,c9,e0,90,24,18, f2
8560  1730 data 29,1f,aa,e8,84,49,a0,ff, a8
8568  1740 data c8,b9,66,9e,10,fa,ca,d0, e8
8570  1750 data f7,c8,b9,66,9e,30,06,20, d6
8578  1760 data 47,ab,4c,48,9d,4c,ef,a6, 3a
8580  1770 data 4c,f3,a6,ad,f9,cb,4c,24, a9
8588  1780 data a7,20,fe,9d,a6,7a,a0,04, af
8590  1790 data 84,0f,bd,00,02,8d,f9,cb, d4
8598  1800 data c9,e0,90,04,18,4c,bb,9d, 5f
```


WWW.HOMECOMPUTERWORLD.COM

```
85a0  1810 data 10,07,c9,ff,f0,3e,e8,d0,  9c
85a8  1820 data e9,c9,20,f0,37,85,08,c9,  42
85b0  1830 data 22,f0,56,24,0f,70,2d,c9,  68
85b8  1840 data 3f,d0,04,a9,99,d0,25,c9,  f1
85c0  1850 data 30,90,04,c9,3c,90,1d,84,  22
85c8  1860 data 71,a0,00,84,0b,88,86,7a,  dd
85d0  1870 data ca,c8,e8,bd,00,02,38,f9,  6a
85d8  1880 data 9e,a0,f0,f5,c9,80,d0,30,  e1
85e0  1890 data 05,0b,a4,71,e8,c8,99,fb,  eb
85e8  1900 data 01,b9,fb,01,f0,36,38,e9,  4a
85f0  1910 data 3a,f0,04,c9,49,d0,02,85,  7d
85f8  1920 data 0f,38,e9,55,d0,94,85,08,  23
8600  1930 data bd,00,02,f0,df,c5,08,f0,  02
8608  1940 data db,c8,99,fb,01,e8,d0,f0,  ab
8610  1950 data a6,7a,e6,0b,c8,b9,9d,a0,  12
8618  1960 data 10,fa,b9,9e,a0,d0,b4,bd,  f9
8620  1970 data 00,02,10,be,4c,09,a6,a6,  93
8628  1980 data 7a,ca,e8,8e,f2,cb,a2,ff,  d4
8630  1990 data 8e,fb,cb,8e,fa,cb,ee,fa,  f3
8638  2000 data cb,ae,fa,cb,bd,66,9e,f0,  50
8640  2010 data 21,10,f3,ee,fb,cb,ac,f2,  00
8648  2020 data cb,88,c8,b9,00,02,8d,f9,  e4
8650  2030 data cb,e8,bd,66,9e,38,ed,f9,  93
8658  2040 data cb,f0,ef,c9,80,f0,13,4c,  59
8660  2050 data 0d,9e,ae,f2,cb,bd,00,02,  38
8668  2060 data f0,07,c9,3a,f0,bc,e8,d0,  c2
8670  2070 data f4,60,ae,f2,cb,ad,fb,cb,  9d
8678  2080 data 18,69,e0,9d,00,02,e8,c8,  10
8680  2090 data b9,00,02,9d,00,02,f0,da,  31
8688  2100 data e8,c8,d0,f4,4c,39,9e,ff,  ae
8690  2110 data 41,d4,54,43,4f,cc,4f,4c,  05
8698  2120 data c4,48,49,52,45,d3,54,45,  f4
86a0  2130 data 58,d4,43,4c,45,41,d2,49,  08
86a8  2140 data 4e,56,45,52,d3,47,43,4f,  6b
86b0  2150 data cc,50,4c,4f,d4,43,50,4c,  0d
86b8  2160 data 4f,d4,4c,49,4e,c5,43,4c,  ff
86c0  2170 data 49,4e,c5,42,4f,d8,43,42,  f6
86c8  2180 data 4f,d8,42,4c,4f,43,cb,43,  01
86d0  2190 data 42,4c,4f,43,cb,57,49,4e,  71
86d8  2200 data 44,4f,d7,56,4f,cc,41,44,  f6
86e0  2210 data 53,d2,57,41,56,c5,50,4c,  2d
86e8  2220 data 41,d9,45,52,d2,44,49,d2,  60
86f0  2230 data 44,49,53,cb,46,4f,52,4d,  80
86f8  2240 data 41,d4,42,4c,4f,41,c4,53,  e2
8700  2250 data 50,4f,d3,53,50,52,49,54,  74
8708  2260 data c5,57,49,44,54,c8,53,4f,  03
8710  2270 data 46,c6,4d,55,4c,54,c9,00,  39
8718  2280 data ce,c6,ed,c6,14,c7,00,c0,  bd
8720  2290 data 17,c0,2e,c0,46,c0,6e,c0,  fb
8728  2300 data ac,c0,a9,c0,61,c1,5e,c1,  13
8730  2310 data 19,c3,16,c3,98,c3,95,c3,  51
8738  2320 data 27,ca,60,c7,69,c7,dc,c7,  ae
8740  2330 data 10,c8,a7,c4,1a,c4,87,c4,  45
8748  2340 data c1,c4,94,c8,c4,c8,1b,c9,  18
8750  2350 data 50,c8,e3,c9,0d,ca,08,af,  40
8758  2360 data 93,0d,20,20,20,20,20,2a,  7c
8760  2370 data 2a,2a,2a,20,43,36,34,20,  2b
8768  2380 data 50,4f,57,45,52,20,42,41,  46
8770  2390 data 53,49,43,20,56,31,2e,30,  50
8778  2400 data 20,2a,2a,2a,2a,0d,0d,20,  00
8780  2410 data 20,20,28,57,29,20,42,59,  c3
8788  2420 data 20,54,2e,53,54,41,52,4b,  c1
8790  2430 data 45,20,2d,20,33,37,38,38,  2e
8798  2440 data 37,20,42,59,54,45,53,20,  42
87a0  2450 data 46,52,45,45,0d,0d,45,53,  e6
87a8  2460 data 20,46,52,45,45,0d,0d,a9,  83
87b0  2470 data 94,8d,00,dd,a9,04,8d,88,  d4
87b8  2480 data 02,a9,38,8d,18,d0,a9,3b,  bc
87c0  2490 data 8d,11,d0,4c,ae,a7,a9,97,  18
87c8  2500 data 8d,00,dd,a9,04,8d,88,02,  be
87d0  2510 data a9,15,8d,18,d0,a9,1b,8d,  bd
87d8  2520 data 11,d0,4c,ae,a7,a2,20,a9,  5c
87e0  2530 data e0,a0,00,85,fb,84,fa,98,  93
87e8  2540 data 91,fa,c8,d0,fb,e6,fb,ca,  d5
87f0  2550 data d0,f6,4c,ae,a7,a2,20,a9,  f5
87f8  2560 data e0,a0,00,85,fb,84,fa,78,  b3
8800  2570 data a9,35,85,01,b1,fa,49,ff,  f9
8808  2580 data 91,fa,c8,d0,f7,e6,fb,ca,  d1
8810  2590 data d0,f2,a9,36,85,01,58,4c,  e0
8818  2600 data ae,a7,01,00,ea,20,9e,b7,  b8
8820  2610 data e0,10,b0,31,8e,6b,c0,20,  10
8828  2620 data fd,ae,20,9e,b7,e0,10,b0,  09
8830  2630 data 24,8e,6c,c0,a2,04,a9,cc,  bd
8838  2640 data a0,00,85,fb,84,fa,ad,6b,  f6
8840  2650 data c0,0a,0a,0a,0a,0a,0d,6c,c0,  5e
8848  2660 data 91,fa,c8,d0,fb,e6,fb,ca,  d5
8850  2670 data d0,f6,4c,ae,a7,4c,48,b2,  69
8858  2680 data a9,80,2c,a9,00,85,97,20,  9f
8860  2690 data eb,b7,e0,c8,b0,ef,a5,15,  9d
8868  2700 data c9,01,90,08,d0,e7,a5,14,  c9
8870  2710 data c9,40,b0,e1,8a,4a,4a,4a,  98
8878  2720 data 0a,a8,b9,23,c1,85,fa,b9,  74
8880  2730 data 24,c1,85,fb,8a,29,07,18,  3e
8888  2740 data 65,fa,85,fa,a5,14,29,f8,  b9
8890  2750 data 85,fc,18,a9,00,65,fa,85,  09
8898  2760 data fd,a9,e0,65,fb,85,fe,18,  29
88a0  2770 data a5,fd,65,fc,85,fd,a5,fe,  41
88a8  2780 data 65,15,85,fe,a5,14,29,07,  8a
88b0  2790 data 49,07,aa,bd,55,c1,a0,00,  63
88b8  2800 data 78,a2,35,86,01,24,97,10,  ea
88c0  2810 data 05,49,ff,31,fd,2c,11,fd,  6e
88c8  2820 data 91,fd,a2,36,86,01,58,4c,  91
88d0  2830 data ae,a7,00,00,40,01,80,02,  c4
88d8  2840 data c0,03,00,05,40,06,80,07,  6a
88e0  2850 data c0,08,00,0a,40,0b,80,0c,  56
88e8  2860 data c0,0d,00,0f,40,10,80,11,  42
88f0  2870 data c0,12,00,14,40,15,80,16,  2e
88f8  2880 data c0,17,00,19,40,1a,80,1b,  1a
8900  2890 data c0,1c,00,1e,01,02,04,08,  81
8908  2900 data 10,20,40,80,00,a9,00,2c,  dc
8910  2910 data a9,80,85,97,20,7a,c1,20,  5e
8918  2920 data 95,c1,20,e3,c1,20,77,c1,  68
8920  2930 data 20,63,c2,4c,ae,a7,20,fd,  5d
8928  2940 data ae,20,eb,b7,8a,a8,a6,15,  34
8930  2950 data c0,c8,b0,0d,a5,14,e0,01,  09
8938  2960 data 90,06,d0,05,c9,40,b0,01,  8b
8940  2970 data 60,4c,48,b2,8c,1c,cb,8d,  58
8948  2980 data 1a,cb,8e,1b,cb,85,14,86,  96
8950  2990 data 15,98,4a,4a,4a,aa,bd,f0,  eb
8958  3000 data c2,85,ad,8a,29,03,aa,bd,  72
8960  3010 data 0a,c3,85,ac,98,29,07,18,  7f
8968  3020 data 65,ac,85,ac,a5,14,29,f8,  54
8970  3030 data 85,63,a9,e0,05,ad,85,ad,  1c
8978  3040 data 18,a5,ac,65,63,85,ac,a5,  9f
8980  3050 data ad,65,15,85,ad,a5,14,29,  cc
8988  3060 data 07,49,07,aa,bd,0e,c3,85,  08
8990  3070 data ab,60,08,78,a0,35,84,01,  c9
8998  3080 data a0,00,a5,ab,24,97,30,05,  52
89a0  3090 data 49,ff,31,ac,2c,11,ac,91,  06
89a8  3100 data ac,a0,36,84,01,58,28,a4,  ec
89b0  3110 data 6f,60,a5,ac,29,07,c9,07,  ec
89b8  3120 data f0,05,38,a9,00,b0,04,a9,  26
89c0  3130 data 38,e6,ad,65,ac,85,ac,a9,  c4
89c8  3140 data 00,65,ad,85,ad,60,30,e2,  5e
89d0  3150 data a5,ac,29,07,f0,05,18,a9,  74
89d8  3160 data ff,90,04,a9,c7,c6,ad,65,  13
89e0  3170 data ac,85,ac,a5,ad,e9,00,85,  6e
89e8  3180 data ad,60,46,ab,90,0e,66,ab,  25
89f0  3190 data a5,ac,c8,18,69,08,85,ac,  e2
89f8  3200 data 90,02,e6,ad,60,10,eb,06,  fa
8a00  3210 data ab,90,0e,26,ab,a5,ac,88,  2d
8a08  3220 data 38,e9,08,85,ac,b0,02,c6,  0c
8a10  3230 data ad,60,48,ad,1b,cb,4a,ad,  d6
8a18  3240 data 1a,cb,6a,4a,4a,85,6f,68,  3c
8a20  3250 data 48,38,ed,1a,cb,48,8a,ed,  02
8a28  3260 data 1b,cb,85,6c,b0,0a,68,49,  2e
```


WWW.HOMECOMPUTERWORLD.COM

```
8a30  3270 data ff,69,01,48,a9,00,e5,6c,  70
8a38  3280 data 85,6a,85,6e,68,85,69,85,  f9
8a40  3290 data 6d,68,8d,1a,cb,8e,1b,cb,  05
8a48  3300 data 98,18,ed,1c,cb,90,04,49,  46
8a50  3310 data ff,69,fe,85,6b,8c,1c,cb,  3e
8a58  3320 data 66,6c,38,e5,69,aa,a9,ff,  b7
8a60  3330 data e5,6a,85,70,a4,6f,b0,05,  6f
8a68  3340 data 0a,20,4e,c2,38,a5,6d,65,  12
8a70  3350 data 6b,85,6d,a5,6e,e9,00,85,  af
8a78  3360 data 6e,84,6f,20,e3,c1,e8,d0,  72
8a80  3370 data 05,e6,70,d0,01,60,a5,6c,  9a
8a88  3380 data b0,de,20,1f,c2,18,a5,6d,  b4
8a90  3390 data 65,69,85,6d,a5,6e,65,6a,  46
8a98  3400 data 50,dd,65,6a,50,43,00,00,  7c
8aa0  3410 data 01,02,03,05,06,07,08,0a,  fa
8aa8  3420 data 0b,0c,0d,0f,10,11,12,14,  fa
8ab0  3430 data 15,16,17,19,1a,1b,1c,1e,  fa
8ab8  3440 data 1f,00,40,80,c0,01,02,04,  9c
8ac0  3450 data 08,10,20,40,80,a9,00,2c,  84
8ac8  3460 data a9,80,85,97,20,7a,c1,8c,  f2
8ad0  3470 data 20,cb,8d,21,cb,8e,22,cb,  55
8ad8  3480 data 20,95,c1,20,e3,c1,20,77,  f7
8ae0  3490 data c1,8c,23,cb,8d,24,cb,8e,  33
8ae8  3500 data 25,cb,ac,20,cb,20,63,c2,  32
8af0  3510 data ac,20,cb,ad,24,cb,ae,25,  8c
8af8  3520 data cb,20,95,c1,20,e3,c1,ac,  d1
8b00  3530 data 23,cb,ad,24,cb,ae,25,cb,  58
8b08  3540 data 20,63,c2,ac,23,cb,ad,24,  b5
8b10  3550 data cb,ae,25,cb,20,95,c1,20,  a4
8b18  3560 data e3,c1,ac,23,cb,ad,21,cb,  1f
8b20  3570 data ae,22,cb,20,63,c2,ac,23,  60
8b28  3580 data cb,ad,21,cb,ae,22,cb,20,  ab
8b30  3590 data 95,c1,20,e3,c1,ac,20,cb,  7c
8b38  3600 data ad,21,cb,ae,22,cb,20,63,  bd
8b40  3610 data c2,4c,ae,a7,a9,00,2c,a9,  a8
8b48  3620 data 80,85,97,20,7a,c1,8c,20,  97
8b50  3630 data cb,8d,21,cb,8e,22,cb,20,  ab
8b58  3640 data 95,c1,20,e3,c1,20,77,c1,  68
8b60  3650 data 8c,23,cb,8d,24,cb,8e,25,  69
8b68  3660 data cb,ad,20,cb,cd,23,cb,b0,  38
8b70  3670 data 2c,ac,20,cb,ad,21,cb,ae,  7e
8b78  3680 data 22,cb,20,95,c1,20,e3,c1,  a5
8b80  3690 data ac,20,cb,ad,24,cb,ae,25,  8c
8b88  3700 data cb,20,63,c2,ad,20,cb,cd,  d6
8b90  3710 data 23,cb,d0,03,4c,ae,a7,ee,  7c
8b98  3720 data 20,cb,4c,c2,c3,ac,20,cb,  4c
8ba0  3730 data ad,21,cb,ae,22,cb,20,95,  8b
8ba8  3740 data c1,20,e3,c1,ac,20,cb,ad,  6b
8bb0  3750 data 24,cb,ae,25,cb,20,63,c2,  2e
8bb8  3760 data ad,20,cb,cd,23,cb,d0,03,  b0
8bc0  3770 data 4c,ae,a7,ce,20,cb,4c,ee,  2b
8bc8  3780 data c3,a9,01,a2,fd,a0,00,20,  b7
8bd0  3790 data f9,fd,a2,08,a0,60,20,00,  f6
8bd8  3800 data fe,a9,24,85,fd,20,d5,f3,  b2
8be0  3810 data a9,08,20,09,ed,a9,60,20,  3b
8be8  3820 data c7,ed,a9,00,85,90,a0,03,  14
8bf0  3830 data 84,fd,20,13,ee,85,fe,a4,  57
8bf8  3840 data 90,d0,35,20,13,ee,a4,90,  0f
8c00  3850 data d0,2e,a4,fd,88,d0,e9,a6,  44
8c08  3860 data fe,20,cd,bd,a9,20,20,d2,  c4
8c10  3870 data ff,20,13,ee,a6,90,d0,18,  d2
8c18  3880 data aa,f0,06,20,d2,ff,4c,62,  5e
8c20  3890 data c4,a9,0d,20,d2,ff,a4,91,  ee
8c28  3900 data c0,fb,f0,fa,a0,02,d0,c0,  68
8c30  3910 data 20,42,f6,4c,ae,a7,a9,01,  36
8c38  3920 data a2,08,a0,0f,20,00,fe,a9,  9e
8c40  3930 data 00,85,b7,20,c0,ff,a2,01,  74
8c48  3940 data 20,90,aa,20,b5,ab,a9,01,  cc
8c50  3950 data 20,c3,ff,4c,ae,a7,a9,08,  b8
8c58  3960 data 20,09,ed,a9,6f,20,c7,ed,  83
8c60  3970 data 20,13,ee,20,d2,ff,c9,0d,  69
8c68  3980 data d0,f6,20,ef,ed,4c,ae,a7,  b3
8c70  3990 data 20,57,e2,a5,b7,c9,14,90,  78
8c78  4000 data 03,4c,08,af,a0,00,b1,bb,  a6
8c80  4010 data 99,bb,c6,c8,c4,b7,d0,f6,  c3
8c88  4020 data 8c,b9,c6,a0,00,b1,bb,c9,  3a
8c90  4030 data 2c,f0,08,c8,c0,11,d0,f5,  07
8c98  4040 data 4c,08,af,c8,8c,ba,c6,a9,  1a
8ca0  4050 data 7b,85,fb,a9,c5,85,fc,a9,  da
8ca8  4060 data 00,85,fd,a9,05,85,fe,a2,  ab
8cb0  4070 data 0a,a9,08,20,0c,ed,a9,6f,  a3
8cb8  4080 data 20,b9,ed,a9,4d,20,dd,ed,  c8
8cc0  4090 data a9,2d,20,dd,ed,a9,57,20,  3a
8cc8  4100 data dd,ed,a5,fd,20,dd,ed,a5,  24
8cd0  4110 data fe,20,dd,ed,a9,23,20,dd,  96
8cd8  4120 data ed,a0,00,b1,fb,20,dd,ed,  66
8ce0  4130 data c8,c0,23,d0,f6,20,fe,ed,  41
8ce8  4140 data a5,fb,69,23,85,fb,90,03,  07
8cf0  4150 data e6,fc,18,a5,fd,69,23,85,  8f
8cf8  4160 data fd,90,03,e6,fe,18,ca,d0,  69
8d00  4170 data b0,a9,08,20,0c,ed,a9,6f,  49
8d08  4180 data 20,b9,ed,a9,4d,20,dd,ed,  c8
8d10  4190 data a9,2d,20,dd,ed,a9,45,20,  28
8d18  4200 data dd,ed,a9,e8,20,dd,ed,a9,  39
8d20  4210 data 05,20,dd,ed,20,fb,ed,4c,  9c
8d28  4220 data ae,a7,a5,0a,20,4b,f2,85,  e3
8d30  4230 data 43,a0,00,84,1b,a2,00,a5,  f5
8d38  4240 data 39,99,00,03,a5,1b,99,02,  be
8d40  4250 data 03,a5,0a,99,03,03,a5,13,  62
8d48  4260 data 99,04,03,a5,12,99,05,03,  6f
8d50  4270 data a9,0f,99,06,03,99,07,03,  9b
8d58  4280 data a5,1b,45,0a,45,12,45,13,  29
8d60  4290 data 99,01,03,98,69,08,a8,e6,  25
8d68  4300 data 1b,a5,1b,c5,43,90,c8,a9,  9f
8d70  4310 data 03,85,31,98,48,8a,9d,00,  73
8d78  4320 data 07,e8,d0,fa,20,30,fe,68,  7c
8d80  4330 data a8,88,20,e5,fd,20,f5,fd,  2f
8d88  4340 data a9,07,85,31,20,e9,f5,85,  9c
8d90  4350 data 3a,20,8f,f7,a9,00,85,32,  ae
8d98  4360 data 20,0e,fe,a9,ff,8d,01,1c,  be
8da0  4370 data a2,05,50,fe,b8,ca,d0,fa,  b3
8da8  4380 data a2,0a,a4,32,50,fe,b8,b9,  5a
8db0  4390 data 00,03,8d,01,1c,c8,ca,d0,  d7
8db8  4400 data f3,a2,09,50,fe,b8,a9,55,  a4
8dc0  4410 data 8d,01,1c,ca,d0,f5,a9,ff,  63
8dc8  4420 data a2,05,50,fe,b8,8d,01,1c,  ff
8dd0  4430 data ca,d0,f7,a2,bb,50,fe,b8,  ff
8dd8  4440 data bd,00,01,8d,01,1c,e8,d0,  2e
8de0  4450 data f4,a0,00,50,fe,b8,b1,30,  cb
8de8  4460 data 8d,01,1c,c8,d0,f5,a9,55,  0f
8df0  4470 data a2,08,50,fe,b8,8d,01,1c,  fc
8df8  4480 data ca,d0,f7,a5,32,18,69,0a,  c5
8e00  4490 data 85,32,c6,1b,d0,95,50,fe,  8a
8e08  4500 data b8,50,fe,b8,20,00,fe,4c,  7f
8e10  4510 data 9e,fd,a0,00,b9,40,06,99,  27
8e18  4520 data 00,02,c8,cc,3e,06,d0,f4,  0d
8e20  4530 data 8c,74,02,ac,3f,06,8c,7b,  b8
8e28  4540 data 02,a9,00,85,7f,20,00,c1,  73
8e30  4550 data b9,00,02,85,12,b9,01,02,  8f
8e38  4560 data 85,13,20,07,d3,a9,1a,8d,  42
8e40  4570 data 05,1c,a9,0b,8d,07,1c,a9,  7f
8e48  4580 data c0,85,00,a5,00,30,fc,a9,  b9
8e50  4590 data 0f,8d,07,1c,a2,01,86,0a,  8a
8e58  4600 data a9,e0,85,02,a5,02,30,fc,  22
8e60  4610 data e8,e0,24,d0,f1,4c,40,ee,  53
8e68  4620 data 00,00,00,00,00,00,00,00,  00
8e70  4630 data 00,00,00,00,00,00,00,00,  00
8e78  4640 data 00,00,00,00,00,20,9e,b7,  c7
8e80  4650 data e0,28,b0,15,8a,48,20,fd,  b7
8e88  4660 data ae,20,9e,b7,e0,19,b0,09,  e2
8e90  4670 data 68,a8,18,20,f0,ff,4c,ae,  48
8e98  4680 data a7,4c,48,b2,20,9e,b7,e0,  4a
8ea0  4690 data 10,b0,f6,8e,21,d0,20,fd,  3d
8ea8  4700 data ae,20,9e,b7,e0,10,b0,e9,  0b
8eb0  4710 data 8e,20,d0,20,fd,ae,20,9e,  ee
8eb8  4720 data b7,e0,10,b0,dc,8e,86,02,  09
```


WWW.HOMECOMPUTERWORLD.COM

```
8ec0  4730 data 4c,ae,a7,a5,2b,18,69,04, 18
8ec8  4740 data 85,fd,a5,2c,85,fe,a0,00, 28
8ed0  4750 data b1,fd,f0,08,c8,c0,58,d0, 2c
8ed8  4760 data f7,4c,08,af,c8,98,a0,00, d4
8ee0  4770 data 18,65,fd,91,2b,85,fd,90, 32
8ee8  4780 data 02,e6,fe,a5,fe,c8,91,2b, 11
8ef0  4790 data 88,b1,fd,aa,c8,b1,fd,f0, 4d
8ef8  4800 data 07,85,fe,86,fd,38,b0,f0, 7e
8f00  4810 data a5,fd,18,69,02,85,2d,a5, 5d
8f08  4820 data fe,20,55,a6,4c,86,e3,20, 15
8f10  4830 data 9e,b7,8e,18,d4,4c,ae,a7, eb
8f18  4840 data 20,9e,b7,e0,01,f0,45,e0, d1
8f20  4850 data 02,f0,4e,e0,03,f0,57,4c, a1
8f28  4860 data 48,b2,20,fd,ae,20,9e,b7, 2f
8f30  4870 data e0,10,b0,f3,60,20,7b,c7, 80
8f38  4880 data 8a,0a,0a,0a,0a,48,20,7b, e7
8f40  4890 data c7,8e,0d,03,68,0d,0d,03, a8
8f48  4900 data 8d,0c,d4,20,7b,c7,8a,0a, 68
8f50  4910 data 0a,0a,0a,48,20,7b,c7,8e, a0
8f58  4920 data 0d,03,68,0d,0d,03,8d,0d, ef
8f60  4930 data d4,4c,ae,a7,a9,05,8d,9a, 25
8f68  4940 data c7,a9,06,8d,b0,c7,4c,86, 47
8f70  4950 data c7,a9,0c,8d,9a,c7,a9,0d, 0c
8f78  4960 data 8d,b0,c7,4c,86,c7,a9,13, ad
8f80  4970 data 8d,9a,c7,a9,14,8d,b0,c7, 81
8f88  4980 data 4c,86,c7,20,9e,b7,e0,01, 32
8f90  4990 data f0,12,e0,02,f0,17,e0,03, 6f
8f98  5000 data f0,1c,4c,48,b2,20,fd,ae, b7
8fa0  5010 data 20,9e,b7,60,20,ee,c7,8e, 45
8fa8  5020 data 04,d4,4c,ae,a7,20,ee,c7, 7c
8fb0  5030 data 8e,0b,d4,4c,ae,a7,20,ee, 43
8fb8  5040 data c7,8e,12,d4,4c,ae,a7,20, 9d
8fc0  5050 data 9e,b7,e0,01,f0,15,e0,02, 7d
8fc8  5060 data f0,1d,e0,03,f0,25,4c,48, 7d
8fd0  5070 data b2,20,fd,ae,20,8a,ad,20, 03
8fd8  5080 data f7,b7,60,20,22,c8,8c,00, 66
8fe0  5090 data d4,8d,01,d4,4c,ae,a7,20, 9a
8fe8  5100 data 22,c8,8c,07,d4,8d,08,d4, 5a
8ff0  5110 data 4c,ae,a7,20,22,c8,8c,0e, fd
8ff8  5120 data d4,8d,0f,d4,4c,ae,a7,20, a8
9000  5130 data 9e,b7,e0,01,f0,19,e0,02, 79
9008  5140 data f0,21,e0,03,f0,29,4c,48, 75
9010  5150 data b2,20,fd,ae,20,8a,ad,20, 03
9018  5160 data f7,b7,c9,10,b0,f0,60,20, f8
9020  5170 data 62,c8,8c,02,d4,8d,03,d4, 9a
9028  5180 data 4c,ae,a7,20,62,c8,8c,09, 42
9030  5190 data d4,8d,0a,d4,4c,ae,a7,20, a3
9038  5200 data 62,c8,8c,10,d4,8d,11,d4, 9a
9040  5210 data 4c,ae,a7,20,57,e2,a9,08, 3b
9048  5220 data 85,ba,a0,00,b1,7a,c9,2c, 3e
9050  5230 data d0,13,20,73,00,20,8a,ad, 27
9058  5240 data 20,f7,b7,a6,14,a4,15,a9, 18
9060  5250 data 00,85,b9,f0,05,a9,01,85, 1e
9068  5260 data b9,98,20,d5,ff,4c,ae,a7, 26
9070  5270 data 4c,48,b2,20,9e,b7,e0,08, 54
9078  5280 data b0,f6,8a,a8,a9,01,e0,00, 23
9080  5290 data f0,04,0a,88,d0,fc,48,0d, 7d
9088  5300 data 15,d0,8d,15,d0,8a,0a,8d, 80
9090  5310 data 10,c9,69,01,8d,13,c9,20, d2
9098  5320 data fd,ae,20,eb,b7,a5,15,c9, e3
90a0  5330 data 00,f0,06,c9,01,f0,0e,d0, 9f
90a8  5340 data c7,68,49,ff,2d,10,d0,8d, 09
90b0  5350 data 10,d0,4c,0d,c9,68,0d,10, de
90b8  5360 data d0,8d,10,d0,a5,14,8d,00, a1
90c0  5370 data d0,8e,01,d0,4c,ae,a7,4c, 6d
90c8  5380 data 48,b2,20,9e,b7,e0,08,b0, 49
90d0  5390 data f6,8a,a8,a9,01,e0,00,f0, 9d
90d8  5400 data 04,0a,88,d0,fc,48,8a,18, d8
90e0  5410 data 48,69,f8,8d,44,c9,68,69, c4
90e8  5420 data 27,8d,74,c9,20,fd,ae,20, f7
90f0  5430 data 9e,b7,8e,f8,07,20,fd,ae, b3
90f8  5440 data 20,9e,b7,e0,02,b0,c8,e0, 94
9100  5450 data 01,f0,0d,68,48,49,ff,2d, 88
9108  5460 data 1c,d0,8d,1c,d0,4c,69,c9, e1
9110  5470 data 68,48,0d,1c,d0,8d,1c,d0, a0
9118  5480 data 20,fd,ae,20,9e,b7,e0,10, 68
9120  5490 data b0,a5,8e,27,d0,20,fd,ae, 6f
9128  5500 data 20,9e,b7,e0,02,b0,98,e0, 64
9130  5510 data 01,f0,0d,68,48,49,ff,2d, 88
9138  5520 data 1b,d0,8d,1b,d0,4c,99,c9, 11
9140  5530 data 68,48,0d,1b,d0,8d,1b,d0, a0
9148  5540 data 20,fd,ae,20,9e,b7,e0,02, 76
9150  5550 data b0,3d,e0,01,f0,0d,68,48, 53
9158  5560 data 49,ff,2d,1d,d0,8d,1d,d0, eb
9160  5570 data 4c,bc,c9,68,48,0d,1d,d0, 79
9168  5580 data 8d,1d,d0,20,fd,ae,20,9e, f0
9170  5590 data b7,e0,02,b0,1a,e0,01,f0, 76
9178  5600 data 0c,68,49,ff,2d,17,d0,8d, 48
9180  5610 data 17,d0,4c,ae,a7,68,0d,17, 1b
9188  5620 data d0,8d,17,d0,4c,ae,a7,4c, 84
9190  5630 data 48,b2,20,9e,b7,e0,09,b0, 4a
9198  5640 data f6,e0,08,f0,17,8a,a8,a9, bb
91a0  5650 data 01,e0,00,f0,04,0a,88,d0, e5
91a8  5660 data fc,49,ff,2d,15,d0,8d,15, 41
91b0  5670 data d0,4c,ae,a7,a9,00,8d,15, ab
91b8  5680 data d0,4c,ae,a7,20,9e,b7,e0, e4
91c0  5690 data 10,b0,cc,8e,25,d0,20,fd, 17
91c8  5700 data ae,20,9e,b7,e0,10,b0,bf, 35
91d0  5710 data 8e,26,d0,4c,ae,a7,20,9e, 74
91d8  5720 data b7,e0,02,b0,b2,e0,01,f0, 0e
91e0  5730 data 19,78,a9,31,8d,14,03,a9, ec
91e8  5740 data ea,8d,15,03,a9,00,8d,1a, 8a
91f0  5750 data d0,a9,81,8d,0d,dc,58,4c, 59
91f8  5760 data 17,c0,78,a9,6d,8d,14,03, 18
9200  5770 data a9,ca,8d,15,03,a9,7f,8d, a3
9208  5780 data 0d,dc,a9,01,8d,1a,d0,a9, 72
9210  5790 data 01,85,b6,a9,3b,8d,11,d0, 79
9218  5800 data 58,4c,ae,a7,78,ad,19,d0, 28
9220  5810 data 8d,19,d0,29,01,f0,2b,c6, 91
9228  5820 data b6,10,04,a9,01,85,b6,a6, 8d
9230  5830 data b6,bd,a7,ca,8d,11,d0,bd, 64
9238  5840 data a9,ca,8d,12,d0,bd,ab,ca, 4d
9240  5850 data 8d,18,d0,bd,ad,ca,8d,00, f8
9248  5860 data dd,8a,f0,06,68,a8,68,aa, ba
9250  5870 data 68,40,58,4c,31,ea,3b,1b, 9c
9258  5880 data c9,12,38,15,94,97,00,00, d7
9260  5890 data 00,00,00,00,00,00,00,00, 00
```

```
60000 for a=32768 to 37472 step    4279
      8:s=0:c=0:for b=0 to 7:
      gosub 60040
60010 s=s+h+1-c:c=s>255:s=255-s    5034
      and 255:poke a+b,h+1:next
      b:gosub 60040
60020 f$="pruefsumme ok":if s<>    5699
      h+1 then f$="[rvs]
      pruefsummenfehler":f=f+1
60030 gosub 60090:next a:print "   2267
      [down]"f"fehler":end
60040 read h$:gosub 60050:h=1*     2715
      16:h$=mid$ (h$,2)
60050 l=asc (h$+"x")-48:if l<0     3198
      or l>22 then 60080
60060 if l<10 then return          1044
60070 l=l-7:if l>9 then return     1334
60080 f$="[rvs]eingabefehler":l=   2460
      0:f=f+1
60090 print "zeile" peek (64)*    2726
      256+peek (63)f$:return
```

saven des programms nach starten des loaders mit

```
poke 43,  0:poke 44,128
poke 45, 97:poke 46,146
save "name",8
```

Der Porsche unter den Basicerweiterungen: „Power-Basic"

WWW.HOMECOMPUTERWORLD.COM

Datasette als Datensafe

System: C64/Floppy/Datasette
Typ: Utility
Sprache: ASM

Bis zu 202 Diskblocks umfassende Programme verarbeitet der Datasettenspeeder. Zu „Turbo-Tape" aufzeichnungskompatibel, liest er auch dessen Daten. Je nach Größe des Files greift Turbo-Tape ebenfalls auf „Gamer-Tape"-Dateien zu. Der Befehlssatz:

£L für LOAD (laden)
£S für SAVE (speichern)
£V für VERIFY (prüfen)

Kassetten eignen sich hervorragend als sicherer Datenspeicher. Dabei beschleunigt „Gamer-Tape" die Speicher-/Ladegeschwindigkeit um den Faktor 10 und der Bandverbrauch sinkt auf 10 Prozent.

Abgespeicherte Programme belegen zirka 1/10 der üblichen Kassettenbandlänge. Pro C60-Kassette speichert man durchschnittlich 11 bis 12 Disketten.

WICHTIG!!!

„Gamer-Tape" darf nur zum Kopieren zwischen Diskette und Kassette oder zum Laden von Kassette und anschließendem Starten benutzt werden. Ändert man ein Programm oder schreibt ein neues, speichert „Gamer-Tape" das File falsch ab.

softteam

```
  1 rem erstellt durch softteam
  5 poke 53280,0:poke 53281,0:print chr$      2086
    (5)
 10 print chr$ (147);"[7spaces]gamer -         2790
    tape spezial"
 12 print :print                                371
 15 print "basiert auf den routinen von       3546
    turbo tape"
 20 print "allerdings kann man mit turbo      3318
    tape "
 22 print "spezial auch programme mit mehr     3217
    als "
 24 print "152 blocks diskettengroesse         3724
    laden."
 30 print "(max. 202 blocks)"                  1987
 32 print                                       153
 34 print "allerdings erscheint dann immer     3606
    "
 36 print "nach ende des ladevorgangs"         2567
 38 print "'load error !', was sie aber        2921
    nicht"
 40 print "stoeren sollte."                    2153
 42 print                                       153
 43 print "benutzen sie bitte 'f' anstatt      3489
    '←' !!!"
 44 print "(z. bsp. 'f1' fuer 'load')."        3589
 45 print :print :print "bitte warten"          2091
 50 a=52224                                     641
 52 b=52974                                     914
100 for i=a to b:read q:poke i,q:next           1778
200 sys 52224                                   778
250 print chr$ (147);"fertig !"                1898
500 end                                         128
1000 data 169, 11, 141, 8, 3, 169, 204,        3587
     141, 9, 3, 96
1002 data 32, 115, 0, 240, 4, 201, 92, 240,    3458
     3, 76, 231
1004 data 167, 32, 115, 0, 201, 83, 240,       3822
     19, 201, 76, 240
1006 data 28, 201, 86, 240, 33, 201, 80,       3444
     240, 38, 201, 70
1008 data 240, 43, 76, 8, 175, 32, 115, 0,     3952
     32, 98, 204
1010 data 162, 55, 134, 1, 76, 174, 167,       4024
     32, 115, 0, 32
1012 data 90, 205, 76, 174, 167, 32, 115,      3720
     0, 32, 93, 205
1014 data 76, 174, 167, 32, 115, 0, 32,        3917
     129, 206, 76, 174
1016 data 167, 32, 115, 0, 32, 171, 206,       4293
     76, 174, 167, 162
1018 data 5, 134, 171, 32, 212, 225, 162,      4239
     54, 134, 1, 162
1020 data 4, 181, 42, 149, 171, 202, 208,      4218
     249, 32, 56, 248
1022 data 32, 143, 246, 32, 247, 204, 32,      4145
     11, 205, 165, 185
1024 data 24, 105, 1, 202, 32, 43, 205,        3491
     162, 8, 185, 172
1026 data 0, 32, 43, 205, 162, 6, 200, 192,    3963
     5, 234, 208
1028 data 242, 160, 0, 162, 4, 177, 187,       4341
     196, 183, 144, 3
1030 data 169, 32, 202, 32, 43, 205, 162,      3705
     5, 200, 192, 187
1032 data 208, 237, 169, 2, 133, 171, 32,      4844
     11, 205, 152, 32
1034 data 43, 205, 132, 215, 162, 7, 234,      4310
     177, 172, 32, 43
1036 data 205, 162, 3, 230, 172, 208, 4,       4049
     230, 173, 202, 208
1038 data 165, 172, 197, 174, 165, 173,        5987
     229, 175, 144, 231, 234
1040 data 165, 215, 32, 43, 205, 162, 7,       4214
     136, 208, 246, 200
1042 data 132, 192, 88, 24, 169, 0, 141,       4088
     160, 2, 162, 55
1044 data 134, 1, 76, 147, 252, 160, 0,        4091
     132, 192, 173, 17
1046 data 208, 41, 239, 141, 17, 208, 202,     3724
     208, 253, 136, 208
1048 data 250, 120, 96, 160, 0, 169, 2, 32,    4026
     43, 205, 162
1050 data 7, 136, 192, 9, 208, 244, 162, 5,    3979
     198, 171, 208
1052 data 238, 152, 32, 43, 205, 162, 7,       4077
     136, 208, 247, 202
1054 data 202, 96, 133, 189, 69, 215, 133,     4327
     215, 169, 8, 133
1056 data 163, 6, 189, 165, 1, 41, 247, 32,    4374
     77, 205, 162
1058 data 17, 234, 9, 8, 32, 77, 205, 162,     3606
     14, 198, 163
1060 data 208, 233, 96, 202, 208, 253, 144,    4354
     5, 162, 11, 202
1062 data 208, 253, 133, 1, 96, 162, 0, 44,    4045
     162, 1, 164
1064 data 43, 165, 44, 134, 10, 134, 147,      4308
     132, 195, 133, 196
1066 data 32, 212, 225, 32, 119, 205, 32,      4010
     122, 225, 76, 116
1068 data 164, 32, 219, 205, 165, 171, 201,    3837
```


WWW.HOMECOMPUTERWORLD.COM

```
      2, 240, 8, 201
1070 data 1, 208, 243, 165, 185, 240, 10,     3688
      173, 60, 3, 133
1072 data 195, 173, 61, 3, 133, 196, 32,      4426
      80, 247, 32, 228
1074 data 255, 240, 251, 32, 44, 168, 164,    4238
      183, 240, 11, 136
1076 data 177, 187, 217, 65, 3, 208, 206,     4786
      152, 208, 245, 132
1078 data 144, 32, 210, 245, 173, 62, 3,      3860
      56, 237, 60, 3
1080 data 8, 24, 101, 195, 133, 174, 173,     4565
      63, 3, 101, 196
1082 data 40, 237, 61, 3, 133, 175, 32,       4100
      240, 205, 165, 189
1084 data 69, 215, 5, 144, 240, 4, 169,       3941
      255, 133, 144, 76
1086 data 169, 245, 32, 41, 206, 201, 0,      4216
      240, 249, 133, 171
1088 data 32, 87, 206, 145, 178, 200, 192,    4105
      192, 208, 246, 240
1090 data 45, 32, 41, 206, 32, 87, 206,       3664
      196, 147, 208, 2
1092 data 145, 195, 209, 195, 240, 2, 134,    4664
      144, 69, 215, 133
1094 data 215, 230, 195, 208, 2, 230, 196,    4697
      165, 195, 197, 174
1096 data 165, 196, 229, 175, 144, 221, 32,   4868
      87, 206, 32, 247
1098 data 204, 200, 132, 192, 88, 8, 169,     3720
      0, 141, 160, 2
1100 data 76, 147, 252, 32, 23, 248, 32,      4361
      247, 204, 132, 215
1102 data 169, 7, 141, 6, 221, 162, 1, 32,    4589
      106, 206, 38
1104 data 189, 165, 189, 201, 2, 208, 245,    4234
      160, 9, 32, 87
1106 data 206, 201, 2, 240, 249, 196, 189,    3943
      208, 232, 32, 87
1108 data 206, 136, 208, 246, 96, 169, 8,     4223
      133, 163, 32, 106
1110 data 206, 38, 189, 234, 234, 234, 198,   4186
      163, 208, 244, 165
1112 data 189, 96, 169, 16, 44, 13, 220,      4628
      240, 251, 173, 13
1114 data 221, 142, 7, 221, 72, 169, 25,      3936
      141, 15, 221, 104
1116 data 74, 74, 96, 169, 0, 141, 32, 208,   3735
      141, 33, 208
1118 data 141, 0, 4, 141, 1, 4, 141, 2, 4,    3315
      169, 149
1120 data 141, 0, 221, 169, 5, 141, 24,       3843
      208, 169, 128, 141
1122 data 136, 2, 133, 56, 169, 4, 133, 44,   4034
      32, 68, 229
1124 data 96, 32, 24, 229, 162, 0, 189,       3358
      196, 206, 240, 6
1126 data 157, 0, 4, 232, 208, 245, 162, 5,   3534
      160, 0, 24
1128 data 32, 108, 229, 96, 42, 42, 42, 42,   3490
      42, 46, 46
1130 data 46, 46, 46, 7, 2, 5, 5, 20, 9,      2802
      14, 7
1132 data 19, 46, 6, 18, 15, 13, 46, 40,      3296
      46, 50, 50
1134 data 51, 41, 46, 46, 46, 46, 42, 42,     3062
      42, 42, 42
1136 data 0, 0, 0, 255, 255, 255, 255, 255,   3792
      255, 255, 255
```

„Gamer-Tape" nutzt Kassetten als billigen Datenspeicher

Organisierte Datasetten: „Tape-Directory"

System: **Commodore 64, Datasette, Drucker**
Typ: **Anwendung**
Sprache: **Basic**

Um zu wissen, was auf bespielten Kassetten geschrieben ist, muß man das komplette Band auf Programme absuchen. „Tape-Directory" erspart diese langweilige und zeitraubende Arbeit ab.

Nachdem das Programm geladen und gestartet ist, legt man eine beliebige Kassette in die Datasette. Danach drückt man PLAY, den Rest erledigt „Tape-Directory". Jedes gefundene Programm erscheint mit Angaben zur Länge und Anfangsadresse am Bildschirm. Ist die Kassette komplett eingelesen, erscheinen die Daten wahlweise nochmals am Bildschirm oder werden auf dem Drucker ausgegeben. Maximal 100 Programme je Kassettenseite werden bearbeitet.

Kern des Programmes ist eine Maschinenspracheunterroutine in 2 Teilen, die ab Zeile 54 in Basic-DATAs steht. Die Daten stehen nach RUN im Speicher ab Stelle 52992, wobei Teil 2 den Bereich zwischen 53062 und 53107 belegt.

Michael Maier

```
 1 goto 10                                     427
 2 goto 16                                     451
 3 goto 44                                     449
 4 rem **********************
 5 rem *    tape-directory    *
 6 rem **********************
 7 rem * (c) michael maier   *
 8 rem **********************
 9 rem
10 print chr$ (147) chr$ (14) chr$ (8)         4884
   chr$ (153):poke 53280,0:poke 53281,0:
   df=0
11 poke 56,100:clr :dim as$(100,5)             1926
12 print tab(11)"[rvs]Kassettendirectory       3777
   [off]":print :print :print :print
13 print " Bitte Band einlegen und             4688
    zurueckspulen!":print :print
14 gosub 62                                    449
15 goto 34                                     447
16 print "[clr,cbm 6]Kassette enthaelt:":      3264
   ez=ps:print
17 if ps=0 then print "Kein Programm           4327
    gefunden.":gosub 25:goto 53
18 for i=0 to ez                               998
19 for j=0 to 3:print as$(i,j):next j:         2611
   print
20 if df then 22                               466
21 if int (i/3)=i/3 and i>0 then gosub 25      2334
```


WWW.HOMECOMPUTERWORLD.COM

```
22 next i:if df then print#4:close 4:goto 53 2078
23 gosub 25 453
24 goto 28 461
25 print :print "Weiter: Taste":poke 198, 0 2341
26 get w$:if w$="" then 26 1708
27 print "[clr,2down]":return 733
28 print "[down]Drucken,Bildschirm oder Ende[3spaces](d/b/e) ?" 4867
29 get w$:if w$<>"d" and w$<>"e" and w$<>"b" then 29 3855
30 if w$="e" then 53 1310
31 if w$="b" then 16 1216
32 open 4,4,0:close 4:if st=-128 then 32 2439
33 open 4,4,7:cmd 4:df=1:goto 16 2183
34 print "[clr,2down,wht]Suchen beenden: run/stop" 3324
35 for i=828 to 849:poke i,0:next 1819
36 open 1,1,0,"":print "[blk]" 1144
37 if peek (828)>0 and peek (828)<5 then 39 2282
38 close 1:goto 34 751
39 on peek (828) goto 40,41,42,43,44 2268
40 as$(ps,0)="Basicprogramm":goto 45 2376
41 goto 38 463
42 as$(ps,0)="Maschinenprog":goto 45 2593
43 as$(ps,0)="Datenfile[4spaces]":goto 46 1965
44 as$(ps,0)="Bandende[5spaces]":goto 16 1976
45 gosub 49:gosub 52:gosub 51:goto 47 2376
46 gosub 51 443
47 close 1:ps=ps+1:if ps<101 then 34 2563
48 goto 16 451
49 kb=( peek (832)*256+peek (831))-( peek (830)*256+peek (829)) 4291
50 as$(ps,2)="Laenge: "+str$ (kb)+" byte" :return 3314
51 for i=833 to 848:as$(ps,1)=as$(ps,1)+chr$ ( peek (i)):next :return 3997
52 as$(ps,3)="Anfangsadresse: "+str$ ( peek (830)*256+peek (829)):return 5106
53 end 128
54 data 120,169,207,141,41,3,169,48,141, 40,3,88,96,0,0,0 5028
55 data 169,71,141,119,2,169,207,141,120, 2,169,50,141,121,2,169 5957
56 data 13,141,122,2,169,4,133,198,162, 255,134,145,24,108,2,160 5451
57 data 165,145,201,127,240,3,76,237,246, 32,147,252,104,104,169,0 6691
58 data 141,160,2,240,203,0 2000
59 data 169,81,141,0,3,169,207,141,1,3, 96,224,5,208,27,169 5552
60 data 71,141,119,2,169,207,141,120,2, 169,51,141,121,2,169,13 5461
61 data 141,122,2,169,4,133,198,76,116, 164,76,139,227,255 5011
62 for x=52992 to 53107:read d:poke x,d: next :sys 52992:sys 53062:return 4853
```

Gibt das „Directory" einer Kassette aus.

DMON-Diskmonitor

System: **Commodore 64, Laufwerk**
Typ: **Anwendung**
Sprache: **ASM**

Ein Diskmonitor bearbeitet direkt einzelne Spuren und Sektoren des Laufwerkes. Sämtliche Eingaben und Ausgaben im Programm erfolgen hexadezimal. Listen des Speicherinhalts erfolgt mit SHIFT-Taste, der Abbruch des Befehls mit STOP. Die Befehle im einzelnen:

R: Read Track/Sektor
'R 12 01'
liest einen Block ein, der auf Track 12 (= 18 dezimal), Sektor 01 liegt. Hierbei handelt es sich um den ersten Block des Directory. 'R' liest den logischen nächsten Block ein.

W: Write Track/Sektor
'W 12 01'
schreibt den Block im Speicher auf die Diskette nach 12 01 zurück. 'W' schreibt den vorher eingelesenen Block an seinen Platz zurück.

B: BAM anzeigen
'B' zeigt auf dem Bildschirm die belegten Blöcke der Diskette an. In horizontaler Richtung die Tracks von $01 bis $23 (35 dezimal), und in vertikaler Richtung die Sektoren von $00 bis $14 (20 dezimal). Tracks, die weniger als 20 Sektoren haben, werden durch '*' markiert.

Üblicherweise nutzt man das Laufwerk 1541 über DOS (Disk Operating System). Dadurch sind Manipulationen eingeschränkt. DMON geht über diese Grenzen hinaus.

BA: Block im BAM belegen
'BA 02 03' kennzeichnet diesen Block als belegt. Um sich von der Wirkung zu überzeugen, gibt man danach 'B' ein. Der belegte Block ist an dem weißen Quadrat zu erkennen.

BF: Block in BAM frei machen
'BF 02 03' sorgt dafür, daß der oben belegte Block wieder als frei in die BAM eingetragen wird.

M: Anzeige des Floppy-Speichers (RAM und ROM) als Hex-Dump
'M 00' gibt den Inhalt der Floppy-Zero-Page als Hex-Dump aus. Der Inhalt dieser Zero-Page wird ständig durch Interrupts ausgelesen. In der Zero-Page ruft man verschiedene Floppy-Kommandos direkt auf.

Die folgenden Job-Codes sind von Bedeutung:

$80 LESEN eines Blocks in einen Puffer
$90 SCHREIBEN eines Blocks aus einem Puffer
$D0 oder
$E0 AUSFÜHREN eines Maschinenprogrammes im Puffer. $E0 stellt zusätzlich den Motor an.

Durch Überschreiben der ersten Speicherstellen liest man den Job-Code LESEN ($80) in die Speicherstelle für Puffer 0 ein. Beispiel:
Nach 'M' erscheint:
0000 01 01 00 0B 02 00 12 04
0008 01 02 00 23 12 00 21 33
'STOP' beendet das Listen.
Die ersten vier Zahlen (0000) geben die Adresse der Floppy-Speicherstelle an. Es folgen die obengenannten Jobspeicher 0 bis 5, und in den letzten beiden Zahlen sind die Track- und Sektornummern für Puffer 0. Um den ersten Directory-Block einzulesen, muß man '12 01' als Track/Sektornummer in $0006/07 eingeben. Da-

WWW.HOMECOMPUTERWORLD.COM

nach schreibt man den Job-Code in Speicherstelle 0000. Änderung:
0000 80 01 00 0B 02 00 12 01
Hierbei ist zu beachten, daß zuerst die Track- und Sektornummer eingegeben wird, damit beim Eintragen des Job-Codes ($80) der zu lesende Block festgelegt ist. Der Job-Code führt nach 'RETURN' sofort zur Befehlsausführung.
Fehlerkanal abfragen:
Den Floppyfehlerkanal liest man mit dem 'Klammeraffen' aus. Zusammen mit einem Buchstaben werden Kommandos an die Floppy übermittelt. Beispiel:
I = Initialisieren
Auf diese Art werden alle anderen Diskkommandos an die Floppy übermittelt.
$: Direktory anzeigen
L: Linkung der Blöcke angeben. Hierbei besteht die Möglichkeit; entweder einzelne Programme ('LINK-ONE') oder die ganze Diskette ('LINK-ALL') zu überprüfen. Vor allem bei 'Validate'-Fehlern ist dieser Befehl hilfreich. Der dritte Linkbefehl ('LINK-FRE') zeigt die Linkung einer bestimmbaren Anzahl an Blöcken an, die nicht in der BAM eingetragen sind.
V: Verschieben eines Programmes aus dem Speicher in einen Floppy-Puffer:
Der Block aus Puffer 0 muß anschließend durch Eingabe von 'M' und Schreiben des Jobcodes $90 in die erste Speicherstelle auf der Diskette abgelegt werden. K setzt Koppeladresse auf '00 FF'. Dieser Befehl sorgt dafür, daß die ersten beiden Bytes des eben abgelegten Blocks nicht länger eine zufällige Bytefolge aufweisen.
'H' zeigt alle Kommandos an.
'X' verläßt das Programm.
Das vorliegende Listing ist als kombinierter Hex-Dump-Basicladder abgedruckt. Wie man mit dieser Art Listings umgeht, steht auf Seite 8. Wird das Programm als Basic-Lader eingegeben, so muß der Lader vor dem Start abgespeichert werden. Nach RUN werden dann die Daten in den Speicher gepoket. Das fertige Maschinenprogramm speichert man mit:
POKE 43, 0: POKE 44,192
POKE 45,40: POKE 46,205
SAVE „name",8
auf Diskette ab. Wird es als Hexdump eingegeben, so wird der Checker benutzt. Als Startadresse ist C000 einzugeben.

N. Mann

```
c000   10 data 4c,a6,c1,20,2a,c0,c9,20, 5a
c008   20 data f0,f9,d0,03,20,2a,c0,20, 59
c010   30 data 21,c0,0a,0a,0a,0a,85,a7, 40
c018   40 data 20,2a,c0,20,21,c0,05,a7, 56
c020   50 data 60,c9,3a,90,02,69,08,29, bb
c028   60 data 0f,60,20,cf,ff,c9,0d,d0, 74
c030   70 data f8,a9,3f,20,d2,ff,4c,a6, e7
c038   80 data c1,a2,fb,20,03,c0,95,01, d1
c040   90 data 20,0c,c0,95,00,e8,e8,60, df
c048  100 data 20,cf,ff,c6,d3,c9,0d,60, 42
c050  110 data a5,fc,20,57,c0,a5,fb,48, 40
c058  120 data 4a,4a,4a,4a,20,62,c0,68, 16
c060  130 data 29,0f,c9,0a,90,02,69,06, c9
c068  140 data 69,30,4c,d2,ff,a9,0d,20, f6
c070  150 data d2,ff,8a,4c,d2,ff,84,ac, bc
c078  160 data 85,ab,a0,00,b1,ab,f0,1d, 52
c080  170 data c9,03,b0,06,aa,bd,25,cd, b4
c088  180 data d0,09,c9,0d,b0,05,aa,ca, 0c
c090  190 data 20,ba,c0,20,d2,ff,c8,d0, d1
c098  200 data e3,e6,ac,d0,df,60,e6,fb, 42
c0a0  210 data d0,02,e6,fc,60,a2,30,38, 6e
c0a8  220 data e9,0a,90,03,e8,b0,f9,69, 32
c0b0  230 data 3a,60,a9,0d,2c,a9,20,4c, cd
c0b8  240 data d2,ff,a9,20,20,d2,ff,ca, df
c0c0  250 data d0,fa,60,a9,00,85,90,20, 79
c0c8  260 data a5,c3,a9,93,20,d2,ff,20, 25
c0d0  270 data 88,c1,a9,24,20,a8,ff,20, a3
c0d8  280 data ae,ff,a5,90,10,03,4c,28, f5
c0e0  290 data c1,a9,08,20,b4,ff,a9,60, fe
c0e8  300 data 20,96,ff,20,a5,ff,20,a5, 8a
c0f0  310 data ff,4c,0b,c1,20,a5,ff,aa, cd
c0f8  320 data 20,a5,ff,20,cd,bd,20,b5, d5
c100  330 data c0,20,a5,ff,f0,05,20,d2, 7e
c108  340 data ff,d0,f6,20,a5,ff,20,a5, 26
c110  350 data ff,a5,90,d0,13,20,b2,c0, ff
c118  360 data ad,8d,02,c9,02,d0,d5,ad, b4
c120  370 data 8d,02,c9,01,d0,f9,f0,cc, 4c
c128  380 data 20,ab,ff,a9,08,20,b1,ff, 65
c130  390 data a9,e0,20,93,ff,20,ae,ff, e4
c138  400 data 20,41,c1,20,b2,c0,4c,cf, ef
c140  410 data c1,a9,08,20,b4,ff,a9,6f, ef
c148  420 data 20,96,ff,20,a5,ff,c9,31, a7
c150  430 data 08,90,0b,48,20,b2,c0,20, 4a
c158  440 data b2,c0,68,20,d2,ff,20,a5, 88
c160  450 data ff,c9,0d,f0,0a,28,08,90, ae
c168  460 data f5,20,d2,ff,4c,5e,c1,20, 36
c170  470 data ab,ff,a9,00,85,90,28,60, 12
c178  480 data 20,e4,ff,f0,fb,60,a9,08, 87
c180  490 data 20,b1,ff,a9,6f,4c,93,ff, 7c
c188  500 data a9,08,20,b1,ff,a9,f0,4c, 09
c190  510 data 93,ff,a9,08,20,b1,ff,a9, fa
c198  520 data e0,4c,93,ff,a9,08,20,b4, 35
c1a0  530 data ff,a9,6f,4c,96,ff,a0,cc, e4
c1a8  540 data a9,60,20,76,c0,a0,cd,a9, 37
c1b0  550 data 11,20,76,c0,20,e4,ff,f0, f3
c1b8  560 data fb,20,7e,c1,a9,49,20,a8, 6f
c1c0  570 data ff,20,ae,ff,20,aa,c3,a9, 1e
c1c8  580 data 01,8d,40,03,8d,41,03,a2, 5e
c1d0  590 data ff,9a,20,b2,c0,a2,3e,20, 0f
c1d8  600 data 6d,c0,20,cf,ff,c9,3e,f0, 83
c1e0  610 data f9,c9,20,f0,f5,a2,0d,dd, e4
c1e8  620 data ba,c3,f0,09,ca,d0,f8,20, af
c1f0  630 data b2,c0,4c,d5,c1,8a,0a,aa, 01
c1f8  640 data e8,bd,c6,c3,48,ca,bd,c6, a3
c200  650 data c3,48,60,20,48,c0,d0,0a, 08
c208  660 data a9,00,8d,b2,c3,8d,b3,c3, a9
c210  670 data f0,1a,20,03,c0,8d,b3,c3, 15
c218  680 data 20,48,c0,d0,07,a9,00,8d, 9a
c220  690 data b2,c3,f0,08,20,03,c0,29, 8a
c228  700 data f8,8d,b2,c3,20,3d,c2,a0, 5f
c230  710 data 00,20,a5,ff,99,00,cf,c8, 26
c238  720 data d0,f7,4c,ab,ff,20,7e,c1, 16
c240  730 data a2,00,bd,af,c3,20,a8,ff, fb
c248  740 data e8,e0,06,90,f5,20,ae,ff, 01
c250  750 data 4c,9c,c1,a9,cf,85,fc,a9, 64
c258  760 data 00,85,fb,60,20,03,c2,a9, 4c
c260  770 data 00,2c,a9,ff,8d,3c,03,20, b3
c268  780 data 53,c2,a2,3a,20,6d,c0,ad, bf
c270  790 data b3,c3,20,57,c0,ad,b2,c3, bb
c278  800 data 20,57,c0,a0,20,a2,00,20, 48
c280  810 data b5,c0,20,b5,c0,a1,fb,20, 5a
c288  820 data 57,c0,a1,fb,20,cc,c2,d0, 84
c290  830 data f1,a9,08,18,6d,b2,c3,8d, 29
c298  840 data b2,c3,08,c9,f8,d0,06,20, 3d
c2a0  850 data 2c,c2,20,53,c2,28,90,11, 50
c2a8  860 data ad,3c,03,10,03,4c,cf,c1, 29
c2b0  870 data ee,b3,c3,20,2c,c2,20,53, 15
c2b8  880 data c2,20,e1,ff,d0,03,4c,cf, cd
c2c0  890 data c1,ad,8d,02,f0,f3,a9,00, 44
c2c8  900 data 85,c6,f0,9e,c9,20,90,0c, 3d
c2d0  910 data c9,60,90,0a,c9,c0,90,04, 83
c2d8  920 data c9,db,90,04,a9,2e,29,3f, de
c2e0  930 data 29,7f,91,d1,ad,86,02,91, 03
c2e8  940 data f3,20,9e,c0,c8,c0,28,60, 81
c2f0  950 data 20,39,c0,a5,fb,8d,b8,c3, 64
c2f8  960 data a5,fc,8d,b9,c3,20,27,c3, 84
c300  970 data a0,20,a2,00,20,2a,c0,20, b7
c308  980 data 2a,c0,20,0c,c0,20,a8,ff, c7
c310  990 data 20,cc,c2,d0,f2,20,ae,ff, c7
c318 1000 data 4c,d7,c1,20,9e,c0,a5,fb, 9e
c320 1010 data c5,fd,a5,fc,e5,fe,60,20, 99
c328 1020 data 7e,c1,a2,00,bd,b5,c3,20, 09
c330 1030 data a8,ff,e8,e0,06,90,f5,60, bc
c338 1040 data 20,3e,c3,4c,cf,c1,20,39, 4e
c340 1050 data c0,a2,fd,20,3b,c0,a5,fd, 1d
c348 1060 data 8d,b8,c3,a5,fe,8d,b9,c3, 59
c350 1070 data a9,20,8d,ba,c3,20,27,c3, 62
c358 1080 data a0,00,b1,fb,20,a8,ff,c8, 05
```


WWW.HOMECOMPUTERWORLD.COM

```
c360  1090 data c0,20,90,f6,20,ae,ff,18, 93
c368  1100 data a9,20,6d,b8,c3,b0,0c,8d, d0
c370  1110 data b8,c3,a9,20,65,fb,85,fb, 72
c378  1120 data 4c,55,c3,a9,08,8d,ba,c3, 83
c380  1130 data 60,20,48,c0,d0,06,20,41, 71
c388  1140 data c1,4c,cf,c1,20,7e,c1,20, c6
c390  1150 data cf,ff,20,a8,ff,c9,0d,d0, bc
c398  1160 data f6,20,ae,ff,4c,cf,c1,20, a3
c3a0  1170 data a5,c3,20,74,a4,a9,37,85, 3c
c3a8  1180 data 01,60,a9,36,85,01,60,4d, ab
c3b0  1190 data 2d,52,00,00,ff,4d,2d,57, 63
c3b8  1200 data 00,00,08,3a,42,4d,56,40, d9
c3c0  1210 data 24,52,57,4c,53,4b,48,58, d5
c3c8  1220 data ef,c2,af,c6,5b,c2,37,c3, 24
c3d0  1230 data 80,c3,c2,c0,e1,c3,9b,c4, b4
c3d8  1240 data b7,c4,be,c8,36,c9,bb,ca, 47
c3e0  1250 data 9e,c3,20,48,c0,d0,22,20, a6
c3e8  1260 data 8f,c4,20,3c,c4,a2,00,bd, 15
c3f0  1270 data 00,cf,20,88,ca,b0,2d,20, f1
c3f8  1280 data a8,ff,e8,bd,00,cf,20,92, 94
c400  1290 data ca,b0,21,20,a8,ff,4c,0f, 01
c408  1300 data c4,20,3c,c4,20,4e,c4,20, 92
c410  1310 data ae,ff,20,31,c4,20,7c,c4, fa
c418  1320 data 20,8f,c4,20,74,c4,20,62, a3
c420  1330 data c2,4c,cf,c1,a9,9f,a0,cc, 61
c428  1340 data 20,76,c0,20,74,c4,4c,cf, 77
c430  1350 data c1,20,64,c4,a9,80,20,a8, e2
c438  1360 data ff,4c,ae,ff,a9,06,8d,b8, d9
c440  1370 data c3,a9,00,8d,b9,c3,a9,02, 2a
c448  1380 data 8d,ba,c3,4c,27,c3,20,03, cc
c450  1390 data c0,20,88,ca,b0,ce,20,a8, b8
c458  1400 data ff,20,03,c0,20,92,ca,b0, ca
c460  1410 data c3,4c,a8,ff,a9,00,8d,b8, 9d
c468  1420 data c3,8d,b9,c3,a9,01,8d,ba, a6
c470  1430 data c3,4c,27,c3,a9,08,8d,ba, 4e
c478  1440 data c3,4c,ae,ff,a9,00,8d,b2, a9
c480  1450 data c3,8d,b3,c3,20,3d,c2,20, ab
c488  1460 data a5,ff,30,f0,4c,ab,ff,a9, de
c490  1470 data 03,8d,b3,c3,a9,00,8d,b2, ea
c498  1480 data c3,4c,2c,c2,20,48,c0,f0, 89
c4a0  1490 data 06,20,3c,c4,20,4e,c4,20, d5
c4a8  1500 data ae,ff,20,64,c4,a9,90,20, f6
c4b0  1510 data a8,ff,20,74,c4,4c,cf,c1, db
c4b8  1520 data a9,35,a0,cc,20,76,c0,a9, 09
c4c0  1530 data 01,8d,3d,03,8d,3e,03,20, e0
c4c8  1540 data e4,ff,f0,fb,c9,31,d0,03, 3e
c4d0  1550 data 4c,1e,c6,c9,32,f0,03,4c, 25
c4d8  1560 data 7c,c9,20,67,c5,a0,05,20, 77
c4e0  1570 data b2,c0,20,b2,c0,20,b2,c0, f2
c4e8  1580 data a9,12,20,d2,ff,a2,f1,b9, 79
c4f0  1590 data 00,cf,f0,47,20,d2,ff,c8, 5f
c4f8  1600 data e8,d0,f4,ad,3d,03,20,57, 62
c500  1610 data c0,a9,92,20,d2,ff,20,b5, c7
c508  1620 data c0,98,38,e9,11,48,a8,20, c9
c510  1630 data 51,c5,a9,00,8d,01,ce,20, 6e
c518  1640 data b2,c0,20,9d,c5,20,c2,c5, 17
c520  1650 data 20,41,c5,68,18,69,22,a8, 65
c528  1660 data 90,b5,ad,3e,03,f0,0c,ad, bd
c530  1670 data 3d,03,18,69,03,8d,3d,03, 9a
c538  1680 data 4c,da,c4,20,74,c4,4c,cf, 43
c540  1690 data c1,ad,00,ce,38,ed,01,ce, c6
c548  1700 data f0,52,a9,bb,a0,cc,4c,76, 36
c550  1710 data c0,18,69,1b,aa,bd,00,cf, 14
c558  1720 data 8d,00,ce,20,b5,c0,20,b5, 9a
c560  1730 data c0,ad,00,ce,4c,57,c0,20, db
c568  1740 data 3c,c4,a9,12,20,a8,ff,ad, d9
c570  1750 data 3d,03,20,a8,ff,20,ae,ff, 3f
c578  1760 data 20,31,c4,20,7c,c4,20,8f, dc
c580  1770 data c4,a0,ff,8c,b2,c3,20,3d, 69
c588  1780 data c2,20,a5,ff,99,00,cf,20, 8f
c590  1790 data ab,ff,a0,00,b9,00,cf,d0, 03
c598  1800 data 03,8c,3e,03,60,b9,00,cf, 8b
c5a0  1810 data 48,c8,b9,00,cf,48,20,3c, a4
c5a8  1820 data c4,68,aa,68,20,a8,ff,20, f4
c5b0  1830 data 57,c0,8a,20,a8,ff,20,57, 74
c5b8  1840 data c0,ee,01,ce,20,b5,c0,4c, e5
c5c0  1850 data ae,ff,20,31,c4,20,ae,ff, f1
c5c8  1860 data 20,7c,c4,a9,00,8d,b2,c3, 22
c5d0  1870 data a9,03,8d,b3,c3,20,3d,c2, 9d
c5d8  1880 data 20,a5,ff,48,20,a5,ff,48, 64
c5e0  1890 data 20,ab,ff,20,3c,c4,68,aa, 8a
c5e8  1900 data 68,20,a8,ff,f0,2d,20,88, 4c
c5f0  1910 data ca,90,03,4c,24,c4,20,57, 1b
c5f8  1920 data c0,8a,20,a8,ff,20,57,c0, 24
c600  1930 data 20,ae,ff,ee,01,ce,20,b5, 23
c608  1940 data c0,20,e1,ff,d0,03,4c,3b, 5f
c610  1950 data c5,ad,8d,02,f0,ac,68,68, e6
c618  1960 data 4c,23,c5,4c,74,c4,a9,51, aa
c620  1970 data a0,cc,20,76,c0,a2,00,8e, 0f
c628  1980 data 01,ce,20,cf,ff,9d,02,ce, 1c
c630  1990 data e8,c9,0d,d0,f5,e8,8a,8d, 67
c638  2000 data 00,ce,20,b2,c0,20,67,c5, e3
c640  2010 data a0,05,a2,00,bd,02,ce,c9, fb
c648  2020 data 0d,f0,28,d9,00,cf,08,e8, c0
c650  2030 data c8,28,f0,f0,98,18,69,20, 68
c658  2040 data 29,f5,09,05,a8,90,e3,ad, 86
c660  2050 data 3e,03,d0,03,4c,3b,c5,ad, 30
c668  2060 data 3d,03,18,69,03,8d,3d,03, 9a
c670  2070 data 4c,3d,c6,98,38,ed,00,ce, bb
c678  2080 data 48,18,69,02,a8,20,b2,c0, 10
c680  2090 data a9,12,20,d2,ff,a2,10,b9, 99
c688  2100 data 00,cf,20,d2,ff,c8,ca,d0, b1
c690  2110 data f6,ad,3d,03,20,57,c0,a9, 62
c698  2120 data 92,20,d2,ff,68,a8,20,51, d4
c6a0  2130 data c5,20,b2,c0,20,9d,c5,20, bf
c6a8  2140 data c2,c5,20,41,c5,4c,cf,c1, 62
c6b0  2150 data 20,48,c0,f0,12,20,cf,ff, 6a
c6b8  2160 data c9,41,d0,03,4c,2d,c8,a9, 91
c6c0  2170 data 46,8d,87,c8,4c,32,c8,20, 3a
c6c8  2180 data cd,c6,4c,eb,c6,20,a5,c3, f0
c6d0  2190 data a9,07,8d,b3,c3,a9,00,8d, 09
c6d8  2200 data b2,c3,20,3d,c2,a0,00,20, d5
c6e0  2210 data a5,ff,99,00,ce,c8,d0,f7, 1d
c6e8  2220 data 4c,ab,ff,a2,00,bd,8f,c8, 09
c6f0  2230 data 20,d2,ff,f0,03,e8,d0,f5, 54
c6f8  2240 data a2,90,bd,00,ce,20,d2,ff, 4e
c700  2250 data e8,e0,a2,90,f5,a2,00,bd, b0
c708  2260 data a2,c8,20,d2,ff,f0,03,e8, 53
c710  2270 data d0,f5,ad,a2,ce,20,d2,ff, 66
c718  2280 data ad,a3,ce,20,d2,ff,a2,04, 28
c720  2290 data a0,00,20,0a,e5,a2,34,a9, 83
c728  2300 data 31,20,23,c8,20,b2,c0,e0, bb
c730  2310 data 30,b0,f4,a2,46,a9,30,20, 80
c738  2320 data 23,c8,20,b2,c0,e0,41,b0, 3c
c740  2330 data f4,a2,39,a9,30,20,23,c8, 4d
c748  2340 data e0,30,30,05,20,b2,c0,d0, 38
c750  2350 data f2,a2,02,a0,03,20,0a,e5, bb
c758  2360 data a0,23,a9,30,a2,0f,20,d2, d6
c760  2370 data ff,88,f0,0a,ca,d0,f7,18, 34
c768  2380 data 69,01,a2,10,d0,f0,a2,03, 78
c770  2390 data a0,03,20,0a,e5,a0,23,a9, 71
c778  2400 data 31,a2,0f,20,d2,ff,18,69, 01
c780  2410 data 01,88,f0,12,ca,d0,06,a9, ae
c788  2420 data 30,a2,10,d0,ee,c9,3a,d0, 5e
c790  2430 data ea,18,69,07,d0,e5,a9,01, c6
c798  2440 data 85,02,a2,04,a5,02,69,02, 29
c7a0  2450 data a8,20,0a,e5,a2,14,86,8b, 36
c7a8  2460 data a9,ce,85,8d,a9,01,a6,02, 1e
c7b0  2470 data 69,04,ca,d0,fb,85,8c,a5, bb
c7b8  2480 data 8b,a6,02,e0,12,90,20,e0, cb
c7c0  2490 data 19,90,14,e0,1f,90,08,c9, 8d
c7c8  2500 data 11,90,14,a2,08,d0,2a,c9, 8e
c7d0  2510 data 12,90,0c,a2,08,d0,22,c9, 7f
c7d8  2520 data 13,90,04,a2,08,d0,1a,a5, 94
c7e0  2530 data 8b,4a,4a,4a,a8,b1,8c,a8, 1c
c7e8  2540 data a5,8b,29,07,aa,98,3d,ab, e0
c7f0  2550 data c8,d0,04,a2,00,f0,02,a2, cc
c7f8  2560 data 04,a0,04,bd,b3,c8,20,d2, e6
c800  2570 data ff,e8,88,d0,f6,a5,8b,f0, bb
c808  2580 data 05,a9,11,20,d2,ff,c6,8b, 5c
c810  2590 data 10,a5,e6,02,a5,02,c9,24, 96
c818  2600 data d0,80,20,e7,ff,20,78,c1, 1f
c820  2610 data 4c,cf,c1,20,6f,c0,a9,2d, 49
c828  2620 data 20,d2,ff,ca,60,a9,41,8d, ef
c830  2630 data 87,c8,a9,08,20,b1,ff,a9, 25
c838  2640 data f2,20,93,ff,a9,23,20,a8, 63
c840  2650 data ff,20,ae,ff,20,03,c0,8d, dd
c848  2660 data 40,03,20,a5,c0,8e,89,c8, ab
c850  2670 data 8d,8a,c8,20,03,c0,8d,41, 3a
c858  2680 data 03,20,a5,c0,8e,8c,c8,8d, 05
c860  2690 data 8d,c8,20,7e,c1,a2,00,bd, ca
c868  2700 data 85,c8,20,a8,ff,e8,e0,09, 23
c870  2710 data 90,f5,20,ae,ff,a9,08,20, 4c
c878  2720 data b1,ff,a9,e2,20,93,ff,20, e5
c880  2730 data ae,ff,4c,37,c9,42,2d,41, 37
c888  2740 data 00,1e,1e,00,1e,1e,00,93, 6d
c890  2750 data 2e,3d,46,52,45,49,0d,44, ab
c898  2760 data 49,53,4b,4e,41,4d,45,3a, f2
c8a0  2770 data 20,00,20,20,20,20,49,44, 25
c8a8  2780 data 3a,20,00,01,02,04,08,10, 0f
c8b0  2790 data 20,40,80,12,20,92,9d,05, 74
c8b8  2800 data 2e,05,9d,9e,2a,05,9d,a9, 41
c8c0  2810 data 01,8d,3d,03,20,67,c5,a0, 8c
c8c8  2820 data 02,b9,00,cf,f0,13,98,18, d8
```


WWW.HOMECOMPUTERWORLD.COM

```
c8d0  2830 data 69,20,a8,90,f4,ad,3d,03, e2
c8d8  2840 data 18,69,03,8d,3d,03,4c,c4, e8
c8e0  2850 data c8,a9,82,99,00,cf,c8,ad, 54
c8e8  2860 data 40,03,99,00,cf,c8,ad,41, 48
c8f0  2870 data 03,99,00,cf,98,48,a9,d2, c3
c8f8  2880 data a0,cc,20,76,c0,68,a8,a2, dc
c900  2890 data 00,20,cf,ff,c8,e8,c9,0d, 4c
c908  2900 data f0,05,99,00,cf,d0,f2,a9, ca
c910  2910 data a0,99,00,cf,c8,e8,e0,11, e8
c918  2920 data 90,f5,98,18,69,09,a8,a9, 79
c920  2930 data 01,99,00,cf,a9,00,8d,b8, 18
c928  2940 data c3,a9,03,8d,b9,c3,20,53, 54
c930  2950 data c2,20,50,c3,4c,a7,c4,20, 78
c938  2960 data 3c,c4,ad,40,03,20,a8,ff, 71
c940  2970 data ad,41,03,20,a8,ff,20,ae, 6a
c948  2980 data ff,20,31,c4,20,7c,c4,20, 94
c950  2990 data 74,c4,a9,00,8d,b8,c3,a9, 48
c958  3000 data 03,8d,b9,c3,a9,02,8d,ba, e6
c960  3010 data c3,20,27,c3,a9,00,20,a8, 28
c968  3020 data ff,a9,ff,20,a8,ff,20,ae, 50
c970  3030 data ff,20,a7,c4,a9,08,8d,ba, 35
c978  3040 data c3,4c,cf,c1,a9,e5,a0,cc, 1d
c980  3050 data 20,76,c0,20,0c,c0,8d,46, dd
c988  3060 data 03,8d,45,03,a9,ff,a0,cc, 36
c990  3070 data 20,76,c0,a9,08,a0,cd,20, d7
c998  3080 data 76,c0,20,cd,c6,a9,12,8d, ac
c9a0  3090 data 43,03,a0,44,ce,43,03,20, 09
c9a8  3100 data b2,c0,ad,43,03,48,20,57, e0
c9b0  3110 data c0,20,b5,c0,20,b5,c0,68, 58
c9b8  3120 data c9,12,d0,03,4c,cf,c1,b9, 08
c9c0  3130 data 00,ce,d0,12,38,98,e9,04, 75
c9c8  3140 data f0,03,a8,d0,d7,a0,8c,a9, de
c9d0  3150 data 24,8d,43,03,d0,ce,8d,47, 1f
c9d8  3160 data 03,a9,00,8d,44,03,84,a9, ea
c9e0  3170 data a2,08,8e,48,03,c8,b9,00, d4
c9e8  3180 data ce,18,4a,90,11,85,aa,20, 86
c9f0  3190 data 08,ca,a5,aa,ce,47,03,d0, f4
c9f8  3200 data 05,a4,a9,4c,c4,c9,ee,44, 63
ca00  3210 data 03,ce,48,03,d0,e3,f0,d8, 7f
ca08  3220 data 84,a8,ad,45,03,8d,46,03, fd
ca10  3230 data 20,9a,ca,20,36,ca,ce,46, 24
ca18  3240 data 03,d0,f8,ad,43,03,f0,9c, 12
ca20  3250 data a9,12,20,d2,ff,ad,44,03, 78
ca28  3260 data 20,57,c0,a9,92,20,d2,ff, 24
ca30  3270 data 20,b5,c0,a4,a8,60,20,31, bf
ca38  3280 data c4,20,ae,ff,20,7c,c4,a9, 12
ca40  3290 data 00,8d,b2,c3,a9,03,8d,b3, e2
ca48  3300 data c3,20,3d,c2,20,a5,ff,48, 50
ca50  3310 data 20,a5,ff,48,20,ab,ff,20, 86
ca58  3320 data e1,ff,d0,03,4c,cf,c1,ad, 40
ca60  3330 data 8d,02,f0,03,4c,ac,ca,20, c1
ca68  3340 data 3c,c4,68,aa,68,20,88,ca, 3c
ca70  3350 data 90,04,68,4c,ac,ca,20,a8, 02
ca78  3360 data ff,8a,20,92,ca,90,03,4c, f4
ca80  3370 data ac,ca,20,a8,ff,4c,ae,ff, bc
ca88  3380 data f0,06,c9,24,b0,02,18,60, f3
ca90  3390 data 38,60,30,fc,c9,15,b0,f8, 78
ca98  3400 data 18,60,20,3c,c4,ad,43,03, f4
caa0  3410 data 20,a8,ff,ad,44,03,20,a8, 83
caa8  3420 data ff,4c,ae,ff,20,ae,ff,a2, 31
cab0  3430 data ff,9a,a9,00,8d,46,03,a4, b3
cab8  3440 data a8,4c,f2,c9,a9,93,20,d2, e9
cac0  3450 data ff,a9,d2,a0,ca,20,76,c0, e7
cac8  3460 data a9,b5,a0,cb,20,76,c0,4c, e7
cad0  3470 data cf,c1,01,04,2a,2a,2a,20, 15
cad8  3480 data 4b,4f,4d,4d,41,4e,44,4f, e4
cae0  3490 data 53,20,3a,20,2a,2a,2a,0d, 6a
cae8  3500 data 0d,20,42,20,2d,3e,20,41, dd
caf0  3510 data 4e,5a,45,49,47,45,20,44, ce
caf8  3520 data 45,52,20,42,41,4d,0d,20, b3
cb00  3530 data 4d,20,2d,3e,20,41,4e,5a, ef
cb08  3540 data 45,49,47,45,20,44,45,52, cd
cb10  3550 data 20,5a,45,52,4f,2d,50,41, ea
cb18  3560 data 47,45,0d,20,56,20,2d,3e, 14
cb20  3570 data 20,56,45,52,53,43,48,49, cc
cb28  3580 data 45,42,54,20,56,4f,4e,20, 6c
cb30  3590 data 53,50,45,49,43,48,45,52, ed
cb38  3600 data 20,49,4e,20,46,4c,4f,50, fe
cb40  3610 data 50,59,0d,20,40,20,2d,3e, f3
cb48  3620 data 20,46,45,48,4c,45,52,41, ef
cb50  3630 data 42,46,52,41,47,45,20,4f, e0
cb58  3640 data 44,45,52,20,46,4c,4f,50, 2a
cb60  3650 data 50,59,42,45,46,45,48,4c, f1
cb68  3660 data 45,0d,20,24,20,2d,3e,20, 45
cb70  3670 data 44,49,52,45,43,54,4f,52, f4
cb78  3680 data 59,20,41,4e,5a,45,49,47, 43
cb80  3690 data 45,4e,0d,20,52,20,2d,3e, 05
cb88  3700 data 20,52,45,41,44,20,54,52, f8
cb90  3710 data 41,43,4b,2f,53,45,43,54, 17
cb98  3720 data 4f,52,0d,20,57,20,2d,3e, 10
cba0  3730 data 20,57,52,49,54,45,20,54, ad
cba8  3740 data 52,41,43,4b,2f,53,45,4b, df
cbb0  3750 data 54,4f,52,0d,00,20,4c,20, 56
cbb8  3760 data 2d,3e,20,4c,49,4e,4b,55, b4
cbc0  3770 data 4e,47,20,55,4e,54,45,52, bf
cbc8  3780 data 53,55,43,48,45,4e,0d,20, de
cbd0  3790 data 53,20,2d,3e,20,53,43,48, ea
cbd8  3800 data 52,45,49,42,54,20,4e,41, 55
cbe0  3810 data 4d,45,20,49,4e,20,44,49, 08
cbe8  3820 data 52,45,43,54,4f,52,59,0d, 45
cbf0  3830 data 20,4b,20,2d,3e,20,4b,4f, e2
cbf8  3840 data 50,50,45,4c,41,44,52,45, 03
cc00  3850 data 53,53,45,20,4c,4f,45,53, 14
cc08  3860 data 43,48,45,4e,0d,20,48,20, 07
cc10  3870 data 2d,3e,20,48,45,4c,50,20, f0
cc18  3880 data 2c,20,4c,49,53,54,45,54, ff
cc20  3890 data 20,42,45,46,45,48,4c,45, e1
cc28  3900 data 0d,20,58,20,2d,3e,20,45, ef
cc30  3910 data 58,49,54,0d,00,0d,0d,4c, 0a
cc38  3920 data 49,4e,4b,2f,4f,4e,45,28, 35
cc40  3930 data 31,29,2f,41,4c,4c,28,32, ec
cc48  3940 data 29,2f,46,52,45,28,33,29, 15
cc50  3950 data 00,0d,0d,46,49,4c,45,4e, af
cc58  3960 data 41,4d,45,4e,20,3f,3a,00, 06
cc60  3970 data 01,93,0d,0d,20,20,20,20, 6f
cc68  3980 data 20,20,20,20,2a,2a,2a,20, 0a
cc70  3990 data 46,4c,4f,50,50,59,2d,4d, d0
cc78  4000 data 4f,4e,49,54,4f,52,20,2a, e9
cc80  4010 data 2a,2a,0d,20,20,20,20,20, ed
cc88  4020 data 20,20,20,20,20,20,20,56, ca
cc90  4030 data 4f,4e,20,4e,2e,4d,2e,20, c3
cc98  4040 data 26,20,44,2e,57,2e,00,02, 43
cca0  4050 data 0d,0d,12,46,41,4c,53,43, d1
cca8  4060 data 48,45,52,20,42,4c,4f,43, 37
ccb0  4070 data 4b,2f,53,45,4b,54,4f,52, 1e
ccb8  4080 data 92,01,00,02,0d,12,42,4c, 80
ccc0  4090 data 4f,43,4b,41,4e,5a,2e,20, 18
ccc8  4100 data 46,41,4c,53,43,48,20,92, 87
ccd0  4110 data 01,00,0d,4e,41,4d,45,20, da
ccd8  4120 data 44,45,53,20,46,49,4c,45, 36
cce0  4130 data 53,20,3a,20,00,0d,0d,41, 0c
cce8  4140 data 4e,5a,41,48,4c,20,4c,49, 1c
ccf0  4150 data 4e,4b,2d,42,4c,4f,45,43, ed
ccf8  4160 data 4b,45,28,24,29,3a,00,0d, ed
cd00  4170 data 0d,0d,54,52,41,43,4b,00, 4b
cd08  4180 data 20,53,45,43,54,4f,52,0d, 19
cd10  4190 data 00,0d,0d,0d,44,49,53,4b, f6
cd18  4200 data 20,45,49,4e,47,45,4c,45, df
cd20  4210 data 47,54,20,3f,00,00,05,9e, 3c
```

```
60000 for a=49152 to 52512 step 8:s=0:c=0:for b=0 to 7:gosub 60040   4036
60010 s=s+h+1-c:c=s>255:s=255-s and 255:poke a+b,h+1:next b:gosub 60040   5034
60020 f$="pruefsumme ok":if s<>h+1 then f$="[rvs]pruefsummenfehler":f=f+1   5699
60030 gosub 60090:next a:print "[down]"f"fehler":end   2267
60040 read h$:gosub 60050:h=1*16:h$=mid$(h$,2)   2715
60050 l=asc(h$+"x")-48:if l<0 or l>22 then 60080   3198
60060 if l<10 then return   1044
60070 l=l-7:if l>9 then return   1334
60080 f$="[rvs]eingabefehler":l=0:f=f+1   2460
60090 print "zeile" peek(64)*256+peek(63)f$:return   2726
```

saven des programms nach starten des loaders mit

```
poke 43,  0:poke 44,192
poke 45, 40:poke 46,205
save "name",8
```

DMON zeigt, was auf Disketten gespeichert ist.

WWW.HOMECOMPUTERWORLD.COM

HELP hilft beim Fehler suchen

System: C64
Typ: Utility
Sprache: ASM

Die mühsame Fehlersuche hat ein Ende. HELP zeigt einen Fehler, sobald er aufgetreten ist, in der Basic-Zeile invers an.

Um Fehler genau zu lokalisieren, arbeitet HELP in drei Ebenen mit dem Commodore 64 zusammen. Die jeweiligen Programmteile sind über Betriebssystemvektoren in das Betriebssystem eingebaut.
Die benutzten Vektoren sind:

1. Fehlermeldung 768/769
2. Basiceingabe 770/771
3. Tokenumwandlung 772/773

Am Programmanfang steht eine, auf den ersten Blick etwas umständlich programmierte, Initialisierungsroutine. Es wird immer erst der Vektor ausgelesen und dann in die entsprechende Stelle im Programm geschrieben. Das hat den Vorteil, daß HELP auch mit anderen Erweiterungen funktioniert, da HELP nicht stur ins Betriebssystem zurückspringt, sondern zu der Speicherstelle, die vorher durch den Vektor angesprochen wurde.

Zu 1: Fehlermeldungen
In diesem Programmteil überprüft HELP, was für eine Fehlermeldung ausgegeben werden soll. Steht im X-Register 128, so soll nur READY ausgegeben werden. Dann wird der Rest der Routine weggelassen, da ja keine Fehlermeldung ausgegeben werden soll und auch kein Fehler vorliegt. Nun wird noch getestet, ob der Fehler im Direktmodus aufgetreten ist. Im Direktmodus steht in der Speicherstelle 157 eine 128. Ist nun ein Fehler aufgetreten, so wird die aktuelle Zeilennummer aus 57/58 nach 20/21 gerettet. Außerdem wird das Flag in Speicherstelle 2 auf 255 gesetzt, damit der Routine, die in den Basiceingabevektor eingebunden ist, mitgeteilt werden kann, daß ein Fehler aufgetreten ist.

Zu 2: Basiceingabe
Es wird das Flag in Speicherstelle 2 getestet. Ist es gelöscht, so wird die Routine wieder verlassen. Sonst wird die Anfangsadresse der Basiczeile berechnet, deren Zeilennummer in 20/21 steht. Das besorgt die Betriebssystemroutine ab 42515. Dann wird das Flag gelöscht und die Zeile gelistet. Danach geht es bei der Speicherstelle weiter, auf die der Vektor zeigte.

Zu 3: Token-Umwandlung
In den Speicherstellen $5F/60 steht ein Zeiger in die zu listende Zeile. Im Y-Register steht ein Offset, der mit jedem gelisteten Zeichen erhöht wird. Es wird nun das HI-Byte dieses Zeigers in das X-Register gebracht. Das Lo-Byte setzt sich aus dem Offset im Y-Register und dem Wert von der Speicherstelle $5F zusammen. Außerdem wird der Offset um eins erhöht, damit das fehlerhafte Zeichen rechtzeitig erkannt wird. Nun wird der Wert von Akku und X-Register mit dem Zeiger in den Basictext, der auf den Fehler zeigt, verglichen. Sind sie ver-

```
c000    10 data ad,00,03,ac,01,03,8d,5a, 35
c008    20 data c0,8c,5b,c0,a9,43,a0,c0, 15
c010    30 data 8d,00,03,8c,01,03,ad,02, ad
c018    40 data 03,ac,03,03,8d,6b,c0,8c, ad
c020    50 data 6c,c0,a9,5c,a0,c0,8d,02, 64
c028    60 data 03,8c,03,03,ad,06,03,ac, 75
c030    70 data 07,03,8d,95,c0,8c,96,c0, 06
c038    80 data a9,6d,a0,c0,8d,06,03,8c, 1a
c040    90 data 07,03,60,e0,80,f0,12,a5, 82
c048   100 data 3a,c9,ff,f0,0c,a5,39,a4, 7d
c050   110 data 3a,85,14,84,15,a9,ff,85, 2c
c058   120 data 02,4c,00,c0,a5,02,f0,0a, 7f
c060   130 data 20,13,a6,a9,00,85,02,4c, 3c
c068   140 data c9,a6,4c,00,c0,a6,60,18, d1
c070   150 data c8,98,65,5f,90,01,e8,c5, e7
c078   160 data 7a,d0,0c,e4,7b,d0,08,a9, de
c080   170 data 12,85,c7,85,03,d0,0a,a5, 68
c088   180 data 03,f0,06,a9,00,85,c7,85, 2f
c090   190 data 03,88,b1,5f,4c,00,c0,00, d9

60000 for a=49152 to 49296 step    4294
      8:s=0:c=0:for b=0 to 7:
      gosub 60040
60010 s=s+h+1-c:c=s>255:s=255-s    5034
      and 255:poke a+b,h+1:next
      b:gosub 60040
60020 f$="pruefsumme ok":if s<>    5699
      h+1 then f$="[rvs]
      pruefsummenfehler":f=f+1
60030 gosub 60090:next a:print "   2267
      [down]"f"fehler":end
60040 read h$:gosub 60050:h=1*     2715
      16:h$=mid$ (h$,2)
60050 1=asc (h$+"x")-48:if 1<0     3198
      or 1>22 then 60080
60060 if 1<10 then return          1044
60070 1=1-7:if 1>9 then return     1334
60080 f$="[rvs]eingabefehler":1=   2460
      0:f=f+1
60090 print "zeile" peek (64)*     2726
      256+peek (63)f$:return
```

saven des programms nach starten des loaders mit

```
poke 43,  0:poke 44,192
poke 45,151:poke 46,192
save "name",8
```

HELP hilft allen Basic-Programmierern

WWW.HOMECOMPUTERWORLD.COM

schieden, wird noch Flag 2 getestet, um einen eventuell eingeschalteten Reversmodus auszuschalten. Nun wird das Y-Register um eins erniedrigt, weil es am Anfang um eins erhöht worden war und das Betriebssystem die alten Werte benötigt, um korrekt weiterzuarbeiten. Sind die beiden Zeiger gleich, wird der Reversmodus eingeschaltet und Flag2 gesetzt. Anschließend werden noch der Akku und das Y-Register berichtigt.

Noch etwas dazu, wie HELP mit dem Betriebssystem zusammenarbeitet. Der erste Teil (Main 1) überprüft nur, ob im Programmlauf ein Fehler aufgetreten ist. Wenn dies so war, wird die aktuelle Zeilennummer aus den Speicherstellen 57/58 in die Speicherstellen 20/21 gerettet und das Flag gesetzt, damit der Programmteil Main 2 weiß, daß ein Fehler auftrat. Main 2 ist in die Basiceingabewarteschleife eingebunden und wird nur aktiv, wenn dieses Flag gesetzt ist. Dann wird die Startadresse der fehlerhaften Zeile berechnet. Dazu muß die Zeilennummer in den Speicherstellen 20/21 gespeichert sein, das hat Main 1 schon besorgt. Den Rest erledigt eine Betriebssystemroutine ab 42515. Anschließend wird die fehlerhafte Zeile noch gelistet. Das geschieht von Main 2 aus. Hier greift Main 3 ein, der Programmteil, der in die Listroutine eingebunden ist. Um die Zeile zu listen, muß die Startadresse der Zeile in den Speicherstellen 95/96 stehen; das hat Main 2 schon durch den Aufruf der Betriebssystemroutine ab 42515 erreicht. Nun wird noch der Zeiger in die fehlerhafte Zeile (95/96 plus den Offset im Y-Register) ins Lo/Hi-Format gebracht, damit er mit dem Zeiger in den Basictext verglichen werden kann. Bei Übereinstimmung wird das Zeichen revers ausgegeben.

Main 2 stellt fest, ob ein Fehler auftrat und rettet die Zeilennummer. Main 2 berechnet die Anfangsadresse der Zeile und läßt diese vom Betriebssystem listen.

In die Listroutine greift Main 3 ein und gibt das Zeichen, bei dem der Fehler auftrat, revers aus. Das Programm ist als kombinierter Hexdump-Basiclader abgedruckt.

Wie man mit dieser Art Listing umgeht, steht auf Seite 8. Falls Sie den Checker nutzen, müssen sie C000 als Startadresse eingeben.

Geben Sie das Programm als Basic-Lader ein, so speichern Sie die Datas ab, starten den Lader mit RUN und speichern das fertige Maschinenspracheprogramm wie am Ende des Listings angegeben.

Laden mit: LOAD „name",8,1.
Starten mit SYS 49152

(Frank Schusdziarra)

```
1000 -;           HELP 1.3
1010 -;
1020 -;   (C) BY FRANK SCHUSDZIARRA
1030 -;           ROBERT-KOCH-STR. 90
1040 -;           2350 NEUMUENSTER
1050 -;
1060 -;
1070 -.BA $C000
1080 -.EQ ANFANG = 42515
1090 -.EQ LISTIN = 42697
1100 -.EQ NUM    =    57
1110 -.EQ VEK1   =   768
1120 -.EQ VEK2   =   770
1130 -.EQ VEK3   =   774
1140 -.EQ ZEILE  =    20
1150 -.EQ DUMMY  = 49152
1160 -.EQ FLAG   =     2
1170 -.EQ MODE   =    58
1180 -;
1190 -;
1200 -;
1999 -;-------------------------------------
2000 -INIT      LDA VEK1        ;VEK1 LESEN
2010 -          LDY VEK1+1      ;UND AN DAS
2020 -          STA END1+1      ;ENDE VON MAIN1
2030 -          STY END1+2      ;SCHREIBEN
2040 -          LDA #<(MAIN1)   ;ANFANG VON
2050 -          LDY #>(MAIN1)   ;MAIN1 ALS
2060 -          STA VEK1        ;NEUEN VEKTOR
2070 -          STY VEK1+1      ;
2080 -          LDA VEK2        ;EBENSO MIT
2090 -          LDY VEK2+1      ;VEK2, MAIN2
2100 -          STA END2+1      ;VERFAHREN
2110 -          STY END2+2
2120 -          LDA #<(MAIN2)
2130 -          LDY #>(MAIN2)
2140 -          STA VEK2
2150 -          STY VEK2+1
2160 -          LDA VEK3        ;UND AUCH MIT
2170 -          LDY VEK3+1      ;VEK3, MAIN3
2180 -          STA END3+1
2190 -          STY END3+2
2200 -          LDA #<(MAIN3)
2210 -          LDY #>(MAIN3)
2220 -          STA VEK3
2230 -          STY VEK3+1
2240 -          RTS             ;INIT ENDE
2999 -;-------------------------------------
3000 -MAIN1     CPX #$80        ;NUR READY-?
3010 -          BEQ END1        ;JA,ENDE
3020 -          LDA NUM+1       ;DIREKT-
3030 -          CMP #$FF        ;MODUS ?
3040 -          BEQ END1        ;JA, ENDE
3050 -          LDA NUM         ;ZEILENNUMMER
3060 -          LDY NUM+1       ;LADEN UND
3070 -          STA ZEILE       ;RETTEN
3080 -          STY ZEILE+1     ;
3090 -          LDA #$FF        ;FLAG FUER
3100 -          STA FLAG        ;MAIN2 SETZEN
3110 -END1      JMP DUMMY       ;ZUM BETRIEBSSYSTEM
3999 -;-------------------------------------
4000 -MAIN2     LDA FLAG        ;FLAG GESETZT
4010 -          BEQ END2        ;NEIN, ENDE
4020 -          JSR ANFANG      ;ANFANGSADR. BERECHNEN
4030 -          LDA #$00        ;FLAG
4040 -          STA FLAG        ;LOESCHEN
4050 -          JMP LISTIN      ;ZEILE LISTEN
4060 -END2      JMP DUMMY       ; INS BETRIEBSSYSTEM
4999 -;-------------------------------------
5000 -MAIN3     LDX $60         ;XR MIT ZEIGER LADEN
5010 -          CLC             ;OFFSET (YR)
5020 -          INY             ;ERHOEHEN UND
5030 -          TYA             ;ZU $5F (ZEIGER)
5040 -          ADC $5F         ;ADDIEREN
5050 -          BCC COMPARE     ;UEBERTRAG ?
5060 -          INX             ;JA, XR +1
5070 -COMPARE   CMP $7A         ;MIT $7A/7B VER-
5080 -          BNE CONT1       ;GLEICHEN
5090 -          CPX $7B         ;<>, DANN
5100 -          BNE CONT1       ;ZU CONT1
5110 -          LDA #$12        ;SONST, REVERS-
5120 -          STA $C7         ;MODUS EIN
5130 -          STA $03         ;FLAG2 SETZEN
5140 -          BNE CONT2       ;UNBEDINGTER SPRUNG
5150 -CONT1     LDA $03         ;FLAG2 TESTEN
5160 -          BEQ CONT2       ;=0, ENDE
5170 -          LDA #$00        ;<>0, REVERSMODUS AUS
5180 -          STA $C7         ;
5190 -          STA $03         ;FLAG2 LOESCHEN
5200 -CONT2     DEY             ;ENDE, YR UND AKKU
5210 -          LDA (95),Y      ;WERDEN BERICHTIGT
5220 -END3      JMP DUMMY       ;INS BETRIEBSSYSTEM
```

HELP als Assembler-Source-Code

WWW.HOMECOMPUTERWORLD.COM

Floppy auf die Sprünge geholfen mit dem Boot-Sector Manager

System: C128/Floppy 1570/71
Typ: Utility
Sprache: Basic

Um eigene, frei wählbare Boot-Sektoren zu entwickeln, war es bisher notwendig, einen Disk-Monitor zu benutzen. „Boot-Sector Manager" vereinfacht den Umgang mit dieser Autostart-Funktion.

Boot-Programme entwerfen wird mit BSM zum Kinderspiel. Er ermöglicht folgende Arbeiten:

- Erstellen beliebiger Boot-Sektoren (Block 1,0 einer Disk).
- Erstellen von nachfolgenden Boot-Sektoren (Sektoren 1,1 und folgende) die der Sektor 1,0 nachlädt (maximal 100).
- Kopieren des Boot-Sektor und maximal 100 'Folgeboot-Sectoren' auf andere Disketten.
- Wahlweises Arbeiten mit einer oder zwei Floppy-Stationen (Gerät 8+9), wobei es gleich ist ob man auf eine 1541 oder 1571 Floppy zugreift.
- 40- oder 80-Zeichen-Modus. Das Programm arbeitet sowohl mit einem 40-Zeichen-Monitor/Fernseher als auch mit einem 80-Zeichen-Monitor.

Durch die letzten beiden Punkte ist das Programm von Besitzern eines C 128, die ihr System nicht voll ausgebaut haben, nutzbar.

Nach Einladen und Starten des Programmes wählt man anhand der Tasten F1 bis F7 die gewünschte Funktion. STOP/RESTORE ist gesperrt; eventuelle Fehler werden abgefangen.

Im Menüpunkt F3 will der C 128 wissen, ob man mit einer oder zwei Floppies kopieren will. Nach Kopieren des Sektors 1,0 entscheidet man sich, ob Folgeboot-Sektoren kopiert werden. Wichtig: Maximal 100 Folgebootsektoren werden kopiert!

Menüpunkt F5 ermöglicht das Erstellen von Folgeboot-Sektoren. Es muß ein File existieren, das zu Folgeboot-Sektoren umgewandelt wird. Es ist möglich, das File auf einer anderen Diskette zu haben, auf der die Folgeboot-Sektoren stehen sollen. Das Programm verlangt nach Start der Umwandlung die richtige Eingabe des File-Namens, alles weitere geschieht selbständig.

Über F1 erreicht man die Boot-Sektor-Erstellung. Hier werden vom Anwender entsprechende Eingaben gefordert, die das Aussehen des Sektors bestimmen. Der Sektor wird zunächst im Computer komplett erstellt und dann in Block 1,0 eingetragen.

Zu Beginn werden die Bytes 0 bis 6 im Sektor, — die wichtigsten Daten —, angefordert. Automatisch wird vom Programm die Kennung „CBM" in Byte 0 bis 2 eingerichtet. Anschließend wird nach Folgeboot-Sektoren gefragt. Soll der Boot-Sektor keine Folgeboot-Sektoren nachladen, werden die Bytes 3, 4 und 6 automatisch auf '0' gestellt. Man muß dann nur die gewünschte RAM/ROM-Konfiguration des C 128 für Byte 5 eingeben. Sollen weitere Sektoren ab Block 1,1 nachgeladen werden, so ist zunächst in Dezimal! die Anzahl der nachzuladenden Sektoren und anschließend die Adresse im Computer, ab der diese Daten abgelegt werden, anzugeben (hexadezimale Schreibweise). Diese Eingaben befinden sich im Boot-Sektor in Bytes 3, 4 und 6.

Von diesen 7 Bytes hängt es ab, ob der Boot-Sektor richtig funktioniert. Man hat die Möglichkeit, Daten zu verändern oder im Programmablauf weiterzumachen.

Anschließend wird über Textausgaben entschieden. Falls ja, den Text, der in der Meldung „BOOTING. . ." erscheinen soll, eingeben. Durch Eingabe eines Hochkommas (") und anschließender Steuercodes (CTRL/Zahl, etc.) beeinflußt man die Ausgaben des Textes auf dem Bildschirm.

Bei entsprechenden Eingaben ist der Boot-Sektor in der Lage, ein auf der Disk befindliches File (Programm) zu laden. Hierbei richtet sich der Computer nach der Ladeadresse, die im File abgespeichert ist. Gibt man bei der Frage, ob ein Programm gebootet werden soll, „J" und anschließend einen Programmnamen ein, wird völlig unabhängig von etwa vorhandenen Folgeboot-Sektoren das Programm geladen.

Im Anschluß an diese Eingaben bleibt es vollständig dem Anwender überlassen, wie weiter verfahren wird. Der Computer erwartet im Boot-Sektor ein Assemblerprogramm mit dem Anwendungen gesteuert werden.

Über ein Auswahlmenü wählt man, ob ein bereits erstelltes Assemblerprogramm in den Boot-Sektor geladen wird oder bereits festgelegte Verfahrensweisen (Sprung in 64er oder 128er Direktmodus) vom Boot-Sektor ausgeführt werden.

Danach ist der Boot-Sektor innerhalb des Computers erstellt und wird auf der gewünschten Diskette abgespeichert.

R. Pieroth

WWW.HOMECOMPUTERWORLD.COM

```
10 bank 0:poke 792,125:clr :                                    5416
   bs=0:bank 15:color 0,1:color 4,1
20 if rwindow (2)=80 then color 6,1:fast                        2074
   :bs=20
30 print chr$ (27);"m":dim a1%(25500):                          3943
   trap 2790:g1=8:g2=8
40 sys 51748:color 5,14:scnclr :char 1,8+                       5076
   bs,6,"[23cbm @]"
50 char ,8+bs,7,"[2spaces]boot-sector                           3551
    manager[2spaces]",1
60 char ,5+bs,8,"[4spaces]von robert f.                         3063
    pieroth[4spaces]",1
70 char ,5+bs,9," peterstr. 4[2spaces]                          3937
   6052 muehlheim ",1
80 color 5,16:print                                              922
90 :
100 print :print tab(122-bs)"'f1' zum                           5072
    erzeugen eines boot-sectors"
110 print tab(42-bs)"'f3' fuer kopieren                         5300
    von boot-sectoren"
120 print tab(42-bs)"'f5' macht                                 4883
    nachfolgende bootsectoren"
130 print tab(49-bs)"'f7' fuer                                  3793
    programmende"
140 key 1,"1":key 3,"2":key 5,"3":key 7,                        4009
   "4":get key a$:color 5,14
150 on val (a$) goto 700,1820,2300,2690:                        2948
   goto 140
160 :
170 ::::::::::::::::::::::::::::::::::::::
   ::::::::::::::::::::::::::
180 rem :::unterprogramme fuer diskfehler
   & block ,-read,-test,-write
190 :
200 if ds=0 then return :else color 5,2:                        3875
   print "disk-fehler:[2spaces]";ds$
210 color 5,14:print :print "taste                              4195
    druecken":get key n$:gosub 220:goto
   40
220 close 1:close 2:close 3:close 15:                           1654
   return
230 :
240 window 0+bs,2,39+bs,24,1:print tab(                         6021
   240)"[36cbm @]"
250 print "[rvs]lesen von sector"t"[left],                       4159
   "s"[left] - bitte warten[off]":print
260 :
270 gosub 220:open 15,g1,15,"i0":gosub 200                      2734
280 open 2,g1,2,"#":print#15,"ua";2;0;t;s:                      3404
   gosub 200
290 print#15,"b-p";2;0                                          1352
300 :
310 if lf$="j" and g1=g2 then return                            1775
320 for i4=0 to 255:get #2,bt$:a1%(i4)=asc                      5013
   (bt$+chr$ (0)):if i4=6 then fs$=bt$
330 next :gosub 220                                              486
340 :
350 :
360 scnclr :if g1=g2 or m$="j" then begin                       3719
    :print :print :color 5,16
370 :print :print "zieldiskette einlegen.                       5887
    geraet:";g2:print "taste druecken"
380 :get key c1$:color 5,14:bend                                1872
390 :
400 :
410 window 0+bs,2,39+bs,24:print tab(240)"                      5512
   [36cbm @,3spaces]"
420 print "[rvs]testen von block"t"[left],                      4200
   "s"[left] - bitte warten[off,3spaces]"
    :print
430 gosub 220:open 1,g2,15,"i0":gosub 200                       2583
440 open 3,g2,3,"#":print#1,"ua";3;0;t;s:                       3242
   gosub 200
450 print#1,"b-p";3;0:get #3,t$,s$                              2413
460 if t$="" or asc (t$)>40 and s$="" or                        3293
   asc (s$)=1 then 510
470 color 5,2:print :print :print "block                        4825
    ist bereits belegt!!":color 5,16
480 print :print "alte daten                                     3195
    ueberschreiben? (j/n)"
490 get key l$:if l$="n" then gosub 220:                        3873
   goto 630:else if l$<>"j" then 490
500 :
510 scnclr :print#1,"b-f";0;t;s                                 2334
520 color 5,14:print tab(240)"[39cbm @]"                        5337
530 print "[rvs]schreiben von block"t"                          4331
   [left],"s"[left] - bitte warten[off]":
   print
540 print#1,"b-p";3;0:if lf$="j" and g1=                        4353
   g2 or fe$<>"" then return
550 :
560 for i3=0 to 255:print#3,chr$                                2844
   (a1%(i3));:next
570 :
580 print#1,"ub";3;0;t;s:gosub 200                              2446
590 print#1,"b-a";0;t;s:gosub 200:gosub                         2916
   220
600 if lf$="j" or fe$<>"" then return                           1455
610 :
620 color 5,16:print :print :print "sector                      3981
    ist installiert."
630 print :print "auf weiterer disk                             4417
    speichern? (j/n)":color 5,14
640 get key m$:if m$="j" then 360                               1560
650 if m$<>"n" then 640:else m$="":return                       2738
660 :
670 ::::::::::::::::::::::::::::::::::::::
   ::::::::::::::::::::::::
680 rem ::::::::::::::::::::::::::::::::::
   :: boot-sector erzeugen
690 :
700 scnclr :print tab(bs)"[40cbm @]"                            4571
710 char ,bs,1,"aufbau des boot-sector                          5379
    (spur 1, sect. 0)",1:color 5,16
720 print tab(160+bs)"sind eingaben in                          5775
    hexcode notwendig, gibt"
730 print tab(bs)"das programm diese vor.                       5325
    es ist dann[2spaces]'$'"
740 print tab(bs)"vorangestellt."                               2073
750 print tab(24+bs)"beispiel: $ff2a":                          2960
   print
760 :
770 a1%(0)=67:a1%(1)=66:a1%(2)=77                               2735
780 :
790 char ,bs,10,"sind folgebootsectoren                         4888
    vorgesehen? (j/n)"
800 get key a$:if a$="n" then 890:else if                       2463
   a$<>"j" then 800
810 :
820 sys 51748:char ,bs,12,"anzahl[6spaces]                      4435
   <":window 6+bs,12,11+bs,12:input b$
830 if val (b$)>255 then 820:else a1%(6)=                       2918
   val (b$)
840 :
850 sys 51748:char ,bs,14,"an welche                            4501
    adresse zu laden[2spaces]$0000"
860 window 26+bs,14,33+bs,14:input c$:if                        4713
   left$ (c$,1)<>"$" then 850
870 if len (c$)=5 then a1%(4)=dec ( mid$                        5269
   (c$,2,2)):a1%(3)=dec ( right$ (c$,2)):
   else 850
880 :
890 sys 51748:char ,bs,16,                                      6077
   "speicherkonfiguration[2spaces]$00":
   window 21+bs,16,27+bs,16
900 input d$:if left$ (d$,1)="$" and len                        4670
   (d$)=3 then a1%(5)=dec ( right$ (d$,
   2)):else 890
910 :
920 window 0+bs,9,39+bs,20,1:print "                            5023
   [cbm 3,40shift *]"
930 print "[cbm 8]dies sind die ersten 7                       4784
    bytes im sector:":print
940 print "[grn]byte 0 to 2: [cbm 8]$";                       7118
   right$ ( hex$ (a1%(0)),2)" $";right$
   ( hex$ (a1%(1)),2)" $";right$ ( hex$
   (a1%(2)),2)" (c b m)"
950 print "[grn]byte 3: [cbm 8]$";right$                      5331
   ( hex$ (a1%(3)),2)" (lowbyte
    ladeadresse c128)"
960 print "[grn]byte 4: [cbm 8]$";right$                      4655
   ( hex$ (a1%(4)),2)" (highbyte
    ladeadresse)"
```


WWW.HOMECOMPUTERWORLD.COM

```
970 print "[grn]byte 5: [cbm 8]$";right$       5527
    ( hex$ (a1%(5)),2)"
    (konfigurationsbyte)"
980 print "[grn]byte 6: [cbm 8]$";right$       5525
    ( hex$ (a1%(6)),2)" (anzahl
    folgebootsectoren)"
990 print :print "alles richtig? dann          2895
    taste 'j'"
1000 print "zurueck zum anfang mit 'z'"        2610
1010 get key e$:if e$="z" then 40:else if      2537
    e$<>"j" then 1010
1020 :
1030 window 0+bs,10,39+bs,24,1:print "wird     6185
    textausgabe gewuenscht? (j/n)"
1040 print :get key f$:if f$="n" then 1080:    4175
    else if f$<>"j" then 1040
1050 print "text[35spaces]<":window 4+bs,      4233
    12,38+bs,12,1
1060 input g$                                   419
1070 :
1080 window 0+bs,10,39+bs,20,1:print "soll     6493
    ein programm gebootet werden? (j/n)"
1090 get key h$:if h$="n" then 1130:else if    3056
    h$<>"j" then 1090
1100 print :print "prg-name[19spaces]<":       4208
    window 8+bs,12,26+bs,12
1110 input i$                                   423
1120 :
1130 if g$="" and i$="" then a1%(7)=0:         4280
    a1%(8)=0:i2=9:goto 1260
1140 if g$="" then a1%(7)=0:i1=8:goto 1220     2835
1150 :
1160 for i1=7 to len (g$)+6                    1634
1170 a1%(i1)=asc ( mid$ (g$,i1-6,1))           2113
1180 next :a1%(i1)=0:i1=i1+1:b=0               2447
1190 :
1200 if i$="" then a1%(i1)=0:i2=i1+1:goto      3266
    1260
1210 :
1220 for i2=i1 to len (i$)+i1-1:b=b+1          2499
1230 a1%(i2)=asc ( mid$ (i$,b,1))              1846
1240 next :a1%(i2)=0:i2=i2+1                   2010
1250 :
1260 window 0+bs,3,39+bs,24,1:color 5,16       2580
1270 print "ueber assembler-befehle muss       3751
    nun gesteu-"
1280 print "ert werden, was der c128           4838
    ausfuehren soll":print
1290 print "anschliessendes menue              3956
    beinhaltet bereits"
1300 print "zwei (1. & 2. ) fertige jobs       4139
    zur auswahl":print :print
1310 print "[4spaces]1. einsprung zum c64      3768
    direktmodus":print
1320 print "[4spaces]2. rueckkehr in c128      4035
    direktmodus":print
1330 print "[4spaces]3. assembler-code von     3941
    disk laden":print
1340 print "[4spaces]4. assembler-programm     4015
    [2spaces]erstellen":print
1350 print "[4spaces]5. neustart boot-         4122
    sector[2spaces]manager":print :color
    5,14
1360 print "[4spaces]zur auswahl               3390
    zahlentaste druecken"
1370 get key i1$:on val (i1$) goto 1390,       4407
    1420,1440,1540,40:goto 1370
1380 :
1390 restore 1400:for i3=i2 to i2+2:read       3500
    ab:a1%(i3)=ab:next :goto 1710
1400 data 76,77,255                            1094
1410 :
1420 a1%(i2)=96:goto 1710                      1579
1430 :
1440 scnclr :print :print "zum ruecksprung     4408
    [2spaces]'↑' eingeben!":print
1450 print "file-name[18spaces]<":window 9+    3983
    bs,6,25+bs,6,0:input j$
1460 if asc (j$)=94 then 1260                  1530
1470 :
1480 window 0+bs,20,39+bs,22:gosub 220:open    3650
    2,8,0,j$:gosub 200
```

```
1490 :
1500 print "daten werden gelesen":get #2,      4673
    by$,by$:do while i2<256
1510 get #2,by$:a1%(i2)=asc (by$+chr$ (0)):    3691
    i2=i2+1
1520 loop :gosub 220:goto 1710                 1414
1530 :
1540 scnclr :color 5,16:print "erstellen       5036
    eines assembler-programms muss"
1550 print "von adresse $1300 bis maximal      5086
    $" hex$ (4864+255-i2)" vor-"
1560 print "genommen werden, da sonst keine    4621
    funktion"
1570 :
1580 print :color 5,4:print "eingebauter       5317
    maschinesprache-monitor c128"
1590 bank 0:window 0+bs,9,31+bs,24:poke        5798
    2560,13:poke 2561,67:color 5,4
1600 print chr$ (27);"l";chr$ (27);"r":poke    5177
    842,147:poke 843,68:poke 844,49
1610 poke 845,51:poke 846,48:poke 847,48:      5345
    poke 208,6:poke 808,110:monitor
1620 print chr$ (27);"n";chr$ (27);"m":poke    6090
    2560,3:poke 2561,64:poke 808,112:bank
    15
1630 :
1640 window 0+bs,9,39+bs,24:color 5,14:        2531
    scnclr
1650 print "endadresse des programmes          5867
    [2spaces]$13??[8spaces]":window 25+bs,
    9,31+bs,9
1660 input ep$:if len (ep$)<>5 or asc (ep$)    4697
    <>36 then 1640:else ep$=right$ (ep$,4)
1670 if dec (ep$)>(4864+255-i2) or dec         3877
    (ep$)<4864 then 1590
1680 for i4=4864 to dec (ep$):a1%(i2)=peek     4054
    (i4):i2=i2+1:next
1690 :
1700 :
1710 window 0+bs,0,39+bs,24,1:print tab(       5009
    240)"[25cbm @]"
1720 char ,0,7,"boot-sector ist aufgebaut",    4727
    1:print :color 5,16
1730 print :print "disk in floppy '"g2"'       4063
    einlegen, um"
1740 print "die daten in block 1,0             3766
    einzuschreiben!":print
1750 color 5,2:print "taste druecken oder      5255
    '↑' fuer ruecksprung"
1760 get key k$:scnclr :if k$="↑" then 40      1983
1770 color 5,14:t=1:s=0:gosub 410:goto 40      2014
1780 :
1790 :::::::::::::::::::::::::::::::::::::::::
    ::::::::::::::::::
1800 rem ::::::::::::::::::::::::::::::::::bo
    ot-sector kopieren
1810 :
1820 window 0+bs,4,39+bs,24,1:print "          5721
    [36cbm @]"
1830 char ,0,1,"kopieren des boot-sector       5596
    (block 1,0)",1:color 5,16:print
1840 print :print :print :print "1             3773
    quelldisk in floppy:";g1:print
1850 print "2 zieldisk in floppy: ";g2:        2493
    print
1860 print "3 kopieren beginnen":print         2679
1870 print "4 zurueck zum startmenue":print    4071
    :color 5,14
1880 print "zahlentaste waehlen"               2239
1890 get key z$:on val (z$) goto 1900,1910,    3429
    1930,40:goto 1890
1900 gosub 2440:goto 1890                      1296
1910 gosub 2480:goto 1890                      1328
1920 :
1930 t=1:s=0:gosub 240:color 5,16:scnclr :     4665
    if fs$="" or asc (fs$)>100 then 2230
1940 print :print "der boot-sector laedt"      4439
    asc (fs$)"folgeboot-"
1950 print "sectoren nach!":print :print       6034
    "sollen diese auch kopiert werden?
    (j/n)"
1960 get key lf$:if lf$="n" then 2230:else     3960
    if lf$<>"j" then 1960
```


WWW.HOMECOMPUTERWORLD.COM

```
1970 :
1980 window 0+bs,0,39+bs,24,1:print :print      2668
     :print :print
1990 print "quelldiskette einlegen! floppy:     4249
     ";g1:af=asc (fs$)
2000 print :print "taste druecken":get key      2729
     g1$
2010 :
2020 if g1<>g2 then begin :print :print         5588
     "zieldiskette einlegen! floppy:";g2
2030 :print :print "taste druecken":get key     3944
     g2$:color 5,14:for i5=1 to af
2040 :s=s+1:if s=21 then s=0:t=t+1              2870
2050 :gosub 240                                  449
2060 :next :goto 2230                            704
2070 bend                                        304
2080 :
2090 color 5,14:poke 808,110                    1543
2100 for i5=1 to af:fi%=(i5-1)*256              2354
2110 :s=s+1:if s=21 then s=0:t=t+1              2870
2120 :gosub 240:for i6=0 to 255:get #2,bt$:     4387
     a1%(i6+fi%)=asc (bt$+chr$ (0)):next
2130 next :gosub 220                             486
2140 :
2150 print :print "zieldiskette einlegen!        4110
      floppy:";g2
2160 print :print "taste druecken":get key      3177
     g2$:s=0:t=1
2170 for i6=1 to af:fi%=(i6-1)*256              2231
2180 :s=s+1:if s=21 then s=0:t=t+1              2870
2190 :gosub 410:for i7=0 to 255:print#3,        3604
     chr$ (a1%(i7+fi%));:next
2200 gosub 580:next                              843
2210 if fe$<>"" then return                      795
2220 :
2230 color 5,16:print :print "kopieren          5857
      beendet - bitte taste druecken":poke
     808,112
2240 get key kv$:lf$="":goto 40                 1357
2250 :
2260 :
2270 rem ::::::::::::::::::::::::::::::::::
     ::::::::::::::::::::::::
2280 rem ::::::::::::::::::::::::::::::folg
     eboot-bloecke einrichten
2290 :
2300 color 5,14:sys 51748:scnclr :print         5625
     tab(bs)"[29cbm @]"
2310 char ,bs,1,"folgeboot-sectoren              4979
      einrichten",1:color 5,16
2320 window 0+bs,3,39+bs,24:print "ein           5153
      programm wird in diesem abschnitt zu"
2330 print "'boot-sectoren' umgewandelt. je     4144
      nach der"
2340 print "laenge wird es in den sectoren      4056
     [2spaces]1,1[2spaces]und"
2350 print "folgende abgelegt, von wo es        3762
      der pc 128"
2360 print "anhand der angaben im boot-         5024
     sector laedt.":print :color 5,14
2370 :
2380 print "1 file-disk in floppy:"g1:print     5414
     :print "2 folgeboot - disk in:"g2
2390 print :print "[grn]3 directory ansehen     5656
      (g1)":print :print "4 umwandlung
      beginnen"
2400 print :print "5 zurueck zum                3596
      startmenue":print :color 5,16
2410 print "gewuenschte funktion? (taste)"      3260
2420 color 5,16:get key fu$:on val (fu$)        5140
     goto 2440,2480,2520,2550,40:goto 2420
2430 :
2440 window 23+bs,9,26+bs,9:if g1=8 then        4611
     print "neun":g1=9:goto 2460
2450 g1=8:print "acht"                           1383
2460 if z$="1" then return :else 2420            1670
2470 :
2480 window 23+bs,11,26+bs,11:if g2=8 then      5287
     print "neun":g2=9:goto 2500
2490 g2=8:print "acht"                           1385
2500 if z$="2" then return :else 2420            1702
2510 :
2520 window 0+bs,0,39+bs,24,1:print chr$        4132
     (27);"l":directory u(g1):print ds$
2530 print :print "taste druecken":print         5312
     chr$ (27);"m":get key d$:color 5,14:
     goto 2300
2540 :
2550 sys 51748:scnclr :print tab(bs)"zum        5097
      ruecksprung '↑' eingeben"
2560 char ,bs,4,"filename":window 9+bs,4,28      4946
     +bs,4,1:input fe$
2570 window 0+bs,0,39+bs,24,1:if len (fe$)>      3563
     16 then 2550
2580 if asc (fe$)=94 then 40:else t=1:s=0        2293
2590 :
2600 gosub 220:open 2,g1,0,fe$:gosub 200:        4173
     get #2,be$,bt$:poke 808,110
2610 for i8=0 to 25500:get #2,bt$:a1%(i8)=       3797
     asc (bt$+chr$ (0))
2620 if (st and 64)<>64 then next :else          5235
     gosub 220:af=int ((i8-1)/255)+1
2630 :
2640 gosub 2150:print :print "job                4543
      erledigt... taste druecken":get key
     xz$:fe$=""
2645 poke 808,112:sys 51748:goto 2300            2314
2650 :
2660 ::::::::::::::::::::::::::::::::::::
     :::::::
2670 rem :::::::::::::::::::::::::: progr
     ammende
2680 :
2690 scnclr :char ,8+bs,2,"[22cbm @]"            3690
2700 char ,8+bs,3,"sind sie sicher? (j/n)",      4517
     1:get key en$
2710 if en$="n" then 40:else if en$<>"j"         2827
     then 2700
2720 key 1,"dload[3ctrl i,3spaces]"+chr$         5032
     (13):key 3,chr$ (147)+"directory"+chr$
     (13)
2730 key 5,"bload[3ctrl i],b0"+chr$ (13):        5158
     key 7,"[clr]list"+chr$ (13):print chr$
     (27);"l"
2740 scnclr :poke 792,64:poke 808,110:end        1891
2750 :
2760 ::::::::::::::::::::::::::::::::::::
     ::::::::::::
2770 rem ::::::::::::::::::::fehlerbehandl
     ungs-routine
2780 :
2790 if er=14 and el=1670 then window 0+bs,      4285
     3,39+bs,24,1:resume 1540
2800 :
2810 sys 51748:color 5,14:scnclr                 1794
2820 if er=5 then begin                          1029
2830 :char ,bs,4,"das geraet ist nicht           3525
      vorhanden!"
2840 :char ,bs,6,"floppy anschalten oder         3995
      anschliessen!"
2850 :char ,bs,8,"fuer neuen programmstart        3932
      taste druecken"
2860 :get key dp$:dclose :resume 40              1182
2870 bend                                        304
2880 :
2890 window 0+bs,21,39+bs,24,1:print err$        4765
     (er);" fehler in ";el:print
2900 print "fuer neustart taste druecken"        2576
2910 get key re$:dclose :resume 40               1550
2920 ::::
```

Auto-Boot Sektoren mit BSM editieren.

WWW.HOMECOMPUTERWORLD.COM

Auf den Track gekommen

System: C64/Laufwerk
Typ: Utility
Sprache: Basic/ASM

„Spurformat" formatiert einzelne Spuren und Sektoren einer Diskette mit verschiedenen Formaten und zeigt damit den meisten Kopierprogrammen eine lange Nase.

Das Programm deckt vier verschiedene Bereiche ab:
— Schnelles Formatieren
— Diskettenname und ID ändern
— Einzelne Spuren reparieren
— Diskette gegen Kopierer schützen.

Nach Start mit RUN präsentiert „Spurformat" sein Menü:
1) Spur x bis y formatieren
2) Spur x bis y mit n Sektoren formatieren
3) Spur x spezialformatieren
4) BAM erzeugen
5) NAME und ID wechseln
6) Disk-Befehle senden
7) Directory
8) Ende

Durch Drücken der Zifferntasten gelangt man in das entsprechende Unterprogramm:

1) Spur x bis y formatieren

Mit diesem Programmpunkt formatiert man sowohl komplette Disketten als auch gezielt einzelne Spuren und Spurbereiche. Schadhafte Spuren werden repariert und damit wieder benutzbar gemacht, ohne die gesamte Diskette zu löschen. „Spurformat" will wissen, welcher Spurbereich (von Spur x bis Spur y) zu formatieren ist. Üblich sind die Spuren 1 bis 35. Jetzt will das Programm die Disketten-ID wissen. Eingestellt sind zwei geshiftete Spaces, CHR$(160). Von Vorteil ist:
— Die Geschwindigkeit. Eine Diskette wird 10 Sekunden formatiert.
— Der Formatbereich: Maximal 41 Spuren erstellt „Spurformat".

2) Spur x bis y mit n Sektoren formatieren

Einfacher Kopierschutz. Die Parameter sind die gleichen wie unter 1, nur muß zusätzlich die Anzahl der Sektoren pro Spur angegeben werden. Normalerweise fassen die einzelnen Spuren aufgrund ihrer von außen nach innen abnehmenden Länge unterschiedlich viele Sektoren:

Spur 1 bis 17 hat 21 Sektoren
Spur 18 bis 24 hat 19 Sektoren
Spur 25 bis 30 hat 18 Sektoren
Spur 31 bis 35 hat 17 Sektoren

Ein Kopierschutzbeispiel: Spur 1 hat normalerweise 21 Sektoren (0 bis 20). Formatiert man Spur 1 mit 22 Sektoren, entsteht eine Überlänge, die den Anfang von Sektor 0 löscht. Eine Floppy, die versucht Sektor 0 zu lesen, findet diesen nicht und zeigt die Fehlermeldung „20,READ ERROR,01,00". Dieser Trick ist leicht umgehbar. Durch die Überlänge entstand zusätzlich Sektor 21. Ein spezielles Programm (BLOCK-EXECUTE) prüft diesen Sektor, in dem wichtige Teile des Programms (Codewort oder ähnliches) stehen. Die wenigsten Kopierprogramme suchen nach solchen versteckten Blökken. Dieser Trick ist ein wirkungsvoller Kopierschutz.

4) BAM erzeugen

Mit dieser Funktion wird die Directory und die Block-Availability-Map (BAM), in der die belegten Blöcke vermerkt sind, generiert. Als Parameter wird der Diskettenname verlangt, der als Überschrift im Directory erscheint. Diese Funktion MUSS jedem Formatierprozeß folgen, der die Spur 18 betrifft, da sich die Directory auf Spur 18 befindet und beim Formatieren gelöscht wird.

5) NAME und ID wechseln

Ändert Namen und ID einer Diskette ohne Datenverlust.

6) Disk-Befehle senden

DOS-Befehle nutzen, um Files zu bearbeiten.

7) Directory

Die Directory-Funktion zeigt auf Wunsch jederzeit den Diskenteninhalt ohne Programmverlust an. Falls die Directory-Funktion unmittelbar nach einem Format nur wirre Zeichen zeigt, in den Menüpunkt „Disk-Befehle senden" gehen und den Initialisierungsbefehl i geben. Danach ist alles wieder in Ordnung.

8) Ende

Führt ins Basic zurück.

3) Spur x spezialformatieren

Dieser Programmpunkt ist kompliziert und steht daher am Schluß. ALLE Parameter des Blockheaders einer Spur werden selbst bestimmt. Ein Kopierschutzbeispiel sind Füllbytes, die nirgends abgefragt werden und in keiner Prüfsumme erscheinen.

K. Bunkert

```
1000 def fn ok(x)=ok and not x or not ok      1733
     and x
1010 poke 53280,14:poke 53281,14              1974
1020 print chr$ (8) chr$ (14)"{wht,clr,       6420
     2right}SPUR-FORMAT v2.2{2spaces}fuer
      C-64/VC-1541"
1030 print "{2right,35cbm y}"                 7312
1040 print "{7right}(C) 1985 by Klaus         3445
      Burkert"
1050 print "{blu,2down,rvs}1{off}...Spur x-   3140
     y formatieren"
1060 print "{down,rvs}2{off}...Spur x-y mit   4162
      n Sekt. formatieren"
1070 print "{down,rvs}3{off}...Spur x         3189
```


WWW.HOMECOMPUTERWORLD.COM

```
      spezial-formatieren"                                    
1080 print "[down,rvs]4[off]...BAM erzeugen"                   2009
1090 print "[down,rvs]5[off]...NAME und ID wechseln"           2640
1100 print "[down,rvs]6[off]...Disk-Befehle senden"           3017
1110 print "[down,rvs]7[off]...Directory"                     1930
1120 print "[down,rvs]8[off]...Ende"                          1236
1130 input "[2down,cbm 6]Bitte waehlen [2spaces]8[3left]";a$  2713
1140 if val (a$)<1 or val (a$)>8 then run                     1851
1150 on val (a$) goto 1170,1200,1870,1380,1450,1760,1720      4042
1160 print "[clr]":end                                        495
1170 print "[blu,clr,down,10right]Spur x-y formatieren"       3832
1180 print "[10right,20cbm y,cbm 6]"                          5695
1190 sys 5486:goto 1260                                       1435
1200 print "[blu,clr,down,4right]Spur x-y mit n Sekt. formatieren"  4327
1210 print "[4right,32cbm y,cbm 6]"                           7180
1220 sys 5486                                                 764
1230 b$="":input "[down]Anzahl Sektoren [4spaces,4left]";b$:b=val ( left$ (b$+"[2shift-space]",2))  5268
1240 if b<1 or b>25 then print "[3up]":goto 1230              1748
1250 open 1,8,15:print#1,"m-w" chr$ (13) chr$ (7) chr$ (2) chr$ (201) chr$ (b);:close 1  5243
1260 s$="":input "[down]von Spur[3spaces]1[3left]";s$:sv=val ( left$ (s$+"[2shift-space]",2))  4175
1270 if sv<1 then print "[3up]":goto 1260                     1663
1280 s$="":input "[down]bis Spur[3spaces]35[4left]";s$:sb=val ( left$ (s$+"[2shift-space]",2))  4612
1290 if sb<1 then print "[3up]":goto 1280                     1585
1300 input "[down]ID";id$:id$=left$ (id$+"[2shift-space]",2)  2768
1310 open 1,8,15                                              935
1320 print#1,"m-w" chr$ (34) chr$ (6) chr$ (1) chr$ (sv);    3583
1330 print#1,"m-w" chr$ (63) chr$ (6) chr$ (1) chr$ (sb);    3567
1340 print#1,"m-w" chr$ (18) chr$ (0) chr$ (2)id$;           2916
1350 print#1,"m-e" chr$ (0) chr$ (6);                        1939
1360 close 1                                                  258
1370 goto 1840                                                623
1380 print "[blu,clr,down,14right]BAM erzeugen"              3423
1390 print "[14right,12cbm y,cbm 6]"                         4630
1400 input "[down]NAME";na$:na$=left$ (na$+"[16shift-space]",16)  4042
1410 open 1,8,15,"i"                                          1273
1420 sys 6157                                                 746
1430 print#1,"u3:"na$                                         1141
1440 goto 1810                                                599
1450 print "[blu,clr,down,10right]NAME und ID wechseln"      3888
1460 print "[10right,20cbm y,cbm 6]"                         5695
1470 open 1,8,15,"i"                                          1273
1480 open 8,8,8,"#"                                           972
1490 print#1,"u1:8 0 18 0"                                    1293
1500 print#1,"b-p:8 144"                                      1305
1510 print "[down,6right]" chr$ (34);                        1715
1520 for i=1 to 16                                            867
1530 get #8,a$:print a$;                                      1423
1540 next i                                                   276
1550 print chr$ (34)"[up]"                                    1083
1560 input "NAME";na$                                         1323
1570 print#1,"b-p:8 144"                                      1305
1580 print#8,na$;                                             912
1590 print#1,"b-p:8 162"                                      1289
1600 print "[down,4right]" chr$ (34);                        1874
1610 for i=1 to 5                                             854
1620 get #8,a$:print a$;                                      1423
1630 next i                                                   276
1640 print chr$ (34)"[up]"                                    1083
1650 input "ID";id$                                           567
1660 print#1,"b-p:8 162"                                      1289
1670 print#8,id$;                                             920
1680 print#1,"u2:8 0 18 0"                                    1325
1690 close 8                                                  272
1700 print#1,"i"                                              659
1710 goto 1810                                                599
1720 print "[blu,clr,down,15right]Directory"                  3435
1730 print "[15right,9cbm y,cbm 6]"                          3863
1740 sys 6052                                                 662
1750 goto 1810                                                599
1760 print "[blu,clr,down,10right]Disk-Befehle senden"       3787
1770 print "[10right,19cbm y,cbm 6]"                          5493
1780 print "[down,9right]" chr$ (34):print "[2up]"           2757
1790 input "Befehl[3spaces,right]i[4left]";be$                2365
1800 open 1,8,15,be$                                          1305
1810 close 1:open 1,8,15                                      1085
1820 input#1,a$,b$,c$,d$:close 1                              1667
1830 print "[blu,down]"a$","b$","c$","d$                      1696
1840 print "[down,cyn]taste druecken"                         2187
1850 poke 198,0:wait 198,1:poke 198,0                         2514
1860 run                                                      138
1870 print "[blu,clr,down,7right]Spur x spezial-formatieren"  4329
1880 print "[7right,26cbm y,cbm 6]"                           6344
1890 input "[2down]welche Spur";sp$:sp=val (sp$+chr$ (0))     3597
1900 if sp<1 then 1870                                        1045
1910 bl=21                                                    584
1920 if sp>17 then bl=19                                      1600
1930 if sp>24 then bl=18                                      1512
1940 if sp>30 then bl=17                                      1392
1950 if sp>35 then bl=17                                      1552
1960 print "[2down]wieviele Sektoren "bl"[5left]";:input bl$:bl=val (bl$+chr$ (0))  5091
1970 if bl<1 or bl>25 then 1870                               1505
1980 i1=160:i2=160:he$="":for s=0 to bl-1                     2953
1990 print "[clr,blu]Spur"sp"[2spaces]Block"s"[2down]"       1880
2000 input "Header-Kennung[2spaces]8[3left]";hk$:hk=val (hk$+chr$ (0))  4887
2010 print "Sektor "s"[4left]";:if s>9 then print "[left]";  3218
2020 input se$:se=val (se$+chr$ (0))                          1894
2030 print "Spur "sp"[4left]";:if sp>9 then print "[left]";  3003
2040 input tr$:tr=val (tr$+chr$ (0))                          2155
2050 print "ID 1 "i1"[4left]";:if i1>9 then print "[left]";  2614
2060 if i1>99 then print "[left]";                            1691
2070 input i1$:i1=val (i1$+chr$ (0))                          2034
2080 print "ID 2 "i2"[4left]";:if i2>9 then print "[left]";  2664
2090 if i2>99 then print "[left]";                            1695
2100 input i2$:i2=val (i2$+chr$ (0))                          2110
2110 input "Fuellbyte 1[2spaces]15[4left]";f1$:f1=val (f1$+chr$ (0))  3493
2120 input "Fuellbyte 2[2spaces]15[4left]";f2$:f2=val (f2$+chr$ (0))  3528
2130 if hk<0 or hk>255 or se<0 or se>255 or tr<0 or tr>255 or i1<0 or i1>255 or i2<0 or i2>255 then 1990  6580
2140 ck=0:ck=fn ck(se):ck=fn ck(tr):ck=fn ck(i1):ck=fn ck(i2)  5208
2150 print "Pruefsumme "ck"[4left]";:if ck>9 then print "[left]";  4175
2160 if ck>99 then print "[left]";                            1528
2170 input ck$:ck=val (ck$+chr$ (0))                          1997
2180 if ck<0 or ck>255 then 1990                              1973
2190 he$=he$+chr$ (hk)+chr$ (ck)+chr$ (se)+chr$ (tr)          3199
2200 he$=he$+chr$ (i2)+chr$ (i1)+chr$ (f1)+chr$ (f2)          3162
2210 next s                                                   296
2220 open 1,8,15:open 2,8,2,"#0":print#1,"b-p:"2,0:print#2,he$;:close 2  4325
2230 sys 6258                                                 766
2240 print#1,"m-w" chr$ (34) chr$ (6) chr$ (1) chr$ (sp)      3341
2250 print#1,"m-w" chr$ (186) chr$ (6) chr$ (1) chr$ (bl)     3046
2260 print#1,"m-e" chr$ (0) chr$ (6)                          1880
2270 close 1:goto 1840                                        629
```

Eigene Kodierschlüssel machen Kopierern das Leben schwer

WWW.HOMECOMPUTERWORLD.COM

Diskette o.k.?

System: C64/Floppy
Typ: Anwendung
Sprache: Basic

DIAN ist ein kleiner Disk-Analyzer. Er prüft Disketten auf defekte Sektoren.

Read Error! Wer kennt das nicht? Ein Programm ist verschwunden. Jetzt heißt es, retten was zu retten ist. Mit diesem Programm lokalisiert man die zerstörten Blöcke. Es liest jeden Block einer Diskette und sortiert so die Blocks mit Read Error 20,21,22,23,27 und 29 heraus. Die Bedienung ist einfach: Programm laden — Testdisk einlegen — los geht's. Die gefundenen Fehler werden mit Track- und Sektornummer auf dem Bildschirm ausgegeben. Ein Durchgang dauert ungefähr 2.30 Minuten. Auf der Testdisk wird nichts verändert.

T. Lippert

```
100 rem *** (c) 1986 thomas lippert ***
110 f=8:sys 65511:poke 53280,11:poke 53281,15   3930
120 rem
130 print "[clr,down,13spaces]disk-analyser":gosub 190   2934
140 open 1,f,15:for t=1 to 35:read q:for s=0 to q:open 5,f,5,"#":print#1,"b-r"5;0;t;s   5652
150 input#1,a,b$,c,d:if a>1 then print b$;   3911
160 close 5:next :next :close 1:gosub 190:print#2,"[down,2spaces]"z" errors located":close 2:end   4307
170 data 20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20   4503
180 data 18,18,18,18,18,18,18,17,17,17,17,17,17,16,16,16,16,16   5840
190 open 15,f,15,"i":close 15:return   1695
```

Ist die Diskette gesund, kräht kein Hund!

Y mal zwei

System: C64/Laufwerk
Typ: Anwendung
Sprache: Basic

„Grafikdoppler" teilt Grafiken in obere und untere Hälften. Eine Hälfte wird verdoppelt und dadurch in Y-Achsenrichtung gestreckt. Ergebnisse sind meist bizarre Ergebnisse.

Zuerst entscheidet man, welche Hälfte gestreckt wird. F1 wählt oben, F3 unten. Danach wartet der Computer auf den Programmnamen der Grafik. Sie muß sich auf der eingelegten Diskette befinden. Jetzt muß man die Grafik-Startadresse angeben. „Grafikdoppler" lädt das Grafikfile, berechnet neue Punkte und speichert das Ergebnis ab (zirka 3 Minuten). Ausdrucke erstellt man mit Hardcopyprogrammen eigener Wahl. Tip:
Um eine Grafik komplett in Y-Richtung zu verdoppeln, wählt man nacheinander F1 und F2, druckt beide neuen Grafiken aus und klebt sie zusammen — fertig.

Peter Müller-Buschbaum

```
10 rem ********************************
20 rem grafik halbierer
30 rem by peter mueller - buschbaum
40 rem      von der goltz - allee 37
50 rem      2300 kiel 1
60 rem      17.11.1985
70 rem ********************************
80 rem
90 rem
100 if f=23 then 400   1006
110 :
120 :
130 rem ********** hauptprogramm ******
140 gosub 1010   571
150 gosub 1270   627
160 rem abfragen
170 print "[clr,down,cyn]"   752
180 print "[down,2right]grafikfilename:";   2572
190 input gf$   333
200 print "[down,2right]neuer name:";   2108
210 input nf$   347
220 if gf$=nf$ then nf$=nf$+".v"   2168
230 if len (nf$)>16 then nf$=left$ (nf$,16)   2778
240 print "[2down,2right]neue anfangsadresse: 8192[6left]";:input na   5170
250 print "[3down,2right]diskette mit zubearbeitender grafik"   3916
260 print "[2right]einlegen und [rvs]return[off] druecken ."   3686
270 get f$:if f$<>chr$ (13) then 270   2378
280 print "[down,2right]ok."   1230
290 rem floppy
300 open 1,8,15,"i":close 1   1530
310 open 1,8,2,""+gf$+",p,r"   1683
320 get #1,g$:if g$="" then g$=chr$ (0)   2734
330 l=asc (g$)   591
```


WWW.HOMECOMPUTERWORLD.COM

```
340 get #1,g$:if g$="" then g$=chr$ (0)      2734
350 h=asc (g$)                                587
360 close 1                                   258
370 ad=1+256*h                               1161
380 f=23                                      524
390 load ""+gf$,8,1                          1123
400 print "[2down,2right]diskette fuer      3381
    halbierte grafik "
410 print "[2right]einlegen und [rvs]       3686
    return[off] druecken ."
420 f$=""                                     395
430 get f$:if f$<>chr$ (13) then 430         2375
440 print "[down,2right]ok ..[3spaces]      3768
    .arbeitszeit 2,8 min.. "
450 h=int (na/256):l=na-256*h                2846
460 open 1,8,15,"i":close 1                  1530
470 open 1,8,2,""+nf$+",p,w"                 1731
480 print#1,chr$ (l);chr$ (h);               1550
490 on qw gosub 580,790                       993
500 close 1                                   258
510 print "[2down,2right]ready ."             1704
520 f$="":f=0                                 748
530 get f$:if f$="" then 530                 1622
540 run                                       138
550 :
560 :
570 rem ** grafikhaelfte berechnen ***
580 rem obere haelfte
590 for x=0 to 3520 step 320                 1397
600 for a=0+x to 4+x step 4                  1511
610 for xx=0 to 312 step 8                   1415
620 for i=0+a+xx to 3+xx+a                   1954
630 print#1,chr$ ( peek (ad+i));             1388
640 print#1,chr$ ( peek (ad+i));             1388
650 next i                                    276
660 next xx                                   403
670 next a                                    260
680 next x                                    306
690 a=3840                                    636
700 for xx=0 to 312 step 8                   1415
710 for i=0+a+xx to 3+xx+a                   1954
720 print#1,chr$ ( peek (ad+i));             1388
730 print#1,chr$ ( peek (ad+i));             1388
740 next i                                    276
750 next xx                                   403
760 return                                    142
770 :
780 :
790 rem untere haelfte
800 for x=3840 to 7360 step 320              1941
810 for a=0+x to 4+x step 4                  1511
820 for xx=0 to 312 step 8                   1415
830 for i=0+a+xx to 3+xx+a                   1954
840 print#1,chr$ ( peek (ad+i));             1388
850 print#1,chr$ ( peek (ad+i));             1388
860 next i                                    276
870 next xx                                   403
880 next a                                    260
890 next x                                    306
900 a=7680                                    700
910 for xx=0 to 312 step 8                   1415
920 for i=0+a+xx to 3+xx+a                   1954
930 print#1,chr$ ( peek (ad+i));             1388
940 print#1,chr$ ( peek (ad+i));             1388
950 next i                                    276
960 next xx                                   403
970 return                                    142
980 :
990 :
1000 rem *** bildschirmmasken ********
1010 poke 53280,3:poke 53281,11:print chr$   3124
     (142) chr$ (147);
1020 print "[cyn,rvs]";:for z=1 to 40:print  2499
     " ";:next
1030 print "[cbm 6] grafik-halbierer [cyn,   4054
     12spaces,grn] 17.11.85 ";
1040 print "[cyn] by peter mueller-          3411
     buschbaum[13spaces]";
1050 print :print                             371
1060 print "[3spaces,cbm 8]dieses programm    3658
      ermoeglicht ihnen
1070 print "[3spaces]ausschnitte von          3774
     grafifseiten anzu-
1080 print "[3spaces]fertigen und diese auf   2948
     diskette
1090 print "[3spaces]abzulegen .              1213
1100 print "[3spaces]durch geschickte wahl    3719
     der satrt-
1110 print "[3spaces]adresse koennen die so   3424
     erstellten
1120 print "[3spaces]grafik-haelften mit      3506
     anderen grafik-
1130 print "[3spaces]programmen bearbeitet    3463
     werden .
1140 print                                    153
1150 print "[3spaces]beispiele fuer           3142
     startadressen:
1160 print "[3spaces]hi-eddi" spc(7)"49664    2831
     ($c200)
1170 print "[3spaces]print shop[4spaces]      2983
     49664 ($c200)
1180 print :print                             371
1190 print "[cyn,rvs]";:for z=1 to 40:print   2499
     " ";:next
1200 for z=1 to 40:print " ";:next            1673
1210 print                                    153
1220 print "[3spaces,cbm 8]weiter mit taste   2141
1230 print                                    153
1240 rem    print"■";:for z=1 to 40:print"
     " ";:next
1250 get qq$:if qq$="" then 1250              1745
1260 return                                   142
1270 poke 53280,11:poke 53281,11:print chr$   3604
     (142) chr$ (147);
1280 print "[cyn,cbm a,38shift *,cbm s]";     3403
1290 print "[shift -,blu,rvs,38spaces,cyn,    2666
     off,shift -]";
1300 print "[shift -,blu,rvs,38spaces,cyn,    2666
     off,shift -]";
1310 print "[shift -,blu,rvs,38spaces,cyn,    2666
     off,shift -]";
1320 print "[shift -,blu,rvs,2spaces,wht,     3661
     off]1.haelfte[blu,rvs,27spaces,cyn,
     off,shift -]";
1330 print "[shift -,blu,rvs,38spaces,cyn,    2666
     off,shift -]";
1340 print "[shift -,blu,rvs,38spaces,cyn,    2666
     off,shift -]";
1350 print "[shift -,blu,rvs,38spaces,cyn,    2666
     off,shift -]";
1360 print "[shift -,blu,rvs,38spaces,cyn,    2666
     off,shift -]";
1370 print "[shift -,blu,rvs,35spaces,wht,    3081
     off]f1[blu,rvs] [cyn,off,shift -]";
1380 print "[shift -,blu,rvs,38spaces,cyn,    2666
     off,shift -]";
1390 print "[cbm q,38shift *,cbm w]";         3269
1400 print "[shift -,cbm 7,rvs,38spaces,      2630
     cyn,off,shift -]";
1410 print "[shift -,cbm 7,rvs,35spaces,      2989
     wht,off]f3[cbm 7,rvs] [cyn,off,
     shift -]";
1420 print "[shift -,cbm 7,rvs,38spaces,      2630
     cyn,off,shift -]";
1430 print "[shift -,cbm 7,rvs,38spaces,      2630
     cyn,off,shift -]";
1440 print "[shift -,cbm 7,rvs,38spaces,      2630
     cyn,off,shift -]";
1450 print "[shift -,cbm 7,rvs,38spaces,      2630
     cyn,off,shift -]";
1460 print "[shift -,cbm 7,rvs,2spaces,wht,   3609
     off]2.haelfte[cbm 7,rvs,27spaces,cyn,
     off,shift -]";
1470 print "[shift -,cbm 7,rvs,38spaces,      2630
     cyn,off,shift -]";
1480 print "[shift -,cbm 7,rvs,38spaces,      2630
     cyn,off,shift -]";
1490 print "[shift -,cbm 7,rvs,38spaces,      2630
     cyn,off,shift -]";
1500 print "[cbm z,38shift *,cbm x]";         3297
1510 get qw$                                  449
1520 if qw$="[f1]" then qw=1:return           1476
1530 if qw$="[f3]" then qw=2:return           1556
1540 goto 1510                                587
```

Hardcopies mit Zerreffekten: „Grafikdoppler".

WWW.HOMECOMPUTERWORLD.COM

Adressen verwalten mit dem C64

System: C64/Floppy/Drucker
Typ: Anwendung
Sprache: Basic

Adreßdaten und Telefonnummern speichert „Sequenz64". Wer einen Drucker hat, bringt die Bytes zudem bequem auf's Papier.

In einer sequentiellen Datei speichert das Programm Namen, Vornamen, Anrede/Titel, Straße mit Hausnummer, Wohnort, Postleitzahl und Telefonnummer auf Diskette. Auf Wunsch druckt „Sequenz" die Daten auf einem Epson-kompatiblen Drucker. Nach dem Start befindet man sich im Hauptmenü. Es stehen zur Auswahl:

1.) Name und Adresse eingeben
2.) Laden/Löschen/Suchen
3.) Floppy Befehle
4.) Ende

In Unterprogramm eins gibt man nacheinander die gewünschten Adressen mitsamt Telefonnummern ein. Pro Eingabe sind maximal 22 Zeichen zulässig. Ansonsten meldet der Computer einen Fehler und fordert auf, die Daten gekürzt einzugeben. Am Ende der Eingabe muß man sein O.K. geben, damit der neue Datensatz in die Datei aufgenommen wird. Unter Programmpunkt zwei lädt man bereits gespeicherte Dateien. Mit den Cursor up/- und down-Tasten fährt man durch eingegebene Adressen, mit „Links-Pfeil" gelangt man ins Hauptmenü zurück. „D" druckt Adressen. „Sequenz" fragt, ob die aktuell angezeigte oder alle Adressen auf Papier gebracht werden und ob Telefonnummern zu berücksichtigen sind. Nach Eingabe von „S" für Suchen und den Vor-/Nachnamen gibt das Programm Adresse und Telefonnummer aus, falls ein entsprechender Datensatz gefunden wurde. Ansonsten erscheint „Adresse nicht gefunden". „L" löscht den danach einzugebenden Namen/Datensatz. In Programmteil drei gibt es vier Funktionen:

1.) Directory: Disketteninhaltsverzeichnis ausgeben
2.) Disk-Status: Gibt den Fehlerkanal der Floppy aus.
3.) Disk-Befehl: Sendet Befehle wie S=Scratch oder N=New an die Floppy.
4.) Copy-Datei: Kopiert Adreßdaten von einer Diskette auf eine formatierte, andere Diskette.

„Sequenz" und sein Aufbau:

100 Copyright
200 Anfang/Hauptmenü
440 Adresseneingabe
755 Anhängen
870 Laden
1150 Floppy Menü
1330 Disk-Status
1370 Directory
1480 Disk-Befehl
1620 Neue Datei eröffnen
1760 Bearbeitungsmenü
1940 Blättern
2050 Drucken
2400 Suchen
2630 Datei kopieren
2870 Programmende
3120 Unterprogramme
3130 Wait
3170 Status
3270 Wait+Taste
3310 Wait
3340 Bildaufbau
3490 Zeile zu lang

M. Kohls

```
100 rem ******************
110 rem *   sequenz 64     *
120 rem *------------------*
130 rem *  (c) copyright   *
140 rem *   malte kohls    *
150 rem *  finkenweg 54    *
160 rem * 2070 ahrensburg *
170 rem * te.:04102/59369 *
180 rem ******************
190 dim a$(2000)                                    591
200 rem ********** anfang
210 u1=0:u2=1:u3=2:u4=3:u5=4:u6=5:u7=6:u8=          5016
    7
220 poke 53280,12:poke 53281,3:poke 646,0           2632
230 rem ********** hauptmenue
240 print "[clr,rvs,cbm 5,14spaces,blk]             2823
    sequenz 64[cbm 5,16spaces]"
250 print "[up,rvs,cbm 5,40spaces]"                 1915
260 print "[up,rvs,cbm 5,11spaces,blk]              2830
    ***** menue *****[cbm 5,12spaces]"
270 print "[up,rvs,cbm 5,40spaces]"                 1915
280 print "[blk,3down,5right,30cbm p]"              6793
290 print "[4right]N[28spaces]N[cbm h]"             2648
300 print "[4right]O[28cbm y]P [cbm h]"             6931
310 print "[4right,cbm h] 1) name &                 3916
     adresse eingeben [cbm n] [cbm h]"
320 print "[4right,cbm h,28spaces,cbm n]            2364
    [cbm h]"
330 print "[4right,cbm h] 2) adressdatei            3184
    laden[7spaces,cbm n] [cbm h]"
340 print "[4right,cbm h,28spaces,cbm n]            2364
    [cbm h]"
350 print "[4right,cbm h] 3) floppy menue           3397
    [12spaces,cbm n] [cbm h]"
360 print "[4right,cbm h,28spaces,cbm n]            2364
    [cbm h]"
370 print "[4right,cbm h] 4) programmende           3541
    [12spaces,cbm n] [cbm h]"
380 print "[4right]L[28cbm p,shift @]N"             6774
390 get a$:if a$="" then 390                        1532
400 if a$="1" then 450                              1331
410 if a$="2" then 880                              1373
420 if a$="4" then 2880                             1592
430 if a$="3" then 1160                             1537
440 rem ********** adresseneingabe
450 clr                                              156
460 print "[clr,rvs,cbm 5,14spaces,blk]             2823
    sequenz 64[cbm 5,16spaces]"
470 print "[up,rvs,cbm 5,40spaces]"                 1915
480 print "[up,rvs,cbm 5,10spaces,blk]              2963
    ***** eingabe *****[cbm 5,11spaces]"
490 print "[up,rvs,cbm 5,40spaces,blk]"             2001
```


WWW.HOMECOMPUTERWORLD.COM

```
500 input "[right]vorname ";vn$                          2026
510 gosub 1080                                            627
520 input "[home,6down,right]nachname";nn$               2184
530 gosub 1080                                            627
540 input "[home,7down,right]anrede/titel
    ";at$                                                2237
545 if len (vn$+nn$+at$)>22 then goto 3490               2673
550 gosub 1080                                            627
560 input "[home,8down,right]strassenname
    ";st$                                                3106
570 gosub 1080                                            627
580 input "[home,9down,right]hausnummer ";
    hn$                                                  2730
585 if len (st$+hn$)>22 then goto 3490                   2496
590 gosub 1080                                            627
600 input "[home,10down,right]wohnort ";
    wo$                                                  2673
610 gosub 1080                                            627
620 input "[home,11down,right]postleitzahl
    ";po$                                                2319
625 if len (wo$+po$)>22 then goto 3490                   2708
630 gosub 1080                                            627
640 input "[home,12down,right]telefon nr."
    ;tn$                                                 2918
645 if len (tn$)>22 then goto 3490                       1917
650 gosub 1080                                            627
660 print "[home,4down]"                                  569
670 for x=1 to 5:gosub 1600                              1702
680 next x                                                306
690 print "[home,5down]"                                  722
700 print "[3right]alles korrekt
     eingegeben (j/n)"                                   4012
710 get ak$:if ak$="" then 710                           1632
720 if ak$="n" then 190                                  1244
730 input "[right,rvs]n[off]eue datei
     eroeffnen, oder zu [rvs]a[off]lter
    [5spaces]datei ergaenzen ";xx$                       6337
740 if xx$="n" then 1630                                 1357
750 gosub 3140                                            603
755 rem ********** anhaengen
760 open 5,8,5,"adressen,s,a"                            2047
770 print#5,at$                                           645
780 print#5,nn$                                           908
790 print#5,vn$                                           972
800 print#5,st$                                           789
810 print#5,hn$                                           860
820 print#5,po$                                           940
830 print#5,wo$                                           996
840 print#5,tn$                                           956
850 close 5                                               266
860 goto 190                                              592
870 rem ********** laden
880 print "[clr,rvs,cbm 5,14spaces,blk]
    sequenz 64[cbm 5,16spaces]"                          2823
890 print "[up,rvs,cbm 5,40spaces]"                       1915
900 print "[up,rvs,cbm 5,11spaces,blk]
    ***** laden *****[cbm 5,12spaces]"                    2686
910 print "[up,rvs,cbm 5,40spaces,blk]"                   2001
920 q=0                                                   374
930 print :gosub 3140:print                               823
940 open 5,8,5,"adressen"                                1651
950 gosub 3180                                            635
960 if s1$<>"00" then close 5:goto 200                   1507
970 close 5                                               266
980 q=0:open 5,8,5,"adressen,s,r"                        2616
990 input#5,a$(q)                                         727
1000 q=q+1                                                355
1010 if st<>64 then 990                                  1355
1020 close 5                                              266
1030 gosub 3420                                           599
1040 if u1=q-8 then 1770                                 1525
1050 gosub 3350                                           619
1060 u1=u1+8:u2=u2+8:u3=u3+8:u4=u4+8:u5=u5+
     8:u6=u6+8:u7=u7+8:u8=u8+8                           6091
1070 goto 1030                                            583
1080 print "[home,16down,3right]
     UCCCCCCCCCCCCCCCCCCCCCCCCCCCCI"                      5592
1090 print "[3right]B[right]";at$;" ";nn$;                3158
     ",";vn$;tab(29)"B"
1100 print "[3right]B[right]";st$;" ";hn$;                2586
     tab(29)"B"
1110 print "[3right]B[right]";po$;" ";wo$;                2594
     tab(29)"B"
1120 print "[3right]B[right]";tn$;tab(                    1989
     29)"B"
1130 print "[3right]                                      4430
     JCCCCCCCCCCCCCCCCCCCCCCCCCCCCK"
1140 return                                               142
1150 rem ********** floppy menue
1160 print "[clr,rvs,cbm 5,14spaces,blk]                  2823
     sequenz 64[cbm 5,16spaces]"
1170 print "[up,rvs,cbm 5,40spaces]"                       1915
1180 print "[up,rvs,cbm 5,7spaces,blk]*****               3434
      floppy menue *****[cbm 5,9spaces]"
1190 print "[up,rvs,cbm 5,40spaces,blk]"                   2001
1200 print "[10right,down,cbm a]                          3856
     CCCCCCCCCCCCCCCCCC[cbm s]"
1210 print "[10right]B 1) directory                       3203
     [3spaces]B"
1220 print "[10right]B 2) disk-status B"                   3581
1230 print "[10right]B 3) disk-befehl B"                   3564
1240 print "[10right]B 4) copy-datei                      3132
     [2spaces]B"
1250 print "[10right]B ←) hauptmenue                      3235
     [2spaces]B"
1260 print "[10right,cbm z]CCCCCCCCCCCCCCCCC              4166
     [cbm x]"
1270 get ee$:if ee$="" then 1270                          1665
1280 if ee$="1" then gosub 1380:goto 1160                 2045
1290 if ee$="2" then 1340                                 1232
1300 if ee$="3" then 1490                                 1340
1310 if ee$="4" then 2640                                 1119
1320 if ee$="←" then 200                                  1262
1330 rem ********** disk-status
1340 gosub 3180                                           635
1350 gosub 3280                                           639
1360 goto 1160                                            611
1370 rem ********** directory
1380 print "[clr]"                                        316
1390 gosub 3180                                           635
1400 if s1$<>"00" then 1160                              1447
1410 open 1,8,0,"$":get #1,a$,a$                         1624
1420 get #1,a$,a$:if st=64 then close 1:                 2436
     goto 1460
1430 get #1,a$,b$:print asc (a$+chr$ (0))+               3713
     256*asc (b$+chr$ (0));
1440 get #1,a$:print a$;:if a$<>"" then                  2686
     1440
1450 print :goto 1420                                     593
1460 gosub 3280                                           639
1470 return                                               142
1480 rem ********** disk-befehl
1490 print "[clr,rvs,cbm 5,14spaces,blk]                 2823
     sequenz 64[cbm 5,16spaces]"
1500 print "[up,rvs,cbm 5,40spaces]"                       1915
1510 print "[up,rvs,cbm 5,8spaces,blk]*****               2997
      disk-befehl *****[cbm 5,9spaces]"
1520 print "[up,rvs,cbm 5,40spaces,blk]"                   2001
1530 gosub 3180                                           635
1540 if s1$<>"00" then 1160                              1447
1550 poke 631,34:poke 198,1                              1840
1560 input "[2down,2right]>";11$                         1230
1570 open 1,8,15,11$:close 1:print "[down]"              2136
1580 gosub 3280                                           639
1590 goto 1160                                            611
1600 print "[67spaces]";                                 2666
1605 print "[12spaces]"                                    747
1610 return                                               142
1620 rem ********** neue datei
1630 gosub 3140                                           603
1640 open 5,8,5,"@:adressen,s,w"                         2946
1650 print#5,at$                                          645
1660 print#5,nn$                                          908
1670 print#5,vn$                                          972
1680 print#5,st$                                          789
1690 print#5,hn$                                          860
1700 print#5,po$                                          940
1710 print#5,wo$                                          996
1720 print#5,tn$                                          956
1730 close 5                                              266
1740 goto 200                                             558
1750 gosub 3180                                           635
1760 rem ********** bearbeitungsmenue
1770 print "[home,5down]"                                  722
1780 for x=1 to 4                                         852
1790 print "[72spaces]"                                   2652
1800 next x                                               306
1810 print "[home,5down]"                                  722
1820 print "[4right]durchblaettern mit                    5964
      cursor up, bzw.[6spaces]cursor down."
1830 print "[4right]←[2spaces]-- menue"                   2309
1840 print "[up,4right]d[2spaces]--                       2223
      drucken"
1850 print "[4right]s[2spaces]-- suchen"                  2300
```


WWW.HOMECOMPUTERWORLD.COM

```
1860 print "[up,4right]l[2spaces]--          2485
     loeschen"
1870 get z$:if z$="" then 1870               1821
1880 if z$=chr$ (145) then 1950              1598
1890 if z$=chr$ (17) then 2000               1430
1900 if z$="+" then 200                      1566
1910 if z$="d" then 2060                     1620
1920 if z$="s" then 2410                     1578
1930 if z$="l" then 2900                     1615
1940 rem ********** blaettern
1950 u1=u1+8:u2=u2+8:u3=u3+8:u4=u4+8:u5=u5+  6091
     8:u6=u6+8:u7=u7+8:u8=u8+8
1960 if u1>q-8 then 2000                     1476
1970 gosub 3350                               619
1980 gosub 3420                               599
1990 goto 1810                                599
2000 u1=u1-8:u2=u2-8:u3=u3-8:u4=u4-8:u5=u5-  6347
     8:u6=u6-8:u7=u7-8:u8=u8-8
2010 if u1<0 then 1950                       1290
2020 gosub 3350                               619
2030 gosub 3420                               599
2040 goto 1810                                599
2050 rem ********** drucken
2060 print "[home,3down]"                      620
2070 for x=1 to 4                             852
2080 print "[68spaces]"                       2532
2090 next x                                   306
2100 print "[home,5down]"                     722
2110 print "[rvs]a[off]lles oder [rvs]n      4835
     [off]ur (";a$(u1);" ";a$(u2);",";
     a$(u3);")";:input o$
2120 input "telefonnummer(n) mitdrucken       4224
     (j/n)";nj$
2130 f1=u1:f2=u2:f3=u3:f4=u4:f5=u5:f6=u6:     5644
     f7=u7:f8=u8
2140 if o$="n" then 2290                     1679
2150 print "[rvs]bitte drucker einschalten   2614
     [off]"
2160 gosub 3320                               595
2170 close 2:open 2,4,0:cmd 2                1374
2180 u1=0:u2=1:u3=2:u4=3:u5=4:u6=5:u7=6:u8=  5016
     7
2200 print a$(u1);" ";a$(u3);" ";a$(u2)      2943
2210 print a$(u4);" ";a$(u5)                 1860
2220 print a$(u6);" ";a$(u7)                 1926
2230 if nj$="j" then print "te.:";a$(u8)     2163
2240 print "                                  221
2250 u1=u1+8:u2=u2+8:u3=u3+8:u4=u4+8:u5=u5+  6091
     8:u6=u6+8:u7=u7+8:u8=u8+8
2260 if u1=q then print#2:close 2:goto 2280  1952
2270 goto 2200                                569
2280 u1=f1:u2=f2:u3=f3:u4=f4:u5=f5:u6=f6:    5806
     u7=f7:u8=f8:goto 1770
2290 print "[rvs]bitte drucker einschalten   2614
     [off]"
2300 gosub 3320                               595
2310 open 2,4,0:cmd 2                        1194
2330 print a$(u1);" ";a$(u3);" ";a$(u2)      2943
2340 print a$(u4);" ";a$(u5)                 1860
2350 print a$(u6);" ";a$(u7)                 1926
2360 if nj$="j" then print "te.:";a$(u8)     2163
2370 print#2                                  252
2380 close 2                                  260
2390 goto 1770                                643
2400 rem ********** suchen
2410 print "[home,3down]"                      620
2420 for x=1 to 4                             852
2430 print "[68spaces]"                       2532
2440 next x                                   306
2450 print "[home,5down]"                     722
2460 print "[right]bitte geben sie den       5070
     namen der gesuchten adresse ein:"
2470 input "[right,down]vorname";v1$         1706
2480 input "[right]nachname";v2$             1408
2490 for x=1 to q                             760
2500 if v2$=a$(x) and v1$=a$(x+1) then 2540  2638
2510 next x                                   306
2520 print "[rvs]adresse nicht vorhanden     2550
     [off]"
2530 gosub 3320:x=1                           826
2540 gosub 3350                               619
2550 print "[home,16down,3right]             5592
     UCCCCCCCCCCCCCCCCCCCCCCCCCCCI"
2560 print "[3right]B[right]";a$(x-1);" ";    4765
     a$(x);",";a$(x+1);tab(29)"B":
2570 print "[3right]B[right]";a$(x+2);" ";    3556
     a$(x+3);tab(29)"B"
2580 print "[3right]B[right]";a$(x+4);" ";    3437
     a$(x+5);tab(29)"B"
2590 print "[3right]B[right]";a$(x+6);tab(    2797
     29)"B"
2600 print "[3right]                          4430
     JCCCCCCCCCCCCCCCCCCCCCCCCCCCK"
2610 u1=x-1:u2=x:u3=x+1:u4=x+2:u5=x+3:u6=x+  6232
     4:u7=x+5:u8=x+6
2620 goto 1770                                643
2630 rem ********** datei kopieren
2640 print "[clr,rvs,cbm 5,14spaces,blk]     2823
     sequenz 64[cbm 5,16spaces]"
2650 print "[up,rvs,cbm 5,40spaces]"          1915
2660 print "[up,rvs,cbm 5,9spaces,blk]*****   2997
     copy-datei *****[cbm 5,9spaces]"
2670 print "[up,rvs,cbm 5,40spaces,blk]"      2001
2680 rem   *** alles copieren ***
2690 print :print "[rvs]bitte quelldiskette    3457
     einlegen[off,down]"
2700 gosub 3320                               595
2710 open 5,8,5,"adressen"                   1651
2720 gosub 3180                               635
2730 if s1$<>"00" then close 5:goto 200      1507
2740 close 5                                  266
2750 k=0:open 5,8,5,"adressen,s,r"           2610
2760 input#5,a$(k)                            853
2770 k=k+1                                    325
2780 if st<>64 then 2760                     1497
2790 close 5                                  266
2800 print "[rvs]bitte zieldiskette          3218
     einlegen[off]"
2810 gosub 3320                               595
2820 open 5,8,5,"@:adressen,s,w"             2946
2830 for tt=0 to k-1                         1150
2840 print#5,a$(tt)                           619
2850 next tt:close 5:q=0                     1348
2860 goto 1160                                611
2870 rem ********** programmende
2880 print "[clr,4down,3right,rvs]           2120
     tschuesss!":end
2890 rem ********** adresse loeschen
2900 print "[home,5down]"                     722
2910 for x=1 to 3:gosub 1600                 1638
2920 next x                                   306
2930 print "[home,5down]"                     722
2940 print "bitte vor- und zuname der zu      5620
     loeschendenadresse eingeben."
2950 input "[down]vorname ";w1$              1939
2960 input "nachname ";w2$                   1567
2970 for x=1 to q                             760
2980 if w1$=a$(x) and w2$=a$(x-1) then 3020  2787
2990 next x                                   306
3000 print "[rvs]adresse nicht vorhanden     2550
     [off]"
3010 gosub 3320:goto 1770                    1273
3020 gosub 3140                               603
3030 m=0:open 5,8,5,"@:adressen,s,w"         2835
3040 if a$(m+1)=w2$ and a$(m+2)=w1$ then m=  3515
     m+8
3050 if a$(m)="" then close 5:goto 1770      1922
3060 print#5,a$(m)                            746
3070 if m=q-1 then 3100                      1580
3080 m=m+1                                    335
3090 goto 3040                                595
3100 close 5                                  266
3110 goto 1770                                643
3120 rem ********** unterprogramme
3130 rem wait
3140 print "[2right,down,rvs,blk]bitte        2993
     diskette einlegen"
3150 poke 198,0:wait 198,1:poke 198,0        2514
3160 return                                   142
3170 rem status
3180 print "disk status:"                     1372
3190 print "[down,2right,rvs]";               1089
3200 open 15,8,15                            1060
3210 input#15,s1$,s2$,s3$,s4$                1602
3220 print s1$",";s2$;",";s3$",";s4$         3119
3230 print "[off]"                             312
3240 if s1$<>"00" then wait 198,1:poke 198,  1908
     0
3250 close 15                                 470
3260 return                                   142
3270 rem wait
3280 print "[rvs,down]bitte taste druecken"   1751
3290 poke 198,0:wait 198,1:poke 198,0        2514
3300 return                                   142
```


WWW.HOMECOMPUTERWORLD.COM

```
3310 rem wait
3320 poke 198,0:wait 198,1:poke 198,0          2514
3330 return                                     142
3340 rem bildaufbau
3350 print "[home,16down,3right]                5592
     UCCCCCCCCCCCCCCCCCCCCCCCCCCCI"
3360 print "[3right]B[right]";"[23spaces]";      2542
     tab(29)"B"
3370 print "[3right]B[right]";"[23spaces]";      2542
     tab(29)"B"
3380 print "[3right]B[right]";"[23spaces]";      2542
     tab(29)"B"
3390 print "[3right]B[right]";"[23spaces]";      2542
     tab(29)"B"
3400 print "[3right]                            4430
     JCCCCCCCCCCCCCCCCCCCCCCCCCCK"
3410 return                                     142
3420 print "[home,16down,3right]                5592
     UCCCCCCCCCCCCCCCCCCCCCCCCCCI"
3430 print "[3right]B[right]";a$(u1);" ";       4764
     a$(u2);",";a$(u3);tab(29)"B":
3440 print "[3right]B[right]";a$(u4);" ";       3412
     a$(u5);tab(29)"B"
3450 print "[3right]B[right]";a$(u6);" ";       3478
     a$(u7);tab(29)"B"
3460 print "[3right]B[right]";a$(u8);tab(       2497
     29)"B"
3470 print "[3right]                            4430
     JCCCCCCCCCCCCCCCCCCCCCCCCCCK"
3480 return                                     142
3490 print "[right,down,rvs]zeile zu lang":     4132
     poke 198,0:wait 198,1:poke 198,0:goto
     450
```

„Sequenz 64": Die handliche Datenbank

Die „Wende" in Grusinien

System: C64/Floppy/Drucker
Typ: Spiel
Sprache: Basic

Parteien gründen, Reden halten, Flugblätter verteilen und Spenden verheimlichen — im Basic-Spielprogramm „Partei" geht es zu wie im richtigen Leben. Diplomatisches Handeln ist wichtig, um in der Gunst des Volkes zu steigen.

Das Grusinische Reich ist ein kleiner Staat irgendwo auf der Welt. Es regiert der Reichskanzler, der von der stärksten Regierungspartei gestellt wird. Da eine Partei selten über die absolute Mehrheit verfügt, gibt es regelmäßig Koalitionen. Diese richten sich nach der Fraktionierung — es gibt Rechte und Linke. Doch kann es schon einmal vorkommen, daß eine Partei aus ihrer Fraktion ausschert und die andere Fraktion unterstützt. Einen Fraktionszwang gibt es nicht. Da im Grusinischen Reich direkte Demokratie großgeschrieben wird, wird das Parlament jedes Jahr neu gewählt. Dafür ist eine ausgedehnte Wahlkampfzeit sehr nützlich. Ist man im Parlament und hat sogar einen Posten als Minister, so hat man mehr Möglichkeiten, dem Volk zu zeigen, was in einem steckt. Hier wird man aber natürlich auch viel strenger bewertet, als wenn man nicht im Parlament vertreten ist. Im Vorwahlkampf hat man mehr Möglichkeiten zu Aktionen als während der Legislatur. Diese Periode ist zeitlich begrenzt.

Dann tritt man in die heißeste Phase des Wahlkampfes. Hier teilt man dem Wahlkampfleiter seiner Partei mit, wie intensiv der Wahlkampf geführt wird. Hiernach richten sich die Unkosten einer Partei. Die Finanzen der Partei sind ausschlaggebend für deren Beliebtheit — und die wiederum ist richtungsweisend für das Wahlergebnis.

Koalieren ist alles!

Generell läßt sich sagen, daß radikale Gruppierungen nicht so gern gesehen sind wie demokratische. Aber auch als Partei mit antidemokratischen Grundzügen bleibt man von den demokratischen Prozessen nicht ausgeschlossen. Man darf ungehindert kandidieren und wird von sogenannten Protestwählern gewählt. Diesen kommt es darauf an, daß diese Partei ‚ordentlich auf den Putz haut'. Eine Warnung: Zuviel „auf den Putz hauen" ist nicht gut und schmälert

```
Daten der Partei PRF

Spitzenkandidat: VON BEISSTALER
Staatsamt des Kandidaten: REICHSKANZLER
Zahl der Mitglieder: 58155
Vermoegen der Partei: 20404848 Taler
Sitze im Parlament: 154
Stimmen bei der letzten Wahl: 8534716

Prozente bei der letzten Wahl: 30.89
```

So sehen die Informationen der PRF aus.

WWW.HOMECOMPUTERWORLD.COM

die Aussicht, in das Parlament einzuziehen, erheblich.
Ist die Wahl beendet, wird die Zusammensetzung der neuen Regierung bekanntgegeben. Anschließend ist jedermann Möglichkeit gegeben, Dinge wie Zusammensetzung des Reichstages, genaue Wahlergebnisse und anderes zu erfahren. Einiges kann man sich auf Wunsch auch schriftlich mit in die nächste Legislaturperiode geben lassen.
Die Beliebtheit der Parteien wird kurz vor der Wahl in einer Umfrage festgestellt. Die Anzahl der Wähler hängt aber nicht hiervon ab — selbst Parteien mit relativ hoher Beliebtheit können an der 3,5-Prozent-Hürde scheitern.

Bedienung des Programmes

Gestartet wird dieses Basic-Programm für den C64 mit RUN. Es wird nichts nachgeladen, trotzdem dauert es ein paar Sekunden, bis der erste Reichstag in seiner Zusammensetzung feststeht. Aus Speicherplatzgründen können nicht mehr als 13 Parteien zu gleicher Zeit am Geschehen teilhaben. Da der Computer acht Parteien steuert, ist die Anzahl der Mitspieler somit auf fünf begrenzt. Aufgrund der guten Menüsteuerung ist es für Interessierte ein leichtes, sich in diesem Programm zurechtzufinden. Hier ein wichtiger Hinweis: Eine relativ brauchbare INPUT-Routine befindet sich in den Zeilen 63000 bis 63009. Wenn man das GOSUB 62000 in Zeile 63009 wegläßt, so kann man diese Routine ohne Schwierigkeiten in ein eigenes Programm einbauen. Die Länge der hier einzugebenden Variable A$(2) wird durch Zeile 63008 geregelt. Wer noch eine Maschinenroutine zum Einlesen des Directorys ohne Programmverlust benötigt, der sollte die DATAS in den Zeilen 60500 bis 60590 einmal ab 828 laden und mit SYS 828 starten.
Nach der Darstellung der neuen Regierung hat man folgendes Menü:

F1 — Darstellung der Machtverhältnisse mit Balkengrafik.
F2 — Darstellung der Regierung (Nennung der Minister)
F3 — Politische Einstufung der Parteien
F4 — Die finanzielle Situation der Parteien im Überblick
F5 — Parteienstruktur verändern (Neue Partei gründen und ähnliches)
F6 — Informationen über eine Partei auf dem Bildschirm
F7 — In die Legislaturperiode gehen

Weiterhin sind noch folgende Buchstaben mit Befehlen belegt:

B — Die durch eine Umfrage herausgefundene Beliebtheit der einzelnen Parteien in Balkengrafik und mit der Vergebung von Punkten
F — Die Fraktionierung der Parteien (Rechte, Linke und nicht im Reichstag vertretene)
E — Daten einer einzelnen Partei ausdrucken
D — Zusammensetzung des Reichstages ausdrucken
P — Daten aller Parteien ausdrucken
@ — Storage-Modus. Hier kann man den Spielstand auf Diskette abspeichern, von Diskette laden, die DIRECTORY anschauen oder auch Auslesen des Fehlerkanals

Ist eine besondere Tastenbelegung vorgesehen, wird sie auf dem Bildschirm angezeigt. Damit man sich im Programm zurechtfindet, hier die wichtigsten Variablen:

PR(x) = Prozente einer Partei
SI(x) = Sitze im Reichstag
PN$(x) = Name der Partei
SK$(x) = Name des Spitzenkandidaten der Partei (= Spieler)
ES(x) = Politische Einstufung (0=linksextrem, 15=rechtsextrem)
ZA = Zahl der aktiven Parteien
ST(x) = Stimmen bei der letzten Wahl zum Reichsparlament
GE(x) = Vermögensstand der Partei in Talern (=grus. Währung)
MI(x) = Parteimitglieder
OP(x) = Fraktionierungsvariable (verändert sich nur selten)
SA(x) = Stammwähler der Partei
RE(x) = Unterscheidung Regierung/Opposition
AP$(x) = Amt einer Partei im Staate
AE(x) = Beliebtheit der Partei
GE = Zahl der Wahlberechtigten im Reiche
GM = Gesamtzahl der Mitglieder aller Parteien

Mit Hilfe dieser Variablenliste ist Modifizieren des Programmes nach eigenen Wünschen ohne weiteres möglich. Die Druckerroutinen sind auf einem Compumaster CCP 100 (= Quelle Privileg) geschrieben und funktionieren daher auch auf MPS 802 und kompatiblen Druckern. Da die Druck-Routinen sehr allgemein gehalten sind, macht die Anpassung auf andere Drucker keine allzugroßen Schwierigkeiten.
Da in der jeweils letzten Druckzeile nur mit neudefinierten Zeichen gearbeitet wird, ist hier aufzupassen. *Roland Richter*

```
1 rem dieses ist ein programm von
      roland richter aus martfeld
2 rem      anschrift:
             roland richter
3 rem      la-bazoge 342
             d-2811 martfeld
5 poke 53272,21:poke 53281,0:poke 53280,   4028
  0:poke 657,212
10 print "[clr,wht,3down,5spaces]bitte      2887
   warten[2spaces]- ich lese daten"
11 print "[6down] partei - ein spiel von    4212
   roland richter"
15 read za:read ge:read gm                  1321
16 dim pr(13),si(13),pn$(13),sk$(13),       7325
   es(13),be$(15),st(13),ge(13),mi(13),
   op(13)
17 dim wa(13),wa$(5),sa(13),am$(13),        7732
   ap$(13),ae(13),nz$(22),la$(31),el(31),
   re(13)
18 dim po(13),ma(13)                        1229
20 for x=1 to za                             873
30 read pn$(x)                               600
40 read pr(x)                                603
50 read sk$(x)                               594
```


WWW.HOMECOMPUTERWORLD.COM

```
   51 read es(x)                                                  585
   52 read st(x):sa(x)=int ( rnd (1)*(st(x)/                     6698
      2))+(st(x)/13):st(x)=st(x)+sa(x)
   53 read ge(x)                                                  533
   60 read op(x):read wa(x):read ae(x):read                       3701
      mi(x):next
   70 for x=0 to 15:read be$(x):next                              1549
   71 for x=1 to 5:read wa$(x):next                               2024
   72 for x=1 to 13:read am$(x):next                              1714
   73 for x=1 to 22:for u=1 to 8:read nz:                         4677
      nz$(x)=nz$(x)+chr$ (nz):next :next
   74 for x=828 to 999:read y:poke x,y:next                       2369
   75 for x=1 to 31:read la$(x):read el(x):                       2258
      next
   99 print "[cbm 7]":gosub 50000                                 1121
  100 print "[clr,cbm 7]reichsparlament"                          2015
  110 print "[2down]f1) darstellung der                          3921
      machtverhaeltnisse"
  111 print "f2) darstellung der regierung"                      2987
  120 print "f3) einstufung der parteien"                        2934
  121 print "f4) finanzielle situation"                          2664
  130 print "f5) parteienstruktur                                3210
      veraendern"
  131 print "f6) informationen ueber eine                        3594
      partei"
  140 print "f7) in die legislaturperiode                        3238
      gehen"
  141 print "[down]b ) beliebtheitsskala                         2829
      darstellen"
  142 print "f ) fraktionierung im                               2992
      reichstag"
  143 print "e ) daten einer partei                             3001
      ausdrucken"
  144 print "d ) reichstagszusammensetzung                      4738
      ausdrucken"
  145 print "[up]p ) daten aller parteien                       3218
      ausdrucken"
  146 print "@ ) speichern/laden"                               1936
  200 get a$:if a$="" then 200                                  1380
  201 if a$="[f1]" then goto 51000                              1696
  202 if a$="[f3]" then goto 52000                              1732
  203 if a$="[f5]" then gosub 53000:goto 100                    2447
  204 if a$="[f7]" then goto 23000                              1539
  205 if a$="d" then goto 55000                                 1672
  206 if a$="[f4]" then gosub 57000:goto 100                    2304
  207 if a$="[f6]" then gosub 59000:goto 100                    2344
  208 if a$="f" then gosub 39000:goto 100                       2427
  209 if a$="[f2]" then gosub 37000:goto 100                    2268
  210 if a$="p" then gosub 47000                                1556
  211 if a$="e" then gosub 47800:goto 100                       2453
  212 if a$="@" then gosub 46000:goto 100                       2225
  213 if a$="b" then gosub 45000:goto 100                       2285
  299 goto 200                                                   558
10000 print "[clr,cbm 7]wahlkampf"                             1280
10010 print "[2down]f1) hetzkampagne                           3075
      veranstalten"
10011 print "f2) eine rede halten"                            2427
10020 print "f3) flugblattaktion starten"                     2941
10021 print "f4) untersuchungsausschuss                       4075
      einsetzen"
10030 print "f5) finanzielle situation"                      2672
10031 print "q ) wahlkampf der parteien                      3433
      abbrechen[3down]"
10100 get a$:if a$="" then 10200                              1472
10110 if a$="[f1]" then goto 40000                            1690
10111 if a$="[f3]" then goto 43000                            1734
10112 if a$="[f5]" then gosub 57000:goto                      2595
      10000
10113 if a$="[f4]" then gosub 20000:goto                      2402
      10000
10114 if a$="[f2]" then gosub 49000:goto                      2410
      10000
10115 if a$="q" then ti$=z$                                   1805
10199 goto 10100                                               573
10200 for x=1 to za:if es(x)<0 then es(x)=0                   2564
10201 if es(x)>15 then ex(x)=15                               1655
10202 if ti$>=z$ then goto 48000                              1266
10210 next :print "[up]dauer des                              3326
      wahlkampfes: ";ti$:goto 10100
20000 print "[clr]einen                                       4506
      untersuchungsausschuss einsetzen"
20010 for x=1 to za:print x;"[left]"[red] ";                  2826
      pn$(x);"[cbm 7]":next
20020 print "gib die nummer deiner partei                     5148
      ein: ";:gosub 63000:a$=a$(2)
20030 if val (a$)>0 and val (a$)<za+1 then                    2365
      goto 20050
20040 print "[rvs]falsche eingabe[off]":goto                 2982
      20020
20050 if op( val (a$))=0 then goto 20900                      1756
20051 print "[cbm 6]bist du ";sk$( val                       3474
      (a$));"? (j/n)[cbm 7]"
20060 get b$:if b$="" then 20060                              1579
20070 if b$="n" then print "[down]warum                       4323
      gibst du dich als ";sk$( val (a$))
20080 if b$="n" then print "aus? taste                        6567
      druecken!":poke 198,0:wait 198,1:poke
      198,0:goto 10000
20090 print "[clr,cbm 7]weshalb soll ein                     6188
      untersuchungsausschuss eingesetzt
      werden ?"
20100 print "[down,wht]1) wegen einer                        2959
      anderen partei"
20110 print "2) wegen des reichskanzlers"                    2701
20120 print "3) wegen der gewerkschaften"                    3258
20130 print "4) wegen der industrie"                         2573
20140 print "5) wegen der spionage"                          2459
20150 print "6) wegen der                                    3303
      umweltverschutzung"
20160 print "7) wegen der armee"                             2040
20170 print "8) wegen des verkehrsproblems"                  3233
20180 print "[down,red]gib die nummer ein ";                 2549
20190 get a$(2):if a$(2)="" then 20190                        2487
20200 if val (a$(2))<1 or val (a$(2))>za                      2964
      then goto 20190
20210 if a$(2)="1" then 20300                                 1300
20220 if a$(2)="8" then 21500                                 1387
20221 if a$(2)="2" then 21800                                 1222
20222 if a$(2)="3" then 21600                                 1414
20223 if a$(2)="6" then 21900                                 1258
20224 if a$(2)="7" then 22000                                 1242
20225 if a$(2)="4" then 22100                                 1271
20226 if a$(2)="5" then 22200                                 1304
20300 print "[clr,2down,cbm 7]                               6827
      untersuchungsausschuss wegen einer
      [6spaces]anderen partei einsetzen"
20301 print "[clr,blu]wegen welcher partei                    6051
      soll der ausschuss untersuchen?[cyn]"
20302 for x=1 to za                                            873
20303 if x=val (a$) then print x"[left])                      3300
      keine option":goto 20305
20304 print x"[left]) ";pn$(x)                                1759
20305 next x:print "[red]nummer der partei:                   2909
      ";
20306 gosub 63000:a$(1)=a$(2)                                 1522
20307 if a$(1)=a$ then print "[down,cbm 7]                    4879
      geht nicht":poke 198,0:wait 198,1:poke
      198,0:return
20308 if val (a$(1))<0 or val (a$(1))>za                      4094
      then print "falsche taste":goto 20306
20309 print "[cbm 7]wegen was soll gegen                     6023
      diese partei er-[3spaces]mittelt
      werden?[down,red]"
20310 print "[down]1) steuerhinterziehung"                   2366
20311 print "2) politischem radikalismus"                    2684
20312 print "3) geheimnisverrat"                             2243
20313 print "[down,cbm 7]gib die                             3418
      entsprechende zahl ein."
20314 get a$:if a$="" then 20314                              1578
20315 if val (a$)<0 or val (a$)>3 then print                  3534
      "geht nicht":goto 20314
20316 if a$="1" then goto 20320                               1604
20317 if a$="2" then goto 21100                               1592
20318 if a$="3" then a$=a$(1):goto 21000                      2124
20320 print "[clr,wht]ein parlamentarischer                  4616
      untersuchungsaus-"
20321 print "schuss tagt wegen                               6049
      steuerhinterziehung[3spaces]der
      partei ";pn$( val (a$(1)))
20322 i=int ( rnd (1)*8)+1                                    1421
20323 print "[down,cbm 7]das ergebnis:":sa(                  5754
      val (a$(1)))=sa( val (a$(1)))-(sa( val
      (a$(1)))/10)
20324 if i=1 then print "der                                  5263
      untersuchungsausschuss ist nicht in"
20325 if i=1 then print "der lage, ein                        6409
      ordentliches ergebnis ab- zuliefern."
      :goto 20352
```


WWW.HOMECOMPUTERWORLD.COM

```
20326 if i=2 then print "es liegt keine steuerhinterziehung vor.":goto 63100  5169
20327 if i=3 then print "was wie steuerhinterziehung aussah, war eine spende.":goto 20352  6916
20328 if i=4 then print "der ausschussvorsitzende hatte keine [4spaces]lust mehr."  5993
20329 if i=4 then print "deshalb liegt auch kein ergebnis vor":goto 63100  4952
20330 if i=5 then print "es wird gemunkelt, dass die mitglieder"  4305
20331 if i=5 then print "des ausschusses bestochen worden sind."  4685
20332 if i=5 then print "ob das der grund ist, dass gar kein"  3933
20333 if i=5 then print "ergebnis vorliegt." :goto 63200  3704
20334 if i=6 then print "die vermutung scheint richtig zu sein."  4588
20335 if i=6 then print "der fall wird dem gericht uebergeben."  4166
20336 if i=7 then print "der fall wird dem gericht uebergeben."  4174
20337 if i=8 then print "der ausschuss gab die[2spaces]arbeit auf."  3831
20338 if i=8 then print "durch das kuddelmuddel der finanzen"  4185
20339 if i=8 then print "kann sich doch kein mensch hindurch-"  4796
20340 if i=8 then print "finden.die arbeit wird dem gericht"  3856
20341 if i=8 then print "ueber lassen (ha!ha!ha!ha!)"  3442
20342 print "[down,grn]das gericht sagt:"  1954
20343 i=int ( rnd (1)*4)+1  1413
20345 if i=1 then print "[down,cbm 7]der angeklagte wird freigesprochen.":goto 20352  5315
20346 if i=2 then print "[down,cbm 7]der angeklagte ist nicht schuldig.":goto 63100  5214
20347 if i=3 then print "[down,cbm 7]der angeklagte wird fuer schuldig befunden":goto 63200  5809
20348 if i=4 then print "[down,cbm 7]die kassenbuecher waren zu schlampig [4spaces]gefuehrt!"  5228
20349 print "deshalb wurde das verfahren mangels an"  3199
20350 print "beweisen eingestellt.eine strafe muss[3spaces]aber gezahlt werden: 50000 t."  7334
20351 ge( val (a$(1)))=ge( val (a$(1)))-50000  2553
20352 poke 198,0:wait 198,1:poke 198,0: return  2886
20353 print "[down,cbm 7]der angeklagte muss eine strafe zahlen:":i=int ( rnd (1)*1000000)+100000  6814
20354 print i;"taler":goto 20352  1749
20900 print "[down,cbm 7]eine nicht im parlament vertretene"  3905
20910 print "partei kann keinen untersuchungsaus-[4spaces]schuss einsetzen."  5581
20920 print "[down,rvs,yel]taste druecken [cbm 7,off]"  2477
20930 poke 198,0:wait 198,1:poke 198,0: return  2886
21000 print "[clr]ein untersuchungsausschuss tagt"  3806
21001 print "wegen geheimnisverrates durch"  2907
21002 print "die partei ";:print pn$( val (a$))  2399
21003 print "[2down,wht]er kommt zu folgendem ergebnis:[cbm 6]"  3925
21004 if re( val (a$))<>2 then print "eine partei, die nicht an der regierung"  4774
21005 if re( val (a$))<>2 then print "beteidigt ist, weiss keine geheimnisse,"  5239
21006 if re( val (a$))<>2 then print "die sie verraten koennte.":goto 20352  4368
21007 i=int ( rnd (1)*10)+1  1612
21008 if i=4 then print "es gibt beweise fuer die schuld dieser[2spaces] partei!"  4574
21009 if i=4 then print "der fall wird dem gericht uebergeben:"  4246
21010 if i=4 then a=int ( rnd (1)*100000000)+100000:ge( val (a$))=ge( val (a$))-a  5158
21011 if i=4 then sa( val (a$))=sa( val (a$))-(sa( val (a$))/10)  3734
21012 if i=4 then print "eine strafe von ";a;" talern":print "ist zu zahlen.": goto 20352  6907
21013 print "die partei wurde fuer unschuldig[8spaces]befunden!":goto 20352  4866
21100 print "[clr,cbm 7]der untersuchungsausschuss tagt wegen"  4088
21110 print "des politischen radikalismus bei der"  3250
21120 print "partei ";pn$( val (a$(1)));": [red]"  3052
21130 if es( val (a$(1)))<3 or es( val (a$(1)))>14 then print "die partei ist zu radikal"  4891
21140 if es( val (a$(1)))<3 or es( val (a$(1)))>14 then print "eine entradikalisierung wird";  5955
21141 if es( val (a$(1)))<3 or es( val (a$(1)))>14 then print " vorgenommen"  4874
21142 if es( val (a$(1)))<3 then es( val (a$(1)))=3  2821
21143 if es( val (a$(1)))>13 then es( val (a$(1)))=13:goto 20352  3659
21144 print "der verdacht hat sich nicht bestaetigt.":goto 20352  4208
21500 print "[clr,cbm 7]einen untersuchungsausschuss tagt wegen des verkehrsproblems:[2down,red]"  7727
21510 i=int ( rnd (1)*3)+1  1411
21520 print "er kommt zu folgendem ergebnis: "  3462
21530 if i=1 then print "es liegt ueberhaupt keine verkehrs-"  4275
21531 if i=1 then print "problem vor.deshalb gibt es auch kein"  4098
21532 if i=1 then print "ergebnis."  1896
21540 if i=2 then print "der ausschuss untersucht, doch er"  3875
21541 if i=2 then print "findet nichts.nach einiger zeit loest"  4748
21542 if i=2 then print "er sich auf.ein ergebnis liegt nicht"  4353
21543 if i=2 then print "vor."  1428
21550 if i=3 then print "der ausschuss tagte, und nach einiger"  4486
21551 if i=3 then print "zeit fand man heraus, das es viel zu"  3669
21552 if i=3 then print "wenig strassen gibt.der ausschuss gibt"  4746
21553 if i=3 then print "eine empfehlung an den verkehrsminister"  4940
21554 if i=3 then print "ab, neue strassen zu bauen."  3396
21560 print "[cbm 8,down]taste druecken.": goto 20352  2911
21600 print "[clr,cbm 7]einen untersuchungsausschuss wegen der"  5112
21601 print "gewerkschaften einsetzen[down, yel]"  2687
21610 print "folgendes ergebnis liegt vor:"  3547
21620 i=int ( rnd (1)*10)+1  1612
21630 if i=1 then print "die gewerkschaften werden von ueblen"  4050
21631 if i=1 then print "maechten missbraucht.deshalb brauchen"  4451
21632 if i=1 then print "sie mehr macht.": es( val (a$))=es( val (a$))-1  4484
21640 if i=2 then print "die gewerkschaften  4001
```


WWW.HOMECOMPUTERWORLD.COM

```
        haben viel zu viel"
21641 if i=2 then print "macht.eine          4367
        beschneidung dieser mahcht"
21642 if i=2 then print "ist im interesse    4501
        der aufrachterhaltung"
21643 if i=2 then print "der demokratie in   4375
        unserem staate un-"
21644 if i=2 then print "bedingt notwendig." 4838
        :es( val (a$))=es( val (a$))+1
21650 if i=3 then print "die gewerkschaften  4570
        haben zuwenig geld."
21651 if i=3 then print "eine finanzspritze  4546
        ist im interesse der"
21652 if i=3 then print "arbeitnehmer        3512
        notwendig."
21660 if i=4 then print "die gewerkschaften  3936
        haben viel zuviel"
21661 if i=4 then print "geld.damit sie      4284
        unsere wirtschaft duurch"
21662 if i=4 then print "ihre vielen         4231
        fehlinvestitionen nicht"
21663 if i=4 then print "ruinieren, muessen  3998
        sie mit einer "
21664 if i=4 then print "sondersteuer belegt 3624
        werden."
21670 if i=5 then print "die aktivitaeten    4896
        der gewerkschaften auf"
21671 if i=5 then print "dem bausektor       4496
        muessen unterstuezt"
21672 if i=5 then print "werden, damit       4461
        moeglichst viele neue"
21673 if i=5 then print "schoene wohnungen   4176
        entstehen."
21680 if i=6 then print "die gewerkschaften  3669
        machen auf dem"
21681 if i=6 then print "bausektor nur noch  4730
        murks.dafuer muss"
21682 if i=6 then print "man ihnen einen     4540
        denkzettel verpassen."
21683 if i=6 then print "deshalb raten wir   4658
        von jeder steuerlichen"
21684 if i=6 then print "[up]unterstuetzung  3305
        ab."
21690 if i=7 then print "ein                 5114
        gewerkschaftseigenes wohnungsbau-"
21691 if i=7 then print "unternehmen droht   4300
        pleite zu gehen."
21692 if i=7 then print "wir raten der       4520
        regierung, dieses durch"
21693 if i=7 then print "gezielte            4911
        subventionen zu verhindern, da"
21694 if i=7 then print "sonst die gefahr    3875
        besteht, dass alle"
21695 if i=7 then print "mieter obdachlos    3473
        werden."
21700 if i=8 then print "wir haben nichts an 4429
        den praktiken der"
21701 if i=8 then print "gewerkschaften      4244
        auszusetzen."
21710 if i=9 then print "wir halten den plan 4907
        der gewerkschaften,"
21711 if i=9 then print "die                 4407
        gesamtarbeitszeit zu kuerzen, fuer"
21712 if i=9 then print "sehr bedenklich."   2404
21720 if i=10 then print "wir halten die     4333
        forderung der gewerk-"
21721 if i=10 then print "schaften nach      4515
        kuerzerer arbeitszeit bei"
21722 if i=10 then print "vollem             4224
        lohnausgleich fuer sehr sinn-"
21723 if i=10 then print "voll, um neue      4419
        arbeitsplaetze zu schaffen."
21730 print "[down,cbm 7]taste druecken":    2896
      goto 20352
21800 print "[clr,yel]einsetzung eines       4322
        untersuchungsaus-"
21801 print "schusses wegen des              4125
        reichskanzlers[2down,cbm 7]"
21810 for x=1 to za:if ap$(x)=               3900
      "reichskanzler" then goto 21820
21811 next                                   130
21820 print "untersuchung gegen den          4772
        reichskanzler[4spaces]";sk$(x)
21830 if es(x)>12 then es(x)=13:             1840
21831 if es(x)>12 then print "[down,red]der  6113
        reichskanzler ist zu
        rechtsgerichter.":goto 20352
21840 if es(x)<3 then es(x)=3                1920
21841 if es(x)<3 then print "[down,red]der   6636
        reichskanzler ist zu linksgerichtet."
      :goto 20352
21850 if es(x)>7 then print "[down,red]es    6195
        liegt nichts gegen den kanzler vor.":
      goto 20352
21860 i=int ( rnd (1)*10)+1                  1612
21861 if i=5 then print "[down,red]dieser    6681
        reichskanzler ist nicht
        besondersvertrauenswuerdig."
21862 if i=5 then goto 20352                 1348
21865 print "[down,red]gegen diesen          5932
        reichskanzler ist nichts[3spaces]
      einzuwenden.":goto 20352
21900 print "[cbm 6,clr]ein                  6906
        untersuchungsausschuss tagt wegen
      [3spaces]der umweltverschmutzung[down,
      cbm 7]"
21910 i=int ( rnd (1)*10)+1                  1612
21911 print "hier ist das ergebnis der       3973
        untersuchung:"
21920 if i=1 then print "es gibt gar keine   4470
        umweltverschmutzung"
21921 if i=1 then print "in unserem schoenen 3888
        lande..."
21922 if i=2 then print "tempo 100 muss auf  5648
        den autobahnen ein-[2spaces]gefuehrt
        werden."
21923 if i=3 then print "alle autos muessen  6856
        mit einem katalysatorausgeruestet
        werden."
21924 if i=4 then print "die lage des waldes 3992
        ist bedrohlich"
21925 if i=5 then print "fast der ganze wald 4642
        ist abgestorben"
21926 if i=6 then print "wir duerfen keine   5190
        autobahnen mehr bauen."
21927 if i=7 then print "die atomkraftwerke  6370
        muessen bald abge-[3spaces]schaltet
        werden."
21928 if i=8 then print "alle                 6502
        kohlekraftwerke muessen sofort
      [5spaces]filter bekommen."
21929 if i=9 then print "alternative          6388
        energien muessen gefordert[2spaces]
      werden."
21930 if i=10 then print "durch gezielte      7235
        massnahmen kann der um-[2spaces]welt
        geholfen werden."
21940 print "[down,cbm 8]taste druecken       2875
      [cbm 7]":goto 20352
22000 print "[clr,blu]einen                  6620
        untersuchungsausschuss wegen der
      [2spaces]armee einsetzen[down]"
22010 print "[down,cbm 7]folgendes ergebnis  3173
        liegt vor:[yel,down]"
22020 i=int ( rnd (1)*10)+1                  1612
22030 if i=1 then print "alles ist in        4477
        ordnung mit der armee":goto 20352
22040 if i=2 then print "der                 6895
        verteidigungsminister hat die armee
        verkommen lassen":lk=0
22041 if i=2 then for x=1 to za:if op(x)=2   6234
      and ap$(x)="verteidigungsminister"
      then lk=x:next
22042 if i=2 then st(lk)=st(lk)-(st(lk)/10): 3722
      goto 20352
22043 i=1:goto 22030                         1112
22100 print "[clr]einen                      6900
        untersuchungsausschhuss wegen der
        industrie einsetzen."
22110 print "[down,wht]er kommt zu folgendem 3728
        ergebnis:[down,red]"
22120 i=int ( rnd (1)*3)+1                   1411
22130 if i=1 then print "unsere industrie    3777
        ist gesund."
22131 if i=1 then es( val (a$))=es( val      2872
      (a$))-1
```


WWW.HOMECOMPUTERWORLD.COM

```
22140 if i=2 then print "unsere industrie      6510
      kraenkelt.wir muessen[2spaces]alle
      steuern erlassen."
22141 if i=2 then es( val (a$))=es( val       2879
      (a$))+1
22150 if i=3 then print "unsere industrie     5381
      steht kurz vor dem[5spaces]bankrott."
22151 if i=3 then es( val (a$))=es( val       2887
      (a$))+1
22160 if i=3 then print "wir muessen mit      5164
      subventionen helfen."
22170 goto 20352                               735
22200 print "[clr]einen                       6395
      untersuchungsausschuss wegen der
      [2spaces]spionage einsetzen.[down]"
22210 print "[wht]und hier ist das ergebnis:  2834
      [down,red]"
22220 i=int ( rnd (1)*5)+1                    1415
22222 if i=1 then print "bei uns haben        3881
      spione es schwer..."
22223 if i=2 then print "spione haben es      3730
      leicht bei uns..."
22224 if i=3 then print "bei uns laeuft       5043
      nichts auf dem spionage- sektor"
22225 if i=4 then print "wir muessen unsere   6685
      geheimdinstaktivi-[3spaces]taeten
      steigern."
22226 if i=5 then print "wir brauchen mehr     5019
      qualifizierte spione."
22227 print "[down,grn]taste druecken":goto   2924
      20352
23000 if za=8 then goto 10000                 1563
23010 for az=9 to za:ae=0:q(1)=1              2124
23020 gosub 62000:if ae=5 then goto 23900     2309
23021 if q(1)=0 then goto 48000               1656
23022 print "[clr,cbm 7]die                   2264
      legislaturperiode[down]"
23030 print "[cbm 4]partei: ";pn$(az)         1758
23040 print "staatsamt: ";ap$(az)             2179
23050 print "sitze im parlament:";si(az)      2925
23060 print "[down,cbm 6]"                     527
23070 if si(az)>0 then print "f1 [wht]        4166
      untersuchungsausschuss einsetzen"
23071 if ap$(az)<>"keines" and ap$(az)<>      5180
      "reichskanzler" then gosub 23210
23072 if ap$(az)="reichskanzler" then gosub   3300
      23220
23080 if si(az)>0 then print "[cbm 6]f3       4684
      [wht]misstrauensantrag einbringen"
23090 print "[cbm 6]f5 [wht]eine rede         2002
      halten"
23091 print "[cbm 6]f7 [wht]legislatur        2697
      beschliessen"
23100 get a$:if a$="" then goto 23100         1587
23110 if a$="[f5]" then pz=az:ae=ae+1:gosub   4109
      49070:goto 23020
23111 if si(az)>0 and a$="[f1]" then ae=ae+   2834
      1:goto 23920
23113 if a$="[f3]" and si(az)>0 then gosub    3104
      24000:goto 23020
23114 if a$="[f2]" and ap$(az)<>"keines"      4224
      then ae=ae+1:gosub 23300:goto 23020
23120 if a$="[f7]" then goto 23900            1611
23200 goto 23100                               587
23210 print "[cbm 6]f2 [wht]handlung im ";    3329
      ap$(az);"ium":return
23220 print "[cbm 6]f2 [wht]handlung im       2908
      kanzleramt":return
23300 print "[clr,cbm 7]handlung im ";        1519
23301 if ap$(az)<>"reichskanzler" then a$=    5257
      ap$(az)+"ium":print a$:goto 23303
23302 print "kanzleramt"                       1221
23303 print "[down,cbm 8]was moechten sie      2805
      machen?"
23304 print "[down,cbm 7]a) eine dienstreise  2933
      unternehmen"
23305 print "b) ihren lohn erhohen"           2124
23306 print "c) mehr arbeiten"                1581
23307 print "d) gar nichts tun"               1743
23308 print "[down,yel]geben sie den          4551
      entsprechenden buchstaben ein: ";
23309 get a$:if a$="" then 23309              1692
23310 print a$                                 427
23311 if a$="d" then sa(az)=int (sa(az)*      5362
      0.9):es(az)=es(az)-1:return
23312 if a$="b" then ge(az)=ge(az)+10000:     6865
      sa(az)=int (sa(az)*0.9):es(az)=es(az)+
      1:return
23313 if a$="a" then goto 23320               1618
23314 if a$="c" then ae(az)=ae(az)+1:return   2796
23315 goto 23300                               603
23320 x=int (( rnd (1)*31)+1)                 1805
23321 print "[down,cbm 6]du bereist ";        5391
      la$(x):print "[down]taste druecken":
      ae(az)=ae(az)+el(x)
23322 poke 198,0:wait 198,1:poke 198,0:       2886
      return
23900 next                                     130
23910 goto 63010                               603
23920 rem
23924 a$=str$ (az):gosub 20090:goto 23020     1953
24000 print "[clr,cbm 7]einen                 3398
      misstrauensantrag einbringen"
24010 ae(az)=ae(az)-1                         1610
24020 if re(az)<>2 then ae(az)=ae(az)+2       2361
24030 if ap$(az)="reichskanzler" then         4490
      ae(az)=int (ae(az)/2)
24040 qw=0:for x=1 to za                      1145
24050 if ap$(x)="reichskanzler" then qw=x     2982
24060 next x                                   306
24070 print "[down,cbm 6]";sk$(az);" bringt   3639
      einen miss-"
24080 print "trauensantrag gegen              3424
      reichskanzler"
24090 print sk$(qw);" ein.":qw(1)=0:qw(2)=0   3008
24100 for x=1 to za                            873
24110 if re(x)=1 then qw(1)=qw(1)+si(x)       2750
24120 if x=az then qw(1)=qw(1)+si(x):goto     3235
      24140
24130 if re(x)=2 then qw(2)=qw(2)+si(x)       2643
24140 next x                                   306
24150 print "[down]und hier ist das ergebnis 5059
      der abstimmungim parlament:"
24160 print qw(1);" stimmen gegen ";sk$(qw)    3005
24170 print qw(2);" stimmen fuer ";sk$(qw)     3026
24180 if qw(1)>qw(2) then goto 24230          1912
24190 print "[down]der reichskanzler bleibt   2983
      im amt."
24200 if ap$(az)<>"reichskanzler" and          6092
      ap$(az)<>"keines" and re(az)=2 then
      re(az)=1
24210 if ap$(az)<>"reichskanzler" and          5571
      ap$(az)<>"keines" then ap$(az)=
      "keines"
24220 ae(az)=int (ae(az)-(ae(az)*.2)):goto    3274
      24999
24230 print "[down]der reichskanzler ist      4481
      gestuerzt worden":q(1)=0
24240 ae(qw)=ae(qw)-int (ae(qw)*.3)           2637
24999 poke 198,0:wait 198,1:poke 198,0:       2886
      return
26000 for x=1 to za                            873
26001 if ge(x)>mi(x)*300 then ge(x)=mi(x)*    3168
      301
26002 if ge(x)<-450000000 then ge(x)=-        6232
      300000000-mi(x):sa(x)=int (sa(x)-
      (sa(x)*.9))
26004 if mi(x)>748503 then mi(x)=int (( rnd   5220
      (1)*748560)+778580)-mi(x)
26005 if ap$(x)="reichskanzler" then ae(x)=   3886
      ae(x)-3
26006 if si(x)>(mi(x)*10) then goto 48000     2169
26007 if ae(x)<1 then ae(x)=1                 1790
26008 if pr(x)<0.0000001 then pr(x)=0.000001  2882
26018 next                                     130
26020 for x=1 to za                            873
26030 st(x)=int (ge/100*pr(x))                2064
26040 mi(x)=int ((st(x)/11)+1+si(x)+234)      3279
26050 next :return                             304
30000 y=ae(x)                                  452
30010 for yy=1 to 25                          1018
30020 y=y-2                                    419
30021 if y=-1 then print "[cbm k]";            1459
30030 if y<0 then goto 30050                  1397
```


WWW.HOMECOMPUTERWORLD.COM

```
30040 print "[rvs] [off]";                                    609
30050 next                                                    130
30060 return                                                  142
30100 for x=1 to 8                                            725
30110 i=int ( rnd (1)*5)+0                                   1399
30120 y=int ( rnd (1)*2)+1                                   1425
30130 if y=1 then ae(x)=ae(x)-i                              1877
30140 if y=2 then ae(x)=ae(x)+i                              1884
30170 i=int ( rnd (1)*20)+1                                  1614
30180 if i=5 then es(x)=es(x)-1                              2089
30190 if i=19 then es(x)=es(x)+1                             2360
30191 next                                                    130
30192 for x=1 to za                                           873
30193 if es(x)<1 then es(x)=1                                1918
30194 if es(x)>49 then es(x)=50                              1487
30195 next :return                                            304
37000 print "[clr,cbm 3]darstellung der                      3407
      regierung[down,cbm 7]"
37010 i=0:for x=1 to za                                      1283
37020 if ap$(x)="reichskanzler" then i=x                     2602
37021 next                                                    130
37090 print ap$(i);": ";sk$(i)                               1941
37100 for x=1 to za                                           873
37110 if x=i then goto 37121                                 1466
37120 if re(x)=2 then print ap$(x);": ";                     2510
      sk$(x)
37121 next                                                    130
37130 print "[down,cbm 5]dieses ist das                      6363
      regierungsabinett":print "[cbm 6]
      taste druecken":goto 20352
38000 print "[clr,cbm 7]hier ist das                         5549
      regierungskabbinett:":i=0:si(0)=0
38010 for x=1 to za:i=i+si(x):re(x)=1:next                   2097
38020 a=0:for x=1 to za                                      1275
38030 if op(x)=2 then a=a+si(x)                              1851
38031 next                                                    130
38040 if a>(i/2) then goto 38060                             1945
38041 if a=(i/2) then si(1)=si(1)+1:goto                     3448
      38010
38050 print "[down,cbm 6]die linksparteien                   4454
      kommen an die macht.":xx=1
38051 for x=1 to za:if op(x)=1 then re(x)=2                  2607
38052 next :goto 38070                                        803
38060 print "[down,cbm 6]die rechtsparteien                  4270
      kommen an die macht.":xx=2
38061 for x=1 to za:if op(x)=2 then re(x)=2                  2480
38062 next                                                    130
38070 print "[down,red]regierungsbildung                     2750
      [down]"
38080 xy=0:yx=0:for x=1 to za                                1510
38090 if si(x)>xy and re(x)=2 then xy=si(x):                 2856
      yx=x
38100 next                                                    130
38110 ap$(yx)="reichskanzler"                                2206
38120 uy=0:for x=1 to za                                     1153
38121 if x=yx then goto 38130                                1834
38122 if re(x)=2 and op(x)=xx then uy=uy+1:                  3912
      ap$(x)=am$(uy)
38123 if re(x)=1 then ap$(x)="keines"                        2486
38130 next                                                    130
38140 for x=1 to za                                           873
38150 if re(x)=2 then print ap$(x);": ";                     2510
      sk$(x)
38160 next                                                    130
38200 goto 20352                                              735
39000 print "[clr]fraktionierung der                        2679
      parteien:[down]"
39010 print "[yel]mitte-rechts fraktion:                    2618
      [cyn]"
39020 a=0:for x=1 to za                                      1275
39030 if op(x)=2 then print pn$(x):poke 211,                 5178
      15:poke 214,peek (214)-1:sys 58640:
      print si(x)
39031 if op(x)=2 then a=a+si(x)                              1851
39040 next :print "[15spaces,5cbm t]":print                  3481
      "insgesamt:[5spaces]";a
39050 print "[down,red]linksfraktion:[cyn]"                  2272
39051 a=0:for x=1 to za                                      1275
39052 if op(x)=1 then print pn$(x):poke 211,                 5305
      15:poke 214,peek (214)-1:sys 58640:
      print si(x)
39053 if op(x)=1 then a=a+si(x)                              1978
39054 next :print "[15spaces,4cbm t]":print                  3438
      "insgesamt:[5spaces]";a
39060 print "[down,red]nicht im parlament:                  2500
      [cyn]"
39061 for x=1 to za                                           873
39062 if op(x)=0 then print pn$(x)                           1431
39080 next x                                                  306
39090 poke 198,0:wait 198,1:poke 198,0:                      2886
      return
40000 print "[clr]hetzkampagne veranstalten                 2896
      [down]"
40010 print "wer?"                                           760
40020 for x=1 to za:print x;"[left])[red] ";                  2826
      pn$(x);"[cbm 7]":next
40030 print "gib die nummer deiner partei                   5148
      ein: ";:gosub 63000:a$=a$(2)
40040 if val (a$)>0 and val (a$)<za+1 then                   2430
      goto 40060
40050 print "[rvs]falsche eingabe[off]":goto                2857
      40030
40060 print "[cbm 6]bist du ";sk$( val                      3474
      (a$));"? (j/n)[cbm 7]"
40070 get b$:if b$="" then 40070                             1659
40080 if b$="n" then print "[down]warum                      4323
      gibst du dich als ";sk$( val (a$))
40090 if b$="n" then print "aus? taste                       6567
      druecken!":poke 198,0:wait 198,1:poke
      198,0:goto 10000
40100 print "[down,yel]gegen welche partei                  5922
      willst du die hetz- kampagne starten?
      [cbm 7] ";
40110 gosub 63000:if val (a$(2))>za then                     2684
      goto 40100
40111 c$=a$(2)                                                710
40120 if val (c$)<0 and val (c$)>za then                     4478
      print "[rvs]fehlerhafte eingabe[off]":
      goto 40100
40130 if c$=a$ then print "[down,rvs]du                      5286
      kannst nicht gegen dich selber hetzen
      [off]"
40140 if c$=a$ then poke 198,0:wait 198,1:                   4062
      poke 198,0:goto 10000
40150 print "[down]bitte gib die staerke der                3581
      hetze ein:"
40160 gosub 63000:sv=val (a$(2))                             1487
40170 sa( val (c$))=(sa( val (c$))-(sa( val                  3418
      (c$))/5))
40180 sa( val (a$))=int ((sa( val (a$))-                     3715
      (sa( val (a$))/10)))
40190 if es( val (a$))>=7 then es( val                       3398
      (a$))=es( val (a$))+1
40200 if es( val (a$))<7 then es( val (a$))=                 3376
      es( val (a$))-1
40201 ge( val (a$))=ge( val (a$))-(1000*sv)                  2556
40210 print "[down]ich bin fertig - taste                   3177
      druecken"
40220 poke 198,0:wait 198,1:poke 198,0:goto                  3162
      10000
43000 rem flugblattaktion starten
43010 print "[clr,wht]eine flugblattaktion                  3043
      starten[cbm 6]"
43020 print "[down]wir haben folgende                       2975
      parteien:[cbm 7]"
43030 for x=1 to za                                           873
43040 print x;"[left])";pn$(x)                               1744
43050 next                                                    130
43060 print "[down,yel]welche partei wird                   4163
      die aktion starten?[cbm 7]"
43070 gosub 63000:ap=val (a$(2)):es(ap)=                     2848
      es(ap)-1
43080 if val (a$(2))<1 or val (a$(2))>za                     5171
      then print "[cbm 3,rvs]falsche eingabe
      [off,cbm 7]":goto 43070
43090 a=val (a$(2))                                           564
43100 print "[down,cbm 3]bist du ";sk$(a);"?                3009
      ";:gosub 63000
43110 if left$ (a$(2),1)="j" then goto 43130                 2361
43120 if left$ (a$(2),1)<>"j" then print "                   5566
      [cbm 7]dann eben nicht - taste
      druecken":goto 40220
43130 rem hier geht's weiter
43140 fp=int (( rnd (1)*12)+1):print "                       6159
```


WWW.HOMECOMPUTERWORLD.COM

```
      [cbm 7]preis pro flugblatt:";fp;"taler
       pro 100."
43150 print "[wht]wieviele hunderter-packen     6706
       sollen ver-[3spaces]teilt
       werden?(max.100) ";
43160 gosub 63000:if val (a$(2))>100 then       2947
       goto 43150
43170 ge(ap)=ge(ap)-( val (a$(2))*fp)           2544
43180 print "[down,cbm 7]o.k. die                5748
       flugblaetter sind verteilt.[4spaces]
      taste druecken"
43190 st(a)=st(a)+int (st(a)/1000):mi(a)=       5356
      mi(a)+int (mi(a)/1000):goto 40220
45000 print "[clr]beliebtheitsskala"             2056
45010 print "[down,cyn]die anzahl der            3885
       stimmen richtet sich"
45020 print "natuerlich nach der beliebtheit     2832
       der"
45030 print "einzelnen parteien. in einer        3325
       umfrage"
45040 print "hat ein                            3612
       meinungsforschungsinstitut"
45050 print "diese festgestellt. die umfrage     3835
       wurde"
45060 print "kurz vor der wahl gemacht. sie      3686
       kann als"
45070 print "richtlinie fuer die politik in      3072
       dieser"
45080 print "legislaturperiode gesehen           3724
       werden."
45090 print "die bewertung geht von einem        4327
       punkt, was"
45100 print "mit sehr unbeliebt                  3926
       gleichzusetzen ist,"
45110 print "bis hin zu 50 punkten, die          3191
       absolute"
45120 print "spitze sind.[yel,down]"             1977
45121 print "[down]taste druecken":gosub         5480
      45160:print "[clr,cbm 3]die
       beliebtheitsskala:[cbm 8,down]"
45130 for x=1 to za                              873
45140 print left$ (pn$(x),2);":";:poke 646,      3203
      (x):gosub 30000
45150 print "[cbm 8]";ae(x):next                 1090
45160 poke 198,0:wait 198,1:poke 198,0:          2886
      return
46000 print "[clr,cyn]storagemodus";chr$        5913
      (13);"[cbm 7]spielstand abspeichern
       und laden.[down]"
46010 print "[wht]f1 [cbm 3]-[cbm 6]            2968
       abspeichern des spieles"
46011 print "[wht]f2 [cbm 3]-[cbm 6]            3129
       directory von disk nr.8"
46020 print "[wht]f3 [cbm 3]-[cbm 6]            2877
       einladen des spieles"
46021 print "[wht]f7 [cbm 3]-[cbm 6]            3053
       auslesen des fehlerkanals"
46022 print "[wht]q[2spaces,cbm 3]-[cbm 6]      3225
       zuruck zum vormenue"
46100 get a$:if a$="" then 46100                 1560
46110 if a$="q" then return                      1127
46120 if a$="[f2]" then goto 46200               1603
46130 if a$="[f1]" then gosub 46300:goto         2564
      46000
46140 if a$="[f3]" then gosub 46590:goto         2756
      46000
46150 if a$="[f7]" then gosub 46190:goto         2533
      46000
46189 goto 46100                                 603
46190 open 1,8,15                                935
46191 input#1,a,b$,c,d                           885
46192 close 1                                    258
46193 print "[clr,cbm 7]fehlernummer: ";a        2293
46194 print "bezeichnung im klartext:":print    2963
      b$
46195 print "track/sector:":print c,d           2093
46196 poke 198,0:wait 198,1:poke 198,0           2514
46199 return                                     142
46200 print "[clr,cbm 7]directory einlesen       3334
      [cyn,down]":sys 828
46210 poke 198,0:wait 198,1:poke 198,0:goto      3102
      46000
46300 print "[clr,cbm 7]abspeichhern des         3362
       spielstandes[cyn,down]"
46310 print "legen sie bitte die diskette,       2985
       auf der"
46320 print "ich den spielstand abspeichern      3884
       soll, in"
46330 print "das laufwerk mit der                3269
       geraetenummer 8"
46340 print "und druecken sie zum beginnen       3571
       des ab-"
46350 print "speichervorganges die taste         3293
      [rvs]+[off].[down]"
46351 print "wenn sie nicht speichern            6358
       moechten, so[3spaces]druecken sie
      [rvs]f[off]."
46360 get a$:if a$="" then 46360                 1753
46361 if a$="f" then return                      1224
46370 if a$<>"←" then 46360                      1642
46400 open 1,8,1,"@:partei daten,s,w"            2715
46410 print#1,za                                 656
46420 for x=1 to za                              873
46430 print#1,pr(x)                              768
46440 print#1,si(x)                              903
46450 print#1,pn$(x)                             1011
46460 print#1,sk$(x)                             987
46470 print#1,es(x)                              696
46480 print#1,st(x)                              824
46490 print#1,ge(x)                              743
46500 print#1,mi(x)                              855
46510 print#1,op(x)                              728
46520 print#1,wa(x)                              807
46530 print#1,sa(x)                              775
46540 print#1,re(x)                              831
46550 print#1,ap$(x)                             668
46560 print#1,ae(x)                              695
46570 next                                       130
46580 close 1:return                             606
46590 print "[clr,cbm 7]einladen des              3018
       spielstandes[cyn,down]"
46591 print "achtung: der momentane               3542
       spielstand geht"
46592 print "verloren, wenn er nicht              3605
       abgespeichert"
46593 print "ist. wenn sie nicht laden,           3031
       sondern"
46594 print "speichern moechten, so druecken     4760
       sie [rvs]+[off]."
46595 print "andernfalls druecken sie [rvs]f     2831
      [off]."
46596 get a$:if a$="" then 46596                 1757
46597 if a$="←" then return                      1320
46598 if a$<>"f" then goto 46596                 1539
46600 open 1,8,0,"partei daten,s,w"              2350
46610 input#1,za                                 636
46620 for x=1 to za                              873
46630 input#1,pr(x)                              748
46640 input#1,si(x)                              883
46650 input#1,pn$(x)                             991
46660 input#1,sk$(x)                             967
46670 input#1,es(x)                              676
46680 input#1,st(x)                              804
46690 input#1,ge(x)                              723
46700 input#1,mi(x)                              835
46710 input#1,op(x)                              708
46720 input#1,wa(x)                              787
46730 input#1,sa(x)                              755
46740 input#1,re(x)                              811
46750 input#1,ap$(x)                             648
46760 input#1,ae(x)                              675
46770 next                                       130
46780 close 1:return                             606
47000 rem ausdrucken der parteien-
            struckturen
47010 open 4,4:us=0                              1282
47030 print#4,chr$ (14);"daten der parteien"     2276
47040 for x=1 to za                              873
47050 print#4                                    256
47060 print#4,"[down]Name der Partei[up]: ";     2364
      pn$(x)
47070 print#4,"[down]Spitzenkandidat[up]: ";     2587
      sk$(x)
47080 print#4,"[down]Staatsamt des               3625
```


WWW.HOMECOMPUTERWORLD.COM

```
      Kandidaten[up]: ";ap$(x)
47090 print#4,"[down]Zahl der Mitglieder[up]  2845
      :";mi(x)
47091 print#4,"[down]Vermoegen der Partei     4120
      [up]:";ge(x);"[down]Taler[up]"
47100 print#4,"[down]Sitze im Parlament[up]:  3260
      ";si(x)
47101 print#4,"[down]Stimmen bei der letzten  3672
       Wahl[up]:";st(x)
47102 print#4,"[down]Prozente bei der         4722
       letzten Wahl[up]:";pr(x);"%"
47103 print#4,"[down]Einschaetzung[up]: ";    2993
      be$(es(x))
47104 if us=1 then goto 47120                 1404
47110 poke 198,0:wait 198,1:poke 198,0:next   2757
47120 print#4:close 4                          560
47130 gosub 47500:return                       856
47350 close 1:close 4:end                      665
47500 open 1,4,5                               954
47510 open 4,4                                 600
47520 a=1                                      362
47530 print#1,nz$(a)                          1048
47540 print#4,chr$ (254);chr$ (141);          1488
47550 print#1,nz$(a+1)                        1439
47560 print#4,"[2spaces]";chr$ (254);chr$     2665
      (141);
47570 print#1,nz$(a+2)                        1441
47580 print#4,"[3spaces]";chr$ (254);chr$     2333
      (141);
47590 print#1,nz$(a+3)                        1443
47600 print#4,"[4spaces]";chr$ (254);chr$     1924
      (141);
47610 print#1,nz$(a+4)                        1445
47620 print#4,"[5spaces]";chr$ (254);chr$     1871
      (141);
47630 print#1,nz$(a+5)                        1447
47640 print#4,"[7spaces]";chr$ (254);chr$     2318
      (141);
47644 print#1,nz$(a+6)                        1449
47645 print#4,"[8spaces]";chr$ (254);chr$     2659
      (141);
47654 print#1,nz$(a+7)                        1451
47655 print#4,"[9spaces]";chr$ (254);chr$     2831
      (141);
47664 print#1,nz$(a+8)                        1453
47665 print#4,"[10spaces]";chr$ (254);chr$    2920
      (141);
47674 print#1,nz$(a+9)                        1455
47675 print#4,"[11spaces]";chr$ (254);chr$    2588
      (141);
47676 print#1,nz$(a+10)                       1540
47677 print#4,"[13spaces]";chr$ (14);chr$     3314
      (254);chr$ (141);
47678 print#1,nz$(a+11)                       1544
47679 print#4,"[15spaces]";chr$ (14);chr$     2990
      (254);chr$ (141);
47680 print#1,nz$(a+12)                       1548
47681 print#4,"[17spaces]";chr$ (14);chr$     3479
      (254);chr$ (141);
47682 print#1,nz$(a+13)                       1552
47683 print#4,"[19spaces]";chr$ (14);chr$     3650
      (254);chr$ (141);
47684 print#1,nz$(a+14)                       1556
47685 print#4,"[25spaces]";chr$ (254);chr$    3341
      (141);
47686 print#1,nz$(a+15)                       1560
47687 print#4,"[22spaces]";chr$ (254);chr$    2785
      (141);
47688 print#1,nz$(a+16)                       1564
47689 print#4,"[24spaces]";chr$ (254);chr$    3169
      (141);
47690 print#1,nz$(a+17)                       1568
47691 print#4,"[25spaces]";chr$ (254);chr$    3341
      (141);
47692 print#1,nz$(a+18)                       1572
47693 print#4,"[26spaces]";chr$ (254);chr$    3430
      (141);
47694 print#1,nz$(a+19)                       1576
47695 print#4,"[27spaces]";chr$ (254);chr$    3098
      (141);
47696 print#1,nz$(a+20)                       1542
47697 print#4,"[28spaces]";chr$ (254);chr$    2689
      (141);
47698 print#1,nz$(a+21)                       1546
47699 print#4,"[29spaces]";chr$ (254);chr$    2636
      (141);
47700 close 1:close 4:return                   796
47800 us=1:open 4,4:print "[clr]daten einer    4610
       partei ausdrucken[down,cbm 3]"
47801 for x=1 to za                            873
47802 print x;"[left]) ";pn$(x)               1385
47803 next                                     130
47804 print "[down,cbm 6]welcher partei       5671
       daten soll ich[11spaces]
      niederschreiben? [cyn]";
47805 gosub 63000:x=val (a$(2))               1664
47806 print#4,chr$ (14);"[down]Daten der      3917
       Partei [up]";pn$(x):print#4:goto
      47070
48000 rem wahl
48010 print "[cbm 7,clr]wahlkampf"            1252
48020 gosub 62900                              659
48021 if za=8 then goto 48140                 1463
48030 for x=9 to za:print "[clr]"             1282
48050 print "wie soll der wahlkampf der       6111
       partei:[6spaces]";pn$(x);" gefuehrt
       werden?"
48060 print "[down]1) kein wahlkampf"         2072
48070 print "2) wenig wahlkampf"              1960
48080 print "3) normaler wahlkampf"           2328
48090 print "4) starker wahlkampf"            2220
48100 print "5) intensiver wahlkampf"         2376
48110 print "[down]gib die staerke ein: ";    2375
48120 gosub 63000                              591
48125 if val (a$(2))<1 or val (a$(2))>5 then  2974
      goto 48110
48130 wa(x)=val (a$(2)):ge(x)=ge(x)+(wa(x)*   3864
      10000):next
48140 for x=1 to za                            873
48160 ge(x)=int (ge(x)-(st(x)*2.5))+(mi(x)*   4098
      20)
48170 next :for x=1 to za:ae(x)=ae(x)-wa(x):  2967
      next
48171 gosub 62900:gosub 50000:gosub 26000:    2498
      goto 100
49000 rem eine rede halten
49010 print "[clr]eine rede halten"           1491
49020 print "[red]wir haben folgende         3019
       parteien:[cbm 7]"
49030 for x=1 to za:print x;"[left]) ";       2446
      pn$(x):next
49040 print "[down,cbm 6]fuer welche partei  4114
       spricht der redner?"
49050 gosub 63000:if val (a$(2))=0 or val     3973
      (a$(2))>za then goto 49040
49060 pz=val (a$(2)):print "[down]bitte eine  5294
       taste druecken!":poke 198,0:wait 198,
      1:poke 198,0
49070 print "[clr,down,cbm 7]ueber welches    7055
       thema soll gesprochen[5spaces]
      werden?":ae(pz)=ae(pz)-1
49080 print "[down,red]1)[wht] die eigenen    3061
       leistungen"
49090 print "[red]2)[wht] eine andere        2501
       partei"
49100 print "[red]3)[wht] umweltschutz"       2134
49110 print "[red]4)[wht] atomenergie"        2359
49120 print "[red]5)[wht] sozialpolitik"      2148
49130 print "[red]6)[wht] friedenspolitik"    2498
49140 print "[red]7)[wht]                     2576
       sicherheitspolitik"
49150 print "[red]8)[wht] die naechste wahl"  2279
49160 print "[red]9)[wht] das drogenproblem   3557
      [yel]":poke 198,0
49170 gosub 63000:if val (a$(2))=0 or val     3834
      (a$(2))>9 then goto 49070
49180 st(pz)=val (a$(2))+mi(pz)*po(pz)+       3846
      st(pz)
49190 mi(pz)=int ( rnd (1)*mi(pz)+100)+       3416
      mi(pz)
49200 print "[down,cbm 7]du haelst nun die    2565
       rede!"
49201 b=int ( rnd (1)*5)+1                    1408
49202 for x=1 to 1000:next                     966
49203 if b=1 then print "du warst ein sehr     5062
       schlechter redner":ae(pz)=ae(pz)-2
```


WWW.HOMECOMPUTERWORLD.COM

```
49204 if b=2 then print "du warst kein sehr guter redner":ae(pz)=ae(pz)-1   5192
49205 if b=3 then print "du warst ein mittelmaessiger redner"               3753
49206 if b=4 then print "du warst ein guter redner":ae(pz)=ae(pz)+1         4120
49207 if b=5 then print "du warst ein exellenter redner":ae(pz)=ae(pz)+2    4771
49210 print "[2down,cbm 6]":goto 53520                                      1499
50000 goto 50050                                                            653
50010 for x=1 to za                                                         873
50011 ge(x)=int (st(x)*2.3)+ge(x)+(mi(x)*5)                                 3420
50020 si(x)=int (5*pr(x))                                                   1660
50021 mi(x)=int ( rnd (1)*(mi(x)+1000))+(mi(x)-500):if mi(x)<1 then mi(x)=1 5980
50022 if mi(x)<si(x) then mi(x)=si(x)+29                                    3033
50030 if pr(x)<3.5 then si(x)=0                                             2057
50040 next x:goto 50100                                                     851
50050 s=0                                                                   376
50060 for x=1 to za:s=s+st(x):next                                          1660
50070 for x=1 to za                                                         873
50080 pr(x)=(100/s)*st(x):next                                              2650
50090 goto 50010                                                            589
50100 for x=1 to za                                                         873
50110 if si(x)=0 then op(x)=0:goto 50140                                    2177
50120 if es(x)>7 then op(x)=2                                               1674
50130 if es(x)<-7 then op(x)=1                                              1592
50140 next                                                                  130
50141 for x=1 to za                                                         873
50142 if pr(x)>62.34 then ae(x)=int (ae(x)-10):goto 48140                   3711
50143 if ae(x)<1 then ae(x)=1                                               1790
50144 if ae(x)>49 then ae(x)=50                                             1295
50145 if ae(x)>39 then ae(x)=ae(x)-5                                        2170
50146 if ae(x)>30 and ae(x)<40 then ae(x)=ae(x)-3                           2934
50147 if ae((x))<7 then ae(x)=ae(x)+1                                       2320
50148 next                                                                  130
50149 gosub 38000:gosub 30100:return                                        1654
51000 rem gr. darstellung reichstag
51010 print "[clr,cyn]darstellung der sitzverteilung im[7spaces]reichstag [down,cbm 7]"   5066
51030 print "name der partei[2spaces,yel]B [cbm 7] %[yel,4spaces]B[cbm 7]sitze"   3466
51040 print "[yel]CCCCCCCCCCCCCCCCCCCC[shift +]CCCCCC[shift +]CCCCCC[cbm 7]"     4852
51050 for x=1 to za                                                         873
51060 poke 214,(4+x):poke 211,0:sys 58640                                   2487
51070 print pn$(x)                                                          618
51080 poke 214,(4+x):poke 211,16:sys 58640                                  2604
51081 if pr(x)=-1 then print "[yel] B[cbm 7] n.tg":goto 51100               3264
51090 print " [yel]B[cbm 7]";left$ ( str$ (pr(x)),5)                        2124
51091 if pr(x)>=10 then gosub 51400                                         1421
51100 poke 214,(4+x):poke 211,24:sys 58640                                  2601
51101 if si(x)=0 then print "[yel]B[cbm 7] keine":goto 51120                2818
51110 print "[yel]B[cbm 7]";si(x)                                           1300
51120 next                                                                  130
51125 poke 198,0:wait 198,1:poke 198,0                                      2514
51130 print "[clr]graphische darstellung der sitzever-[4spaces]haeltnisse:"  4509
51140 poke 214,22:poke 211,0:sys 58640                                      2591
51150 y=22:a=0                                                              934
51160 for x=1 to 18                                                         770
51170 for z=1 to za:poke 646,z                                              1819
51180 if si(z)>a then print "[rvs,2spaces,off] ";:goto 51190                2422
51185 print "[3spaces]";                                                    626
51190 next z                                                                310
51200 y=y-1:a=a+15                                                          1158
51210 poke 214,y:poke 211,0:sys 58640                                       2527
51220 next x                                                                306
51230 poke 214,23:poke 211,0:sys 58640                                      2655
51240 for x=1 to za:poke 646,15                                             1855
51250 print left$ (pn$(x),2);" ";:next x                                    2055
51330 poke 198,0:wait 198,1:poke 198,0:poke 646,14:goto 100                 3945
51400 poke 214,4+x:poke 211,16:sys 58640                                    2616
51410 print " [yel]B[cbm 7]";left$ ( str$ (pr(x)),6):return                 2366
52000 rem einstufung der parteien
52010 print "[clr] politische einstufung der parteien[2down]"              3684
52030 print "name der partei[2spaces]B einstufung"                          2954
52040 print "CCCCCCCCCCCCCCCCCC[shift +]CCCCCCCCCCCCCCCCCC"                 4854
52050 for x=1 to za                                                         873
52060 poke 214,(4+x):poke 211,0:sys 58640                                   2487
52070 print pn$(x)                                                          618
52080 poke 214,(4+x):poke 211,17:sys 58640                                  2606
52090 print "B";be$(es(x))                                                 1163
52270 next x:gosub 62010                                                    1027
52280 poke 198,0:wait 198,1:poke 198,0:goto 100                             2874
53000 goto 53300                                                            609
53001 print "[clr]gruendung einer neuen partei[down]"                       3009
53010 print "es gibt:";za;"parteien"                                        2450
53020 if za=13 then print "das sind genug parteien!"                        3206
53030 if za=13 then print "taste druecken"                                  2482
53040 if za=13 then poke 198,0:wait 198,1:poke 198,0:goto 100               3794
53050 za=za+1                                                               813
53060 print "[down]name der partei? ";:gosub 63000                          2629
53061 pn$(za)=a$(2)                                                         1401
53070 print "spitzenkandidat? ";:gosub 63000                               2858
53071 sk$(za)=a$(2):ae(za)=19                                               2168
53080 print "[down]welche politische richtung soll die"                    4073
53090 print "neue partei vertreten?"                                       2378
53100 print "[down]1) linksradikal"                                        1966
53110 print "2) links"                                                     1131
53120 print "3) liberal"                                                   1122
53130 print "4) rechts"                                                    1154
53140 print "5) rechtsradikal"                                             2043
53150 get a$:if a$="" then 53150                                            1585
53160 if a$="1" then es(za)=2:ae(za)=ae(za)-2:goto 53220                    3930
53170 if a$="2" then es(za)=4:ae(za)=ae(za)-1:goto 53220                    4090
53180 if a$="3" then es(za)=7:ae(za)=ae(za)+1:goto 53220                    4187
53190 if a$="4" then es(za)=9:ae(za)=ae(za)+2:goto 53220                    4093
53200 if a$="5" then es(za)=13:ae(za)=ae(za)-2:goto 53220                   4286
53210 goto 53150                                                            673
53220 print "[down]ok. die partei ist ab sofort aktiv"                      3293
53230 pr(za)=-1:si(za)=0:ge(za)=20000:st(za)=int ( rnd (1)*110000)+9000     6278
53231 mi(za)=int ( rnd (1)*8000)+1:op(za)=0:sa(za)=int ( rnd (1)*600000)+118000   5871
53232 ap$(za)="keines"                                                      1706
53240 poke 198,0:wait 198,1:poke 198,0:return                               2886
53300 print "[clr]parteienstruktur veraendern"                              3084
53310 print "[3down]f1) gruendung einer neuen partei"                       3896
53320 print "f3) aufloesung einer partei"                                  2752
53330 print "f5) zurueck ins menue"                                        2227
53340 get a$:if a$="" then 53340                                            1585
53350 if a$="[f5]" then return                                              1325
53360 if a$="[f1]" then goto 53001                                          1736
53365 if a$="[f3]" then goto 53400                                          1768
53400 print "[clr,red]aufloesung einer der alten parteien"                  4184
53410 print "[down,wht]es gibt folgende parteien:"                          2645
53420 for x=1 to za                                                         873
53430 print "[cbm 7]";pn$(x);" [cbm 6]";sk$(x)                              1970
53440 next x:print "[cbm 7]"                                                995
53450 print "wie heisst du? ";:gosub 63000:a$=a$(2)                         3643
53460 print "welche partei aufloesen? ";:gosub 63000:b$=a$(2)               4772
53470 for x=1 to za                                                         873
53480 if sk$(x)=a$ then goto 53500                                          1392
```


WWW.HOMECOMPUTERWORLD.COM

```
53490 next x:goto 53510                                          1076
53500 if b$=pn$(x) then goto 53530                               1666
53510 print "[down]du kannst keine partei                        6880
      aufloesen, deren spitzenkanidat du
      nicht bist!"
53520 print "taste druecken":poke 198,0:wait                    3750
      198,1:poke 198,0:return
53530 if x<=8 then goto 53510                                    1779
53531 print "o.k."                                               908
53540 print "die partei ist jezt                                3067
      aufgeloest."
53550 for y=x to za-1                                            1194
53560 pr(y)=pr(y+1)                                              1570
53570 si(y)=si(y+1)                                              1870
53580 pn$(y)=pn$(y+1)                                            1637
53590 sk$(y)=sk$(y+1)                                            1760
53600 es(y)=es(y+1)                                              1750
53610 st(y)=st(y+1)                                              1770
53620 next y                                                     308
53630 za=za-1:gosub 50000:goto 53520                             2304
54000 for x=1 to za                                              873
54010 st(x)=abs ( int (( rnd (1)*(st(x)+                         4806
      10000))+st(x)-10000))
54011 if st(x)>7500000 then goto 54010                           2281
54020 next x                                                     306
54030 print "[clr]es ist gewaehlt worden:"                       2527
54040 print "[down]hier sind die ergebnisse:                    2818
      "
54050 for x=1 to za:print pn$(x);":":next x                      2223
54060 for x=1 to za                                              873
54070 poke 214,2+x:poke 211,7:sys 58640                          2652
54080 print st(x);                                               841
54090 poke 214,2+x:poke 211,20:sys 58640                         2613
54100 print "stimmen":next x                                    1553
54110 poke 198,0:wait 198,1:poke 198,0:gosub                     3726
      50000:goto 100
55000 rem druckerausgabe der machtver -
            haeltnisse"
55001 open 4,4                                                   600
55010 print#4,"sitzverteilung im"                                2084
55011 print#4,"reichstag"                                        1410
55020 print#4                                                    256
55030 print#4,"name der partei[2spaces]                          2458
      Bsitze"
55040 print#4,"CCCCCCCCCCCCCCCCC[shift +]                        3499
      CCCCC"
55050 for x=1 to za                                              873
55060 print#4,pn$(x);( left$ ("[17spaces]",                      4086
      17-len (pn$(x))));"B";si(x)
55080 next x:print#4:close 4:gosub 47500:                        2102
      goto 100
57000 rem finanzielle situation
57010 print "[clr,red]die finanzielle                           4284
      situation der parteien:[cbm 6,down]"
57020 for x=1 to za                                              873
57030 print pn$(x);":";tab(12-val (pn$(x)));                     3847
      ge(x);"taler"
57040 next                                                       130
57050 print "[down,cbm 7]taste druecken":                       3907
      poke 198,0:wait 198,1:poke 198,0:
      return
58000 rem verbot einer partei
58010 print "[clr]eine partei wird verboten                     2986
      [red,down]"
58020 if es(x)<0 then print "wegen                               5150
       linksfaschistischer aktivitaeten"
58021 if es(x)<0 then print "wird die                            3550
       politische partei"
58030 if es(x)>15 then print "wegen                              5833
       rechtsfaschistischer aktivitaeten
       wied die partei"
58040 print pn$(x);" verboten.[down,cbm 7]":                     4592
      a$=sk$(x):b$=pn$(x):goto 53470
59000 print "[clr,wht]informationen ueber                       3636
       die parteien[down]"
59010 print "[red]wir haben folgende                           3019
       parteien:[cbm 7]"
59020 for x=1 to za:print x;"[left]) ";                          2446
      pn$(x):next
59030 print "[down,cbm 6]ueber welche partei                    4303
       soll ich berichten?"
59040 gosub 63000:if val (a$(2))=0 or val                        3966
      (a$(2))>za then goto 59030
59050 pz=val (a$(2)):print "[down]bitte eine                     5422
       taste druecken!":poke 198,0:wait 198,
      1:poke 198,0
59060 print "[clr,wht]informationen ueber ";                    3238
      pn$(pz)
59070 print "[down,cbm 7]spitzenkandidat: ";                    3037
      sk$(pz)
59080 print "[down]die finanzielle situation                   3772
       der partei:"
59090 print "die partei hat";ge(pz);"taler"                    2466
59100 print "[down,cbm 3]situation in der                      3736
       oeffentlichkeit:"
59110 print "die partei ist ";                                 1558
59111 if ae(pz)<11 then print "sehr                             2702
       unbeliebt";
59112 if ae(pz)>10 and ae(pz)<21 then print                      3616
      "nicht sehr beliebt";
59113 if ae(pz)>20 and ae(pz)<31 then print                      3205
      "akzeptiert";
59114 if ae(pz)>30 and ae(pz)<41 then print                      2824
      "beliebt";
59115 if ae(pz)>40 and ae(pz)<51 then print                      3350
      "sehr beliebt";
59116 print "."                                                 422
59117 print "einstufung: ";be$(es(pz))                         2645
59120 print "sitze im parlament:";si(pz)                       2985
59128 if pr(pz)=-1 then print "stimmen bei                      6680
       der letzten wahl zum reichs-
      parlament: keine"
59129 if pr(pz)=-1 then goto 59137                               1589
59130 print "stimmen bei der letzten wahl                       5511
       zum reichs-parlament:";st(pz)
59137 if pr(pz)=-1 then print "prozente:                         2793
       keine"
59138 if pr(pz)=-1 then print "diese partei                      4751
       hat auf der letzten wahl"
59139 if pr(pz)=-1 then print "nicht                             3764
       kanidirt!":goto 59150
59140 print "prozente:";pr(pz)                                  2123
59150 print "mitglieder:";mi(pz)                               2381
59180 print "intensietaet des letzten                          3801
       wahlkampfes:"
59185 print wa$(wa(pz))                                          1326
59190 goto 57050                                                 681
60000 data 8,20000000,250000                                     1423
60010 data prf,27.5,von beisstaler,10,                           5333
      5500000,500000,2,5,35,55000
60020 data cdp,12.5,loewenthalinski,9,                           5369
      2500000,250000,2,4,27,25000
60030 data sdp,8.1,blas,7,1700000,130000,1,                      3937
      3,21,17000
60040 data pdiib,5.2,schilli,0,1700000,                          4767
      125000,1,2,11,17000
60050 data slpb,7.3,meierhof,6,1400000,                          4880
      321000,1,3,20,14000
60060 data usp,24.1,mussnichs,5,4400000,                         4565
      100000,1,5,31,44000
60070 data cbp,10.1,miesemann,8,2200000,                         4413
      130000,2,2,25,22000
60080 data kep,4.9,handlos,1,900000,72000,1,                     4011
      1,9,9000
60090 data linksfaschistisch,linksextrem,                        7193
      linksradikal,dogmatisch links,links
60100 data gemaessigt links,linksliberal,                        6678
      liberal,rechtsliberal,rechts
60110 data konservativ,rechtsstehend,                            7148
      rechtsgerichtet,rechtsradikal,
      rechtsextrem
60115 data rechtsfaschistisch                                    1858
60120 data kein wahlkampf,wenig wahlkampf,                       4662
      normaler wahlkampf
60125 data starker wahlkampf,intensiever                         3476
       wahlkampf
60200 data finanzminister,innenminister,                         7215
      aussenminister,verteidigungsminister
60210 data forschungsminister,                                   5843
      wirtschaftsminister,justizminister
60220 data umweltminister,verkehrsminister,                      5805
      entwicklungshilfeminister
60230 data finanzminister,                                       5928
      gesundheitsminister,familienminister
60231 data 60,66,153,165,165,129,66,60                           2956
60232 data 16,48,97,255,255,1,0,0                                2387
```


WWW.HOMECOMPUTERWORLD.COM

```
60233 data 0,50,123,201,201,127,62,0                    2625
60234 data 0,44,126,211,211,126,44,0                    2677
60235 data 126,255,137,137,207,70,0,0                   3530
60236 data 6,9,9,137,255,137,0,0                        2572
60237 data 8,14,9,1,9,14,8,0                            1650
60238 data 0,1,15,5,8,8,0,0                             1679
60239 data 6,9,9,9,0,0,0,0                              1696
60240 data 255,136,8,8,7,0,0,0                          2504
60241 data 66,129,255,129,129,129,126,0                 2523
60242 data 10,17,62,80,144,144,80,62                    2467
60243 data 17,34,0,60,66,129,129,129                    2545
60244 data 0,60,66,129,137,137,78,8                     3065
60245 data 7,0,0,0,0,0,0,17,95,17                       2566
60246 data 0,31,8,16,16,15                              1676
60310 data 3,1,254,64,32,64,255,1                       2330
60320 data 2,0,255,144,144,255,0,255                    2733
60330 data 144,156,147,96,128,128,255,128               3028
60340 data 128,0,255,144,144,128,0,255                  2765
60350 data 145,145,129,0,255,1,1,1                      2651
60360 data 0,255,129,129,66,60,0,0                      2271
60500 data 169,1,32,195,255,169,36,141,240,             7271
      3,169,48,141,241,3,169,1,162,8,160,0
60505 data 32,186,255,169,2,162,240,160,3               3195
60510 data 32,189,255,32,192,255,169,64,32,             5816
      144,255,162,1,32,198
60520 data 255,32,144,255,32,207,255,32,207,            6051
      255,32,207,255,32,207
60530 data 255,201,0,240,58,32,204,255,32,              6358
      228,255,201,32,208,3,32,196,3
60540 data 162,1,32,198,255,32,207,255,168,             6037
      32,207,255,72,152,170
60550 data 104,32,205,189,169,32,32,210,255,            6672
      32,207,255,201,0,208,8,169,13
60560 data 32,210,255,76,115,3,32,210,255,              6556
      76,162,3,169,1,32,195,255,32
60570 data 204,255,169,204,133,178,96,32,               5997
      228,255,201,32,208,249,96
60580 data 173,17,208,141,30,203,173,24,208,            6088
      141,31,203,169,59,141
60590 data 17,208,169,0,16,24,208,169,96,               4184
      169,254,141,0
61000 data russland,-3                                  1332
61010 data amerika,3                                     950
61020 data australien,0                                 1129
61030 data neuseeland,1                                 1172
61040 data bulgarien,-1                                 1439
61050 data tuerkei,-3                                   1151
61060 data saudi-arabien,-1                             1606
61070 data israel,1                                      778
61080 data suedafrika,-2                                1323
61090 data zaire,2                                       631
61100 data aegypten,1                                   1044
61110 data china,3                                       804
61120 data japan,2                                       671
61130 data indien,3                                      904
61140 data indonesien,1                                 1447
61150 data kuba,-2                                       637
61160 data togo,2                                        935
61170 data brasilien,2                                  1169
61180 data chile,-2                                      947
61190 data bolivien,1                                   1378
61200 data kanada,2                                      708
61210 data argentinien,3                                1517
61220 data venezuela,-1                                 1350
61230 data mexico,2                                     1035
61240 data norwegen,1                                   1512
61250 data daenemark,-1                                 1485
61260 data frankreich,2                                 1369
61270 data polen,-4                                      948
61280 data spanien,2                                     905
61290 data italien,3                                     871
61300 data holland,2                                     995
62000 rem erzeugen eines einzeltons
62001 poke 54296,5:poke 54277,25:poke 54278,            6201
      68:poke 54272,69:poke 54273,29
62002 poke 54276,17:for xx=1 to 50:next :               5509
      poke 54296,0:poke 54276,0:poke 54273,0
62003 poke 54272,0:return                                917
62010 rem erzeugen von melodie 1
62011 poke 54296,5:poke 54277,25:poke 54278,            3731
      68
62012 poke 54273,17:poke 54272,103:poke                 4636
      54276,17:for x=1 to 250:next
62013 poke 54273,21:poke 54272,237:poke                 4549
      54276,17:for x=1 to 250:next
62014 poke 54273,26:poke 54272,20:poke                  4198
      54276,17:for x=1 to 250:next
62015 poke 54273,23:poke 54272,59:poke                  4237
      54276,17:for x=1 to 250:next
62016 poke 54273,29:poke 54272,69:poke                  4245
      54276,17:for x=1 to 250:next
62017 poke 54273,26:poke 54272,20:poke                  4198
      54276,17:for x=1 to 250:next
62018 poke 54273,23:poke 54272,59:poke                  4237
      54276,17:for x=1 to 250:next
62019 poke 54273,21:poke 54272,237:poke                 4549
      54276,17:for x=1 to 250:next
62020 poke 54273,19:poke 54272,137:poke                 4682
      54276,17:for x=1 to 250:next
62021 poke 54273,17:poke 54272,103:poke                 4524
      54276,17:for x=1 to 500:next
62022 poke 54296,0:poke 54272,0:poke 54273,             4665
      0:poke 54296,0:return
62030 rem erzeugen von melodie 2
62031 poke 54296,5:poke 54277,25:poke 54278,            3731
      68
62032 for y=1 to 5:for x=0 to 70:poke 54272,            5964
      103:poke 54273,x:poke 54276,17:next :
      next
62033 poke 54296,0:poke 54277,0:poke 54278,             6139
      0:poke 54272,0:poke 54273,0:poke
      54276,0
62034 poke 54296,0:return                               1077
62900 print "[clr]jetzt wird gewaehlt":for              3303
      x=1 to za
62901 if ae(x)<1 then ae(x)=1                           1790
62902 next :for x=1 to za                                678
62903 y=sa(x)/30*ae(x)+(sa(x)/5):if sa(x)=>             5721
      70000 then y=int ( rnd (1)*71000)+
      69600
62904 yy=int ( rnd (1)*sa(x))+sa(x)+(sa(x)/             4984
      10):xy=int (sa(x)/10)
62905 st(x)=int ( rnd (1)*y)+(y-yy-xy):                 6593
      st(x)=int (((st(x)*wa(x)/3.5)/30)*
      ae(x))
62906 if st(x)<=mi(x) then st(x)=int ( rnd              5041
      (1)*(mi(x)+9000))+mi(x)
62907 if op(x)=0 then op(x)=1:st(x)=int                 3807
      (st(x)*11.3)
62908 st(x)=int ((st(x)+sa(x))/25*ae(x)):               3554
      next
62909 for x=1 to za                                      873
62910 if ae(x)<>1 then ae(x)=ae(x)+3                    2448
62911 if ae(x)>30 and ae(x)<40 then ae(x)=              2934
      ae(x)-3
62912 if ae(x)>39 then ae(x)=ae(x)-5                    2170
62913 if ae(x)<8 then ae(x)=ae(x)+1                     2201
62920 next :return                                       304
63000 a$(2)=""                                           810
63001 get a$(1):if a$(1)="" then goto 63001             2354
63002 if asc (a$(1))=13 then print :poke                2727
      54296,0:return
63003 if asc (a$(1))=20 and len (a$(2))<1               3315
      then goto 63001
63004 if asc (a$(1))=20 then print "[asc20]"            1687
      ;
63005 if asc (a$(1))=20 then a$(2)=left$                3900
      (a$(2),( len (a$(2))-1)):goto 63001
63006 if asc (a$(1))<32 then goto 63001                 2043
63007 if asc (a$(1))>95 then goto 63001                 2087
63008 if len (a$(2))>10 then goto 63001                 2073
63009 print a$(1);:a$(2)=a$(2)+a$(1):gosub              3254
      62000:goto 63001
63010 z$="000"+( str$ (za-8))+"00"                      1675
63011 z$(1)=left$ (z$,3):z$(2)=right$ (z$,              4243
      3):z$=z$(1)+z$(2)
63020 ti$="000000":goto 10000                           1600
63100 sa( val (a$(1)))=sa( val (a$(1)))/100*            3414
      90:goto 20352
63200 sa( val (a$(1)))=sa( val (a$(1)))/100*            3331
      110:goto 20352
```

Das richtige Programm für Kohl, Strauß, Vogel & Co.

WWW.HOMECOMPUTERWORLD.COM

Schillernd texten mit C16 und „AT-Text“

System: C16, Floppy, Drucker/ Plotter
Typ: Anwendung
Sprache: Basic

Elektronisch Briefe schreiben: Mit dem C16 kein Problem — vorausgesetzt man hat „AT-Text“. Das Textverarbeitungsprogramm funktioniert mit Floppy, Datasette und verschiedenen Drukker-/Plottertypen.

Nachdem das Programm komplett eingetippt ist, muß man die Daten vor!! RUN abspeichern.

Während des ersten Programmstartes fragt es nach der Peripherie, mit der man arbeitet. Ist die Eingabe erfolgt, schreibt sich das Programm entsprechend um.
Diese Arbeitsversion speichert man ab.
Die Belegung der Funktionstasten ist:

F1 Sprung zum Text-Anfang
F2 Eine Zeile vor
F3 Zum Druck-Menü
F4 Sprung zum Text-Ende
F5 Eine Zeile zurück
F6 Zum Save-/Load-Menü
F7 Zeile einfügen
F8 Zeile löschen (HELP)

Die Funktion F4 funktioniert ab 9 Zeilen.
Vom Text sind jeweils 19 Zeilen sichtbar. Die Eingabezeile befindet sich in der Mittelzeile (Cursor). Schreiben erfolgt im INPUT-Modus. Dadurch bedingt müssen Kommata, Semikolons und Doppelpunkte simuliert werden, da sie sonst beim Drucken als Trennzeichen interpretiert und dahinter stehende Zeilenteile nicht zum Druck gelangen:

Komma :CONTROL + @
Semikolon :CONTROL + o
Doppelpunkt :CONTROL + p

Die über die ESC erreichbaren Textbearbeitungsfunktionen bleiben erhalten. Mit 'Shift/-Space' rückt man Zeichen ein. Vorsicht ist bei Grafik-Zeichen in der 1. Spalte angebracht, da mit einigen von ihnen die Funktionstasten belegt sind oder die Satzzeichen simulieren.

Vor Laden eines Textes ist zu empfehlen, mit 'Neuanfang' den Schreibspeicher von alten Texten zu befreien. 'Zuladen' setzt weitere Texte nahtlos an den im Textspeicher befindlichen an sofern der Speicherplatz genügt. 'Resave' erlaubt Überschreiben bestehender Text-Files.

Axel Töpfer

Erklärung zum Listing:

Zeilen Zweck
0—8
Vorbereitungen (Beim 1. Start Aufruf von Direktprogrammierung)
10—120
KEY-Belegung ändern
130—990
Dimensionierung und Zeilen-Info
1000—1200
Hauptschleife und Eingabe
2000—2070
F4 — Zum Textende
2200—2250
F2 — Line up
2300—2350
F5 — Line down
3000—3160
F3 — Druck-Menü
3170—3430
Druck zeilenweise
3500—3630
F7 — Insert line
3700—3930
Druck seitenweise
4000—4130
F6 — Save/Load-Menü Resave-Routine
4200—4229
Save-Routine
4300—4370
Load-Routine
4400—4480
Zuladen
4500—4650
F8 — Delete line
5000—5090
Test ob 1. Zeichen im Eingabe-String ein Steuerzeichen ist.
6000—6090
Speicher voll?
7000—7050
Zeilenanzeige/Kopf
9000—9070
SUB Textanfang
10000—10080
SUB Maske
11000—11110
SUB Satzzeichen-Transformation
13000—13530
6 SUBs für Fenster
15000—15040
SUB Sicherheitsabfrage Progr. Ende?
16000—16040
SUB Sicherheitsabfrage Neustart?
18000—18040
SUB Dokument-Name?
19000—19070
SUB Diskerrorcheck
20000—21230
Programmteil zum Anpassen des Progr. im programmierten Direktmodus

Erklärung der Variablen:
SV$ = Save-String f. Floppy
LD$ = Load-String f. Floppy

WWW.HOMECOMPUTERWORLD.COM

RE$	= Resave-String f. Floppy	S$	= H.V. bei Druck	S2	= Letzte Zeile Ausdruck
VN$	= Verneinung b. Resave-Versuch mit Datasette	B1$ — B6$	= Transformation Satzzeichen	LO	= Leerzeilen oben/Druck
T$	= Hilfsvariable (H.V.)	L$	= H.V.	LU	= Leerzeilen unten/Druck
t$(x)	= Textspeicher	P$	= H.V. bei Direktprogr.	ZL	= Zeilen pro Seite/Druck
K$	= H.V.	A	= H.V. bei Direktprogr.	DZ	= H.V. zum Errechnen der Anzahl der zu druckenden Seiten
A$	= Zwischenspeicher Text (Bei Direktprogr. Datazwischenspeicher)	Z	= Zeilenzähler	SD	= H.V. wie DZ
B$	= Steuerzeichen Funktion	ZG	= Zeilen Maximum	I	= H.V.
NA$	= Dokument-Name	ZM	= Höchste benutzte Zeile	X	= H.V.
X$	= H.V.	ZD	= Schleifenzähler Delete	Q	= H.V.
Y$	= H.V.	ZI	= Schleifenzähler Insert	J	= H.V.
F$	= H.V.	ZU	= Zuzuladende Zeilen	ZZ	= H.V.
		S1	= Erste Zeile Ausdruck		

```
  0 rem ********** a t - t e x t *********
  1 sv$=",s,w":ld$=",s,r":re$="@0:":vn$="    6591
    [flashon]Nur mit Floppy moeglich
    [flashoff]"
  2 goto 20000                                567
  3 poke 65286,27                            1036
  7 trap 10                                   505
 10 rem ** keybelegung **
 20 print "[2home,clr]";                      780
 40 key 1,"[cbm q]"+chr$ (13)                1606
 42 key 4,"[cbm w]"+chr$ (13)                1485
 44 key 2,"[cbm e]"+chr$ (13)                1449
 46 key 5,"[cbm r]"+chr$ (13)                1471
 48 key 3,"[cbm t]"+chr$ (13)                1482
 50 key 6,"[cbm y]"+chr$ (13)                1553
 52 key 7,"[cbm d]"+chr$ (13)                1634
 54 key 8,"[cbm f]"+chr$ (13)                1621
 70 print chr$ (142)                          804
 80 print "[clr,2down,3spaces,rvs,flashon]    3271
     achtung neue key-belegung !!! [2down]
    "
 81 trap 80                                   519
 82 print " f1 = text anfang"                1489
 84 print "[down] f4 = text ende"            1640
 86 print "[down] f2 = line up"              1208
 88 print "[down] f5 = line down"            1708
 90 print "[down] f3 = print"                1499
 92 print "[down] f6 = save/load"            1971
 94 print "[down] f7 = insert line"          1725
 96 print "[down] f8 = delete line[down]"    1575
110 for z=0 to 39:print "[rvs]*[off]";:       1771
    next
120 print "[down,3spaces,rvs,flashon]        4979
     programmstart mit leertaste !!! ":
    poke 239,0:get key t$
130 print "[clr]";                            495
460 :
470 print "[ctrl n]":print chr$ (8)           907
480 zg=int (( fre (x)-500)/40)               1930
490 print "[clr,2down,2right]Es stehen"       3292
    zg"Zeilen zu je "
500 print "[down,2right]40 Zeichen zur        3606
     Verfuegung."
538 dim t$(zg)                                982
570 print "[4down,8right](Weiter mit          3621
     'SPACE')"
580 trap 570:get key k$:if k$<>chr$ (32)      2724
    then 580
970 print chr$ (27)"m"                       1089
980 gosub 10000:gosub 7000                   1381
990 gosub 13100:print "[clr]";:gosub 9000    1947
1000 rem hauptschleife
1001 trap 980                                 682
1005 gosub 7030:rem kopf-anz.zeile            670
1020 gosub 13000:if z<=zg then print t$(z)    2323
1030 poke 19,64                               807
1040 input a$                                 407
1050 poke 19,0                                577
1052 goto 5000:rem control asc?               638
1060 t$(z)=a$                                 856
1070 gosub 13100:rem pos 2                    731
1090 if z<zg then print chr$ (27)"v"          2229
1091 if z+10=<zg then 1200                    1496
1092 z=z+1                                    400
1095 gosub 6000:rem ende ?                    644
1097 if zm<z then zm=z                       1612
1100 goto 1000                                559
1200 gosub 13200:print t$(z+10):goto 1092     2456
2000 rem text ende
2002 if zm<9 then 1000                       1164
2005 gosub 13100:print "[clr]":rem pos 2      1353
2010 gosub 13500:rem pos 6                    763
2020 print "[home]";                          493
2030 for i=zm-9 to zm                        1420
2040 print t$(i)                              715
2050 next i                                   276
2060 z=zm-1                                   626
2070 goto 1000                                559
2200 rem line up / f2"
2205 print " [home]"t$(z):gosub 13100:rem p   1599
    os2
2210 gosub 6000                               573
2220 if z<zg then print chr$ (27)"v":z=z+1    2658
2230 gosub 13200:rem pos 3                    739
2240 if z+9=<zg then print t$(z+9)            2068
2250 goto 1000                                559
2300 rem line down / f5
2305 print " [home]"t$(z):gosub 13100:rem p   1599
    os2
2310 gosub 6000                               573
2320 if z>0 then print chr$ (27)"w":z=z-1     2462
2330 gosub 13300:rem pos 4                    747
2340 if z-8=>0 then print t$(z-8)            1980
2350 goto 1000                                559
3000 rem print
3001 trap 3000                                641
3010 gosub 13100:print "[clr]";:rem pos 2     1524
3020 print "[down,2right]Printer-Menue"       2355
3030 print "[2right,13cbm y]"                 3175
3040 print "[down,2right]Seitenweise           2553
    [5spaces]: 1"
3050 print "[down,2right]Zeileweise            2131
    [6spaces]: 2"
3060 print "[down,2right]Schreib-modus         2643
    [3spaces]: 3"
3065 print "[down,2right]Ende[12spaces]: 4"    1753
3070 get key x$:x=val (x$)                    1539
3080 if x<1 or x>4 then 3070                  1463
3090 on x goto 3700,3100,990,15000            1964
3100 rem teilw.
3110 gosub 13400:print "[clr]";:rem pos 5      1548
3120 input "[2right]Von Zeile[2spaces]:       3925
    [2left]";s$:s1=int ( val (s$))-1
3130 if s1<0 or s1>zg then print "[up]";:     2299
    goto 3120
3140 input "[down,2right]Bis Zeile[2spaces]   3765
    :[2left]";s$:s2=int ( val (s$))-1
3150 if s2<0 or s2<s1 or s2>zg then print "    3070
    [up]";:goto 3140
3170 rem zeilenweise
3205 print#4,"Zeile";s1+1;" bis Zeile";s2+    3951
    1:print#4,""
3210 for i=s1 to s2                           1035
3220 gosub 11000                               573
```


WWW.HOMECOMPUTERWORLD.COM

```
3240 print#4,t$(i)                                          895
3245 gosub 11100                                            581
3250 next i                                                 276
3300 close 4:close 6                                        600
3310 trap 980:goto 3000                                    1266
3400 close 4:close 6:print "[clr,2down,                    4935
     8right,flashon]DIVICE NOT PRESENT !
     [flashoff]"
3410 print "[4down,7right]Weiter mit                      3789
      'space'":trap 980
3420 get key y$:if y$<>" " then 3420                       1987
3430 print "[clr]":goto 3000                               900
3500 rem insert line
3505 print " [clr]"t$(z)                                    878
3520 for zi=zg to z step -1                                1430
3525 if zi=0 then 3540                                     1258
3530 t$(zi)=t$(zi-1)                                       1518
3540 next zi                                                347
3600 t$(z)=" "                                              944
3610 gosub 13400:rem pos 5                                  755
3620 print chr$ (27)"w"                                    1094
3630 if zm<zg then zm=zm+1:goto 1000                       2605
3700 rem seitenweise
3710 gosub 13100:print "[clr]";:rem pos 2                   1524
3720 input "[2down,2right]Leerzeilen oben :                2762
     ";l$
3730 lo=val (l$)                                            906
3740 input "[down,2right]Leerzeilen unten:"                3094
     ;l$
3750 lu=val (l$)                                            918
3760 input "[down,2right]Zeilen pro seite:"                2640
     ;l$
3770 zl=val (l$)                                            914
3780 input "[down,2right]Von Zeile[7spaces]                2620
     :";s$
3790 s1=val (s$):if s1-1<0 or s1-1>zg then                 4101
     print "[up]";:goto 3780
3800 input "[down,2right]Bis Zeile[7spaces]                2491
     :";s$
3810 s2=val (s$):if s2-1<0 or s2<s1 or s2>                 3989
     zg then print "[up]";:goto 3800
3820 dz=(s2-s1)+1:sd=(dz/zl):if sd<>int                    4644
     (sd) then sd=int (sd)
3835 zz=0                                                   601
3840 for j=0 to lo:print#4,"":next j                       1810
3850 for i=s1-1 to s2-1                                    1419
3855 gosub 11000                                            573
3860 print#4,t$(i)                                          895
3862 gosub 11100                                            581
3865 zz=zz+1                                               1008
3870 if zz=zl then gosub 3900                              1792
3880 next i                                                 276
3890 close 4:close 6                                        600
3899 goto 3000                                              563
3900 for j=0 to lu:print#4,"":next j                       1684
3910 print#4,"-[38spaces]-"                                2070
3920 for j=0 to lo:print#4,"":next j                       1810
3930 zz=0:return                                            973
4000 rem save/load
4001 trap 4000                                              643
4010 gosub 13100:rem pos 2                                  731
4020 print "[clr,down,10right]Save/Load-                  3918
     Menue"
4030 print "[10right,15shift *]"                          2620
4040 print "[down,3right]Save[21spaces]:                 2512
     [2spaces]1"
4050 print "[down,3right]Resave[19spaces]:               2445
     [2spaces]2"
4060 print "[down,3right]Laden[20spaces]:                2238
     [2spaces]3"
4070 print "[down,3right]Zuladen[18spaces]:              2610
     [2spaces]4"
4090 print "[down,3right]Textmodus                      2853
     [16spaces]:[2spaces]5"
4100 print "[down,3right]Neuanfang                      2854
     [16spaces]:[2spaces]6"
4110 get key x$:x=val (x$):x=int ( abs (x))             2789
4120 if x<1 or x>6 then 4110                               1422
4125 if x<5 then gosub 18000                               1429
4130 on x goto 4200,4150,4300,4400,990,                    2930
     16000
4140 rem resave
4200 rem save
4220 print#1,zm                                             848
4230 for x=0 to zm-1                                       1130
4235 if t$(x)="" then t$(x)="*"                            1660
4240 print#1,t$(x)                                          829
4245 if t$(x)="*" then t$(x)=""                            1928
4250 next x                                                 306
4260 print#1:close 1                                        506
4270 x$="":trap 980:goto 4000                              1454
4299 :close 1:goto 4000                                     898
4300 rem load
4320 input#1,zm                                             828
4330 for x=0 to zm-1                                       1130
4340 input#1,t$(x)                                          809
4345 if t$(x)="*" then t$(x)=""                            1928
4350 next x                                                 306
4360 close 1                                                258
4370 x$="":goto 4000                                        944
4400 rem zuladen
4420 input#1,zu                                             701
4430 if zm+zu>zg then 4470                                 1585
4440 for x=zm to zm+zu-1                                   1940
4450 input#1,t$(x)                                          809
4455 if t$(x)="*" then t$(x)=""                            1928
4460 next x                                                 306
4465 zm=zm+zu                                              1273
4470 close 1                                                258
4480 x$="":goto 4000                                        944
4500 rem delete line
4505 print " [clr]"t$(z)                                    878
4510 for zd=z to zg-1                                      1370
4520 t$(zd)=t$(zd+1)                                       1382
4530 next zd                                                327
4540 t$(zg)=" "                                             968
4600 gosub 13400:rem pos 5                                  755
4610 print chr$ (27)"v"                                     966
4630 gosub 13200:rem pos 3                                  739
4640 if z+9<zg then print t$(z+9)                          2068
4650 if zm>0 then zm=zm-1:goto 1000                        2256
5000 rem control
5005 b$=left$ (a$,1)                                        742
5010 if b$="[cbm q]" then 990:rem f1 tx anf                1634
     .
5020 if b$="[cbm w]" then 2000:rem f4 text                 1693
     ende
5030 if b$="[cbm e]" then 2200:rem f2 line                 1633
     up
5040 if b$="[cbm r]" then 2300:rem f5 line                 1667
     dwn
5050 if b$="[cbm t]" then 3000:rem f3 print                1692
5060 if b$="[cbm y]" then 4000:rem f6 save/                1823
     load
5070 if b$="[cbm d]" then 3500:rem f7 inst                 1735
     line
5080 if b$="[cbm f]" then 4500:rem f8 dele                 1706
     line
5090 goto 1060                                              607
6000 rem ende ?
6010 if z=<zg-1 then return                                1657
6060 char 1,8,24,"":print "[flashon,rvs]                   4102
      Der Speicher ist voll ! [flashoff,
     off,home]"
6065 for d=1 to 1000:next d:char 1,8,24,"                  4153
     [rvs,shift *,cbm e,4shift *,cbm e,
     4shift *,cbm e,4shift *,cbm e,
     4shift *,cbm e,3shift *]"
6080 if z>zg then z=z-1                                    1699
6085 gosub 13100                                            589
6090 return                                                 142
7000 rem kopf
7030 char 1,15,4,"[rvs]":print using "####"                 2700
     ;z+1
7040 char 1,25,4,"[rvs]":print using "####"                 2595
     ;zg
7050 return                                                 142
9000 rem start neue seite / up
9010 gosub 13400:rem pos 5                                  755
9020 print "[clr]";                                        495
9030 for z=0 to 9                                           753
9040 print t$(z)                                            732
9050 next z                                                 310
9060 z=0                                                    383
9070 return                                                 142
10000 rem kopf-maske
10010 print "[2home,clr,rvs,2spaces]F1=Text                4803
      Anf[shift -] F4=Text Ende [shift -]
      F3=Print[2spaces]";
10020 print "[2spaces]F2=Line up [shift -]                4731
      F5=Line down [shift -] F6=Save
     [3spaces]";
10030 print "[2spaces]F7=Insert[2spaces,                  4748
     shift -] F8=Delete[4spaces,shift -,
     5spaces]/Load ";
10035 print "[13shift *,cbm e,14shift *,                  3304
```


WWW.HOMECOMPUTERWORLD.COM

```
      cbm e,11shift *]";
10040 print "[10spaces]Line[8spaces]of          2496
      [16spaces]";
10050 print "[4shift *,cbm r,4shift *,cbm r,    3372
      4shift *,cbm r,4shift *,cbm r,
      4shift *,cbm r,4shift *,cbm r,
      4shift *,cbm r,4shift *,cbm r,off]"
10060 char 1,0,24,"[rvs,4shift *,cbm e,          4130
      4shift *,cbm e,4shift *,cbm e,
      4shift *,cbm e,4shift *,cbm e,
      4shift *,cbm e,4shift *,cbm e,
      4shift *,off]"
10070 poke 4071,241                               999
10080 return                                      142
11000 b1$="@":b2$=",":b3$="[shift £]":b4$=":      4326
      ":b5$="[cbm £]":b6$=";"
11005 for q=1 to len (t$(i))                      1248
11010 if mid$ (t$(i),q,1)=b1$ then mid$           3234
      (t$(i),q,1)=b2$
11020 if mid$ (t$(i),q,1)=b3$ then mid$           3237
      (t$(i),q,1)=b4$
11030 if mid$ (t$(i),q,1)=b5$ then mid$           3240
      (t$(i),q,1)=b6$
11040 next q                                      292
11050 return                                      142
11100 b1$=",":b2$="@":b3$=":":b4$="[shift £]      4326
      ":b5$=";":b6$="[cbm £]"
11110 goto 11005                                  729
13000 rem pos 1
13010 char 1,0,14,"":print chr$ (27)"t            8214
      [39right]";chr$ (27)"b[home]";
13030 return                                      142
13100 rem pos 2
13110 print "[2home,6down]" chr$ (27)"t";        2061
13120 print "[home,39right,17down]" chr$         7217
      (27)"b[home]";
13130 return                                      142
13200 rem pos 3
13210 char 1,0,23,"":print chr$ (27)"t";          2465
13220 print "[home,39right]" chr$ (27)"b          6470
      [home]";
13230 return                                      142
13300 rem pos 4
13310 print "[2home,6down]" chr$ (27)"t";        2061
13320 print "[home,39right]" chr$ (27)"b         6470
      [home]";
13330 return                                      142
13400 rem pos 5
13410 char 1,0,14,"":print chr$ (27)"t";          2242
13420 print "[home,39right,9down]" chr$          6707
      (27)"b[home]";
13430 return                                      142
13500 rem pos 6
13510 char 1,0,6,"":print chr$ (27)"t";           2469
13520 print "[home,39right,10down]" chr$         6740
      (27)"b";
13530 return                                      142
15000 rem **** sicherheitsabfrage 1 ****
15010 input "[clr,8down,7right]Sind Sie           4641
       sicher (j/n)";f$
15020 if f$<>"n" and f$<>"j" then print "         3070
      [3up]";:goto 15010
15030 if f$="n" then 980                          1515
15035 close 15                                    470
15040 sys 65526                                   880
16000 rem **** sicherheitsabfrage 2 ****
16010 input "[clr,7down,10right]Sind Sie          3689
       sicher";f$
16020 if f$<>"j" and f$<>"n" then print "         3078
      [3up]";:goto 16010
16030 if f$="j" then run                          1164
16040 goto 980                                    604
18000 rem *** save/load-name ***
18010 :
18020 print "[clr,5down,3right]";:input           3259
      "Dokument-Name";na$
18030 if na$="" then 18020                        1292
18040 return                                      142
19000 rem *** disk-fehler-meldung ******
19010 :
19020 if ds=0 then return                         911
19030 print "[clr,5right,5down]";ds$             1851
19040 print "[6down]";spc(10);"Weiter mit        3531
       'SPACE'"
19050 get y$                                      483
19060 if y$<>" " then 19050                       1448
19070 return                                      142
20000 rem **** direkt-programmierung ***
20010 :
20020 print "[2home,clr,3down]";spc(17);"at-      2270
      text"
20030 print "[3down,6right]bitte peripherie      3803
       auswaehlen."
20040 print "[3down,8right,rvs]d[off]iskette     4029
       oder [rvs]k[off]assette";:input p$
20050 if p$<>"d" and p$<>"k" then print "         2777
      [4up]";:goto 20040
20060 if p$="d" then poke 210,70                  2121
20070 if p$="k" then poke 210,20                  1930
20080 print "[3down,8right,rvs]d[off]rucker       3751
      [2spaces]oder [rvs]p[off]lotter ";:
      input p$
20090 if p$<>"d" and p$<>"p" then print "         2588
      [4up]";:goto 20080
20095 poke 65286,0                                980
20100 if p$="d" then poke 209,80                  1966
20110 if p$="p" then poke 209,0                   1827
20120 restore 21100+peek (210)                    1460
20130 read a                                      265
20135 print "[clr]";                             495
20140 gosub 20500                                 611
20150 restore 21000+peek (209)                    1520
20160 read a                                      265
20165 print "[clr]";                             495
20170 gosub 20500                                 611
20175 print "[clr]2"                             478
20180 if peek (210)=70 then print                 2128
      "delete20000-"
20190 if peek (210)=20 then print                 2259
      "delete19000-"
20196 print "[2down]run"                         796
20200 poke 1319,19:poke 1320,13:poke 1321,        4724
      13:poke 1322,13:poke 239,4:end
20500 rem *** dir.progr.eing.loop ******
20510 :
20530 for i=1 to a                                728
20540 read a$                                     409
20550 print a$                                    427
20560 next i                                      276
20590 poke 1319,19                                1102
20600 for i=1 to a+1:poke 1319+i,13:next i        2532
20620 poke 239,a+1:end                            1271
21000 rem *** direkt-progr.datas pl. ***
21010 data 7                                      241
21020 data "3190 open6,6,6:trap3400"              2319
21030 data "3195 print#6,1"                       1793
21040 data "3200 open4,6"                         1696
21050 data "3825 open6,6,6:trap3400"              2423
21060 data "3830 print#6,1"                       1593
21070 data "3832 open4,6"                         1856
21072 data "goto 20175"                           1254
21080 rem *** dir.-progr.datas dr. *****
21090 data 3                                      233
21100 data "3190 open4,4,7:trap3400"              2367
21110 data "3825 open4,4,7:trap3400"              2471
21112 data "goto 20175"                           1254
21120 rem *** dir.-progr.datas kass. ***
21130 data 5                                      237
21135 data "4150 char1,8,22,vn$:fori=1to500:      4473
      nexti:goto4000"
21140 data "4210 open1,1,2,na$:trap4299"          3079
21150 data "4310 open1,1,0,na$:trap4299"          3086
21160 data "4410 open1,1,0,na$:trap4299"          3094
21164 data "goto 20150"                           1206
21170 rem *** dir.-progr.datas disk ****
21180 data 7                                      241
21185 data "8 open15,8,15"                        1648
21187 data "4150 open1,8,8,re$+na$+sv$:           5398
      gosub19000:ifds<>0then4299"
21188 data "4160 goto4220"                        1423
21190 data "4210 open1,8,8,na$+sv$:               5396
      gosub19000:ifds<>0then4299"
21200 data "4310 open1,8,8,na$+ld$:               5303
      gosub19000:ifds<>0then4299"
21210 data "4410 open1,8,8,na$+ld$:               5311
      gosub19000:ifds<>0then4299"
21230 data "goto 20150"                           1206
```

Flexibel auf Peripheriegeräte einzustellen: „AT-Text"

WWW.HOMECOMPUTERWORLD.COM

Guten Appetit! Commodore „verdaut“ Karteikarten

System: C16/116/Plus4, Datasette oder Floppy, Drucker
Typ: Anwendung
Sprache: Basic

„Computer sind hervorragend zum Verwalten großer Datenmengen geeignet.“ Daß diese Behauptung sogar für die kleinen C16/116 gilt, beweist „Uni-Kartei“.

Bis zu fünf Datenfelder bearbeitet dieses elektronische Karteikasten-Programm. Um die eingegebenen Daten zu speichern, benötigt man entweder eine Datasette oder eine Diskettenstation.
Das abgedruckte Listing ist die Floppy-/Drucker-Version dieses Programmes. Andere Peripherie-Kombinationen benötigen Programmänderungen, die in der Anleitung beschrieben sind. Schon beim Abtippen muß der Benutzer wissen, welchem Zweck die Kartei dient. Die Bezeichnungen der fünf Felder der Kartei sind in den Zeilen:
Feld 1 : 100,1340,12170
Feld 2 : 110,1345,12175
Feld 3 : 120,1350,12180
Feld 4 : 130,1355,12200
Feld 5 : 140,1360,12205
In einzutragenden Zeilen 100 bis 140 darf jeder Feldname maximal neun Zeichen enthalten.
1. DATASETTE: Änderungen im Programm:
5, 3532, 3545, 3615, 3629, 4042, 4055, 4067, 4125, 4139, 8000 bis 8160 und 20 000 bis 20 050 entfallen ersatzlos.
3525 ('DIRECTORY = D' entfällt.)
3540 OPEN 2,1,0,L$
4035 ('DIRECTORY = D' entfällt.)
4050 OPEN 2,1,2,L$
2. PLOTTER 1520: Änderungen im Programm
1320 OPEN 4,6
12120 OPEN 4,6
12130 (CHR$(14) entfällt eventuell)
12140 (CHR$(15) entfällt eventuell)
12160 (CHR$(18) entfällt eventuell)
3. FLOPPY 1541
In den Zeilen 20 000 bis 20 050 ist der Error-Check anzupassen.

Die verwendeten Steuercodes des SP 180:
CHR$(14) = Doppelte Breite an
CHR$(15) = Doppelte Breite aus
CHR$(18) = Negativdruck
Nach dem Start erfragt das Programm den Kartei-Hauptnamen. Die Dimensionierung der Feld-Variablen ist auf maximal 29 Zeichen fassende fünf Felder ausgelegt. Im Bedarfsfall sind Änderungen in Zeile 111 möglich. Die Formel:

Die verwendeten Variablen:

Variable	Zweck
T$(X)	= Feld 1
N$(X)	= Feld 2
Y$(X)	= Feld 3
D$(X)	= Feld 4
Z$(X)	= Feld 5
U$	= Haupt-Karteiname
KN$	= Unter-Karteiname
Z$	= Trennstrich für Maske
A$	= Mehrfach verwendete Hilfsvariable
L$	= Zwischenspeicher für KN$
F$	= Hilfsvariable für FINDEN und ORDNEN
S$	= Suchstring
G$	= Hilfsvariable bei ORDNEN
H$	= Hilfsvariable bei ORDNEN
S%(X)	= Hilfsvariable bei ORDNEN
D1%	= Maximal mögliche Datensätze
Z	= Aktuelle Datensatzanzahl
I	= Nummer des zur Zeit dargestellten Datensatzes
Y	= Vertikale Position des Pseudocursors bei FINDEN und ORDNEN
P	= Hilfsvariable für ORDNEN und Speicher für ORDNEN-Nummer bei SAVE/LOAD
R	= Hilfsvariable bei EINGABE und ORDNEN
X1	= Pokewert für ORDNEN-Nummer Anzeige
X	= Schleifenvariable bei ORDNEN/FINDEN-Vorauswahl
M	= Hilfsvariable bei FINDEN
S	= FINDEN-Flag
ZZ	= Zwischenspeicher für 'Z' bei KORREKTUR
L	= Hilfsvariable bei ORDNEN
T	= Hilfsvariable bei ORNDEN
J	= Hilfsvariable bei ORDNEN
D	= Hilfsvariable bei ORDNEN
DL	= Schleifenvariable bei AUSTRAG
S1	= Auszugdruck 'von Seite'
S2	= Auszugdruck 'bis Seite'

WWW.HOMECOMPUTERWORLD.COM

Erklärungen zum Listing

Zeile		Zweck
5 —	51	Dimensionierung
55 —	159	Maskenaufbau
200 —	300	Funktionsabfrage und Info-Eintrag
1000 —	1230	EINGABE von Daten
1300 —	1375	Pseudo-HARDCOPY
1500 —	1650	ORDNEN/FINDEN Vorauswahl
1700 —	1810	FINDEN von Daten
2000 —	2060	KORREKTUR von Daten
2500 —	2520	VORblättern
3000 —	3020	ZURÜCKblättern
3500 —	3650	LOAD-Routine
4000 —	4160	SAVE-Routine
4500 —	4520	Datei-RESET
8000 —	8050	Directory
8100 —	8160	Directory Ausdrucken
9000 —	9060	SUB Daten in Maske eintragen
10000 —	10200	SUB Daten-Input/b. Eingabe und Korrektur
10300 —	10330	SUB Maskeneinträge löschen
10400 —	10410	SUB Fenster für Maske
11000 —	11840	ORDNEN-Routine
12000 —	12240	DRUCK von Daten
13000 —	13140	AUSTRAGEN (aktuellen, sichtbaren Datensatz löschen)
20000 —	20050	Disk Error Check

Im Inneren von „Uni-Kartei"

(Freier Restspeicher — 400 Bytes) durch (durchschnittliche Feldnutzung pro Feld in Zeichen mal Feldanzahl). Wenn die Dimensionierung abgeschlossen ist, erscheint die Bildschirmmaske. Im oberen Teil stehen erreichbare Funktionen und der Kartei-Hauptname. Der erste Buchstabe einer jeden Funktion ist negativ dargestellt. Ein Druck auf die entsprechende Taste löst die Funktion aus.

1. LOAD

Lädt bestehende Karteien mit allen Parametern.

2. SAVE

Speichert aktuellen Datensatz mit Unter- und Hauptkartei-Namen.

3. EINGABE

Daten in die Maske eintragen. Systemeigene Textverarbeitungsfunktionen (ESC-Sequenzen) bleiben erhalten. Leerstrings, Kommata, Semikolons, Doppelpunkte und CLR sind zu vermeiden !

4. KORREKTUR

Zum Korrigieren mit 'RETURN' das zu ändernde Feld anwählen und dieses editieren.

5. FINDEN

Zum Suchen eines Datensatzes mit SPACE-Taste den Cursor auf das Feld stellen, in dessen Bereich gesucht wird, 'RETURN' drücken, Suchbegriff eingeben, weiteres 'RETURN' aktiviert den Suchvorgang.

6. ORDNEN

Wählvorgang wie bei 'Finden' eingeben.

7. RESET

Setzt die Kartei auf Seite 1 zurück.

8. VOR

Blättert eine Seite vor.

9. ZURÜCK

Blättert eine Seite zurück.

10. AUSTRAGEN

Den aktuellen sichtbaren Datensatz löschen

11. HARDCOPY

Sichtbaren Datensatz drucken

12. DRUCK

Druckt wahlweise die komplette Kartei oder einen Auszug davon.

1. Nummer der aktuellen Seite
2. Maximale Seitenzahl
3. Anzahl der belegten Seiten
4. Freie Bytes
5. Unterkartei-Name
6. Programmabbruch-Möglichkeit
7. Neustart-Möglichkeit

Ist die Speicherkapazität erschöpft oder die Anzahl der dimensionierten Datensätze erreicht, erfolgt eine Warnung. Neustart nach 'Absturz' ohne Datenverlust mit 'GOTO 51'.

```
  5 open 15,8,15                                          1060
 10 z=1:i=z-1:print chr$ (27)"m"                          2399
 15 d1%=( fre (0)-400)/(18*5)                             1936
 20 dim t$(d1%),n$(d1%),s%(d1%)                           2324
 30 dim y$(d1%),d$(d1%),z$(d1%)                           2324
 37 for y=1 to 8:key y,"":next y                          1780
 41 kn$="[shift *]?[shift *]":print "                     1546
    [2home,clr]"
 42 char 1,8,22,"nur 'return' = kein name"                3682
 43 input "[home,3down,2right]kartei                      2833
     hauptname ";u$
 44 if u$="" then u$="ohne namen"                         2531
 50 z$="[off,40shift *]"                                  3040
 51 gosub 55:goto 160                                     1023
 55 poke 65286,0                                           980
 60 print "[2home,clr]"                                   574
 72 print "[4spaces,rvs]e[off]ingabe                      3994
    [6spaces,rvs]v[off]or[9spaces,rvs]l
    [off]oad/[rvs]s[off]ave"
 74 print "[4spaces,rvs]f[off]inden                       3744
    [7spaces,rvs]z[off]urueck[5spaces,rvs]
    a[off]ustragen"
 76 print "[4spaces,rvs]k[off]orrektur                    3909
    [4spaces,rvs]r[off]eset[7spaces,rvs]o
    [off]rdnen/0"
 80 print "[4spaces,rvs]h[off]ardcopy                     3404
    [5spaces,rvs]m[off]enue[7spaces,rvs]d
    [off]ruck"
 90 print z$;                                              694
100 print "[rvs]feld 1 :"                                 959
105 print z$;                                              694
110 print "[rvs]feld 2 :"                                 960
115 print z$;                                              694
120 print "[rvs]feld 3 :"                                 961
125 print z$;                                              694
130 print "[rvs]feld 4 :"                                 962
135 print z$;                                              694
140 print "[rvs]feld 5 :"                                 963
145 print z$;                                              694
150 poke 65286,27:return                                  1210
160 :
200 poke 2225,peek (2225)+128                             1915
201 char 1,19-( int ( len (u$)/2)),0,"":                  3754
    print "[rvs]";u$;"[off]"
202 char 1,0,16,""                                        1124
204 print chr$ (27)"t":print "[clr]"                      1325
210 get a$:trap 202                                        876
212 print "[home,down,right]aktuelle seite                4183
     :";i;" der datei [rvs]";kn$
215 print "[down,11right]";fre (x)-400;"                  4016
     byte frei"
217 print "[down,7right]";z-1;"seiten                     4123
     belegt von";d1%
218 print "[down,right](ctrl+c= :neustart                 4271
```


WWW.HOMECOMPUTERWORLD.COM

```
      / ctrl+shift :stop)"
  220 if a$="e" then 1000                          1577
  225 if peek (1347)=6 then run                    1635
  227 if peek (1347)=5 then print chr$             3502
      (27)"l[2home,clr]":close 15:end
  230 if a$="f" or a$="o" then 1500                2414
  235 if a$="a" then 13000                         1458
  240 if a$="k" then 2000                          1515
  250 if a$="v" then 2500                          1622
  260 if a$="z" then 3000                          1486
  270 if a$="l" then 3500                          1558
  280 if a$="s" then 4000                          1518
  290 if a$="r" then 4500                          1496
  295 if a$="d" then 12000                         1552
  297 if a$="h" then 1300                          1424
  300 goto 210                                      562
 1000 if fre (x)-400<=0 or z=d1%+1 then            5779
      print "[down,8right]kartei ist voll !"
      :goto 210
 1005 poke 2225,peek (2225)-128                    1947
 1010 poke 2092,peek (2092)+128                    1915
 1020 gosub 10000                                   569
 1220 poke 2092,peek (2092)-128                    1947
 1230 z=z+1:i=z-1:goto 200                         1482
 1300 rem
 1302 poke 2212,peek (2212)+128                    1831
 1305 poke 2225,peek (2225)-128                    1947
 1310 trap 1365                                     777
 1315 :
 1320 open 4,4                                      600
 1340 print#4,"feld 1 : ";t$(i)                    1781
 1345 print#4,"feld 2 : ";n$(i)                    1780
 1350 print#4,"feld 3 : ";y$(i)                    1915
 1355 print#4,"feld 4 : ";d$(i)                    1779
 1360 print#4,"feld 5 : ";z$(i)                    1792
 1362 print#4                                       256
 1365 close 4                                       264
 1370 poke 2212,peek (2212)-128                    1863
 1375 goto 200                                      558
 1380 :
 1500 poke 2225,peek (2225)-128                    1947
 1510 if a$="f" then poke 2132,peek (2132)+        2992
      128
 1512 if a$="o" then poke 2197,peek (2197)+        3217
      128
 1515 gosub 10300                                   593
 1520 char 1,1,16,""                               1132
 1525 print chr$ (27)"t"                            965
 1540 print "[clr,2down,5right]mit 'space'        3371
       feld waehlen."
 1550 print "[down,4right]mit 'return' feld        3792
       bestimmen."
 1560 y=6                                           406
 1570 get key f$                                    462
 1580 for x=1 to 5                                  884
 1590 char 1,9,y,"[rvs]>[off]"                     1197
 1600 get key f$                                    462
 1605 if f$<>chr$ (13) and f$<>chr$ (32)           2697
      then 1600
 1610 if f$=chr$ (13) then char 1,9,y," ":         3605
      x1=x:x=5:next x:goto 1700
 1620 if f$=chr$ (32) then char 1,9,y," ":y=       2811
      y+2
 1630 next x                                        306
 1640 if a$="f" then poke 2132,peek (2132)-        3416
      128:gosub 9000
 1645 if a$="o" then poke 2197,peek (2197)-        3249
      128
 1650 goto 200                                      558
 1700 char 1,1,16,""                               1132
 1701 print chr$ (27)"t":if a$="o" then            2059
      11000
 1702 print "[clr,3down,5right]gesuchter          2700
       begriff ";
 1704 input s$                                      443
 1706 print "[down,3right]weiter ='space'/        3992
       menue ='m'"
 1707 m=0:a$=""                                     954
 1708 for i=0 to z-1                               1141
 1710 on x1 goto 1750,1760,1770,1780,1790          2985
 1720 next i                                        276
 1730 poke 2132,peek (2132)-128                    1881
 1735 poke 2225,peek (2225)+128                    1915
 1740 if m<>0 then i=m:goto 202                    1507
 1745 goto 4500                                     585
 1750 s=instr (t$(i),s$):goto 1800                 1757
 1760 s=instr (n$(i),s$):goto 1800                 1916
 1770 s=instr (y$(i),s$):goto 1800                 1837
 1780 s=instr (d$(i),s$):goto 1800                 1756
 1790 s=instr (z$(i),s$)                           1188
 1800 if s<>0 then gosub 9000:m=i:poke 239,        3837
      0:get key a$
 1805 if a$="m" then i=z-1:goto 1720               2171
 1810 goto 1720                                     603
 2000 poke 2225,peek (2225)-128                    1947
 2010 poke 2172,peek (2172)+128                    1897
 2020 zz=z:z=i                                     1256
 2030 gosub 10005                                   729
 2040 z=zz                                          506
 2050 poke 2172,peek (2172)-128                    1929
 2060 goto 200                                      558
 2500 if i<z-1 then i=i+1                          1953
 2510 gosub 9000                                    579
 2520 goto 202                                      574
 3000 if i>1 then i=i-1                            1388
 3010 gosub 9000                                    579
 3020 goto 202                                      574
 3500 poke 2225,peek (2225)-128                    1947
 3505 poke 2117,peek (2117)+128                    1945
 3508 char 1,0,16,""                               1124
 3510 print chr$ (27)"t":l$=""                     1410
 3520 print "[clr,2down,6right]datei-namen         3318
       eingeben oder"
 3525 print "[down,5right]menue = 'm' /            3820
       directory = 'd'"
 3530 input "[down,3right]eingabe (max. 6          3393
       zeichen) ";l$
 3532 if l$="d" then gosub 8000:goto 3505          2361
 3535 if l$="m" then poke 2117,peek (2117)-        3419
      128:goto 200
 3537 if len (l$)>6 then 3520                      1399
 3538 trap 3545                                     773
 3540 open 2,8,2,"0:"+l$+",s,r"                    2386
 3545 gosub 20000:if ds<>0 then 3630               2116
 3547 kn$=l$                                        852
 3550 input#2,z                                     618
 3555 input#2,kn$                                   858
 3560 for i=1 to z-1                               1149
 3570 input#2,t$(i)                                 871
 3580 input#2,n$(i)                                 823
 3590 input#2,y$(i)                                 911
 3600 input#2,d$(i)                                 743
 3610 input#2,z$(i)                                 919
 3615 gosub 20000                                   571
 3620 next i                                        276
 3625 input#2,u$                                    644
 3627 input#2,p                                     538
 3629 gosub 20000                                   571
 3630 close 2                                       260
 3640 poke 2117,peek (2117)-128                    1977
 3645 char 1,3,0,"[32spaces]"                      1933
 3650 i=1:x1=p:gosub 9000:goto 11835               2358
 4000 poke 2225,peek (2225)-128                    1947
 4010 poke 2122,peek (2122)+128                    1837
 4020 char 1,0,16,""                               1124
 4025 print chr$ (27)"t":l$=""                     1410
 4030 print "[clr,2down,6right]datei-namen         3318
       eingeben oder"
 4035 print "[down,5right]menue = 'm' /            3820
       directory = 'd'"
 4040 input "[down,3right]eingabe (max. 6          3393
       zeichen) ";l$
 4042 if l$="d" then gosub 8000:goto 4010          2211
 4045 if l$="m" then poke 2122,peek (2122)-        3311
      128:goto 200
 4047 if len (l$)>6 then l$="":goto 4030           2303
 4050 open 2,8,2,"@0:"+l$+",s,w"                   2211
 4055 gosub 20000:if ds<>0 then goto 4140          2207
 4057 kn$=l$                                        852
 4060 print#2,z                                     638
 4065 print#2,kn$                                   878
 4067 gosub 20000                                   571
 4070 for i=1 to z-1                               1149
 4080 print#2,t$(i)                                 891
 4090 print#2,n$(i)                                 843
 4100 print#2,y$(i)                                 931
 4110 print#2,d$(i)                                 763
 4120 print#2,z$(i)                                 939
 4125 gosub 20000                                   571
 4130 next i                                        276
 4135 print#2,u$                                    664
 4137 print#2,p                                     558
 4139 gosub 20000                                   571
 4140 close 2                                       260
 4150 poke 2122,peek (2122)-128                    1869
 4155 char 1,3,0,"[32spaces]"                      1933
 4160 i=1:x1=p:gosub 9000:goto 11835               2358
 4500 i=1                                           370
```


WWW.HOMECOMPUTERWORLD.COM

```
 4510 gosub 9000                                          579
 4520 goto 202                                            574
 8000 rem directory
 8010 print "[2home,clr]"                                 574
 8020 directory                                           238
 8025 gosub 20000                                         571
 8030 char 1,30,7,"[rvs] weiter [off]"                   1943
 8032 char 1,30,9,"[rvs,2spaces]mit[3spaces,             1563
      off]"
 8034 char 1,30,11,"[rvs] space[2spaces,off]             1947
      "
 8036 char 1,30,13,"[rvs] print[2spaces,off]             1859
      "
 8037 char 1,30,15,"[rvs,2spaces]mit                     1903
      [3spaces,off]"
 8038 char 1,30,17,"[rvs,2spaces]'p'                     2006
      [3spaces,off]"
 8040 get key a$:if a$<>" " and a$<>"p" then             2616
      8040
 8042 if a$="p" then 8100                                1428
 8045 gosub 55:gosub 9000                                1007
 8050 return                                              142
 8100 print "[2home,clr]"                                 574
 8105 trap 8030                                           675
 8110 open 4,4                                            600
 8120 cmd 4                                               261
 8130 directory                                           238
 8135 gosub 20000                                         571
 8140 print#4                                             256
 8150 close 4                                             264
 8160 goto 8030                                           597
 9000 gosub 10300                                         593
 9010 char 1,9,6,t$(i)                                   1370
 9020 char 1,9,8,n$(i)                                   1176
 9030 char 1,9,10,y$(i)                                  1506
 9040 char 1,9,12,d$(i)                                  1613
 9050 char 1,9,14,z$(i)                                  1508
 9060 return                                              142
10000 gosub 10300                                         593
10005 char 1,9,6,"":gosub 10400                          1633
10010 poke 19,64                                          807
10020 input t$(z)                                         712
10030 poke 19,0                                           577
10040 char 1,9,8,"":gosub 10400                          1442
10050 poke 19,64                                          807
10060 input n$(z)                                         700
10070 poke 19,0                                           577
10080 char 1,9,10,"":gosub 10400                         1517
10090 poke 19,64                                          807
10100 input y$(z)                                         722
10110 poke 19,0                                           577
10120 char 1,9,12,"":gosub 10400                         1645
10130 poke 19,64                                          807
10140 input d$(z)                                         680
10150 poke 19,0                                           577
10160 char 1,9,14,"":gosub 10400                         1518
10170 poke 19,64                                          807
10180 input z$(z)                                         724
10190 poke 19,0                                           577
10200 return                                              142
10300 for r=6 to 14 step 2                               1059
10310 char 1,9,r,"[30spaces]"                            2127
10320 next r                                              294
10330 return                                              142
10400 print chr$ (27)"t[29right]";                       4909
10405 print chr$ (27)"b[29left]";                        5808
10410 return                                              142
11000 p=1:s%(1)=1:s%(2)=z-1                              2123
11005 print "[clr,3down,8right,flashon]es                3263
       wird sortiert ![flashoff]"
11010 l=s%(p):r=s%(p+1):p=p-2                            1934
11020 t=1:j=r:d=(l+r)/2                                  1879
11030 on x1 goto 11040,11070,11100,11130,                2908
      11160
11040 g$=t$(d)                                            743
11050 if t$(t)>=g$ then 11210                            1446
11060 goto 11190                                          721
11070 g$=n$(d)                                            695
11080 if n$(t)>=g$ then 11230                            1562
11090 goto 11190                                          721
11100 g$=y$(d)                                            783
11110 if y$(t)>=g$ then 11250                            1457
11120 goto 11190                                          721
11130 g$=d$(d)                                            615
11140 if d$(t)>=g$ then 11270                            1543
11150 goto 11190                                          721
11160 g$=z$(d)                                            791
11170 if z$(t)>=g$ then 11290                            1460
11190 t=t+1:goto 11500                                   1167
11210 if t$(j)<=g$ then 11320                            1765
11220 goto 11310                                          609
11230 if n$(j)<=g$ then 11320                            1753
11240 goto 11310                                          609
11250 if y$(j)<=g$ then 11320                            1775
11260 goto 11310                                          609
11270 if d$(j)<=g$ then 11320                            1733
11280 goto 11310                                          609
11290 if z$(j)<=g$ then 11320                            1777
11310 j=j-1:goto 11500                                   1125
11320 if t>j then 11500                                  1121
11400 h$=t$(t):t$(t)=t$(j):t$(j)=h$                      2553
11410 h$=n$(t):n$(t)=n$(j):n$(j)=h$                      2553
11420 h$=y$(t):y$(t)=y$(j):y$(j)=h$                      2638
11430 h$=d$(t):d$(t)=d$(j):d$(j)=h$                      2383
11440 h$=z$(t):z$(t)=z$(j):z$(j)=h$                      2553
11450 t=t+1:j=j-1                                        1033
11500 if t<=j then on x1 goto 11050,11080,               4039
      11110,11140,11170
11510 if t>=r then 11610                                 1507
11600 p=p+2:s%(p)=t:s%(p+1)=r                            2693
11610 r=j:if l<r then 11020                              1662
11620 if p<>-1 then 11010                                1725
11820 i=1:gosub 9000                                      918
11830 poke 2197,peek (2197)-128                          2073
11835 p=x1:poke 3228,x1+48                               1726
11840 goto 200                                           558
12000 rem
12010 poke 2225,peek (2225)-128                          1947
12015 poke 2237,peek (2237)+128                          1975
12020 print "[clr,2down,2right]komplett ?               3676
       (j/n) - menue = 'm'"
12030 get a$:if a$<>"j" and a$<>"n" and a$<>             3534
      "m" then 12030
12035 if a$="m" then 12235                               1697
12040 if a$="j" then s1=1:s2=z-1:goto 12090              3123
12050 input "[clr,2down,2right]von seite";s1             1900
12060 if s1<1 or s1>z-1 then 12050                       2204
12070 input "[2down,2right]bis seite";s2                 2160
12080 if s2<1 or s2>z-1 or s2<s1 then print              3729
      "[clr]";:goto 12070
12090 print "[clr,2down,2right]wenn der                 3559
       drucker bereit ist"
12100 print "[down,2right]eine beliebige                3502
       taste druecken."
12110 get key a$:trap 12230                              1300
12120 open 4,4                                            600
12130 print#4,chr$ (14);u$                               1196
12140 print#4,chr$ (15)                                   887
12150 for x=s1 to s2                                     1065
12160 print#4,chr$ (18);str$ (x)" "                      1613
12170 print#4,"feld 1 : ";t$(x)                          1841
12175 print#4,"feld 2 : ";n$(x)                          1840
12180 print#4,"feld 3 : ";y$(x)                          1975
12200 print#4,"feld 4 : ";d$(x)                          1839
12205 print#4,"feld 5 : ";z$(x)                          1852
12220 next x                                              306
12230 close 4                                             264
12235 poke 2237,peek (2237)-128                          2007
12240 goto 200                                           558
13000 rem
13010 poke 2157,peek (2157)+128                          1993
13020 poke 2225,peek (2225)-128                          1947
13022 input "[clr,2down,2right]sind sie                  2300
       sicher (j/n)";a$
13024 if a$<>"j" and a$<>"n" then 13022                  2396
13026 if a$="n" then 13120                               1639
13030 for dl=i to z-1                                    1171
13040 if dl>100 then dl=z-1:goto 13100                   2386
13050 t$(dl)=t$(dl+1)                                    1310
13060 n$(dl)=n$(dl+1)                                    1301
13070 y$(dl)=y$(dl+1)                                    1445
13080 d$(dl)=d$(dl+1)                                    1286
13090 z$(dl)=z$(dl+1)                                    1319
13100 next dl                                             315
13110 z=z-1                                               408
13120 poke 2157,peek (2157)-128                          2025
13130 poke 2225,peek (2225)+128                          1915
13140 goto 4500                                           585
20000 rem disk error check
20005 if ds=0 then return                                 911
20010 print "[clr,down,right]";ds$:a$=""                1654
20020 print "[down,right]weiter = return"               2046
20030 get key a$:if a$<>chr$ (13) then 20030             2050
20050 return                                              142
```

Ordnung halten mit „Uni-Kartei"

WWW.HOMECOMPUTERWORLD.COM

Krater sprengen ist nicht schwer . . .

System: C64/Joystick
Typ: Spiel
Sprache: Basic

Gut getarnt durch einen Höhlengang liegt die Raumbasis des Planetoiden Klee IV. Polizeieinheiten des Imperiums zwingen zu einer schnellen Landung auf dem zerklüfteten Himmelskörper.

Mit Joystick steuert man den Raumer durch die engen, mit Minen versehenen Tunnel. Die Berührung mit etwas anderem als der Landefläche führt zum sicheren Tod. Ein Level besteht aus drei Tunneln, wobei die Schwerkraft des Planetoiden mit jedem Durchgang zunimmt. Der Clou am Spiel: Jeder kann weitere Tunnel konstruieren und dem Programm anfügen!

1.) In Zeile 20200 steht die Anweisung „ON L GOSUB 50000,51000,52000". Von jedem weiteren Tunnel muß hier die erste Basic-Adresse angefügt werden. Beispiel: Tunnel 4 beginnt ab Zeile 52500. In 20200 muß: ON L GOSUB 50000,51000,52000,52500" stehen.

2.) Für jeden weiteren Tunnel muß in Zeile 20219 die Anweisung: „IF L=4 THEN L=1" um Eins erhöht werden. Das Ergebnis unseres Beispiels wäre:
„IF L=5 THEN L=1".

3.) Die Tasten „ #$%&'()+—*" entsprechen den verschiedenen „Bauteilen" des Tunnels. Bricht man das Programm mit RUN/STOP ab und gibt diese Tasten ein, erscheinen am Bildschirm die entsprechenden Grafiken. In PRINT-Befehlen bauen diese das Spielfeld auf. Die Landebahn muß sich in der untersten der 25 PRINT-Zeilen befinden. Die Variable P gibt den X-Achsen-Standort des Raumschiffes an. Anhand der drei bereits vorhandenen Tunnel ist ersichtlich, wie weitere Landschaften aufgebaut werden.
Zum Abschluß die Variablenliste:

A,B	: Hilfsvarible
G	: Graviation
HX	: MSB-Variable
L	: Level
M	: Horizontale Beschleunigung
N	: Vertikale Beschleunigung
P	: Positionierung des Raumers
Q	: Spritedaten
R	: Runde
SI	: Soundchip
U	: Videochip
X,Y,Z	: Musikdaten
Z	: aktuelles Sprite

Die unterschiedlichen Level befinden sich in den Zeilen:
Level 1 : 50000 bis 50020
Level 2 : 51000 bis 51250
Level 3 : 52000 bis 52250

Viel Spaß beim Spielen und Konstruieren wünscht
Arno Scharl

```
 1 :
 2 rem
 3 rem ***********
 4 rem *** ufo ***
 5 rem ***********
 6 rem
 7 :
 8 rem *** by arno scharl / laerchengasse
    35 / a-2601 sollenau ***
 9 rem *** tel.02628/73273 / (c) 1986 ***
10 v=53248:poke v+21,63:poke 2040,224:    6370
   poke 2041,225:poke 2042,226:poke 2043,
   227
11 poke 2044,228:poke 2045,229:print "    4069
   [clr]":poke 53280,0:poke 53281,0
12 for a=14336 to 14336+62:read q:poke a,  3115
   q:next
13 for a=14400 to 14400+62:read q:poke a,  2720
   q:next
14 for a=14464 to 14464+62:read q:poke a,  3075
   q:next
15 poke v+42,1:poke v+43,1:poke v+44,7:    5903
   poke v+23,7:poke v+29,7:for x=1 to 80
16 poke v+2,160:poke v+3,x:poke v,x:poke   5293
   v+1,160-x:poke v+4,x*3:poke v+5,x:next
17 for a=1 to 25:a$=a$+chr$ (13):next :    5023
   print a$:print tab(15);"[wht]written
    by";chr$ (13)
18 print tab(14);"[yel]arno[2spaces]       4874
   scharl";chr$ (13);chr$ (13);chr$ (13);
   chr$ (13)
19 print tab(13);"[grn]please wait..."     2600
20 for a=14528 to 14528+62:read q:poke a,  2925
   q:next
21 for a=14592 to 14592+62:read q:poke a,  3035
   q:next
22 for a=14656 to 14656+62:read q:poke a,  3140
   q:next
23 poke 52,48:poke 56,48:poke 56334,peek   5596
   (56334) and 254:poke 1,peek (1) and
   251
24 for i=0 to 511:poke i+12288,peek (i+    3619
   53248):next :poke 1,peek (1) or 4
25 poke 56334,peek (56334) or 1            1830
26 poke 53272,( peek (53272) and 240)+12   2958
27 for a=12568 to 12655:read q:poke a,q:   2937
   next
30 print "[up]";tab(13);:for a=1 to 14:    3458
   print " ";:for b=1 to 150:next :next
31 for a=33 to 17 step -1:print tab(a);",  3356
    1985":for b=1 to 50:next
32 print "[up,39spaces,up]"                1525
33 next :print tab(17);", 1985"            1492
34 print "[grn,7up]#[9spaces,pur]+[grn]     1989
   $"
35 print "-([9spaces])-"                    852
36 print "-#[8spaces]'$-"                   1019
37 print "-%[7spaces]$---"                  1196
38 print "-'[yel]******[grn]$----"          2022
39 print "-------------"                    1993
40 for a=150 to 197:for b=1 to 30:next :   3990
   poke v+8,a/4:poke v+9,a:next
41 poke v+8,0:poke v+9,0:poke v+6,197/4:   3988
   poke v+7,197
```


WWW.HOMECOMPUTERWORLD.COM

```
10000 rem *** u ***                                          6670
10001 data 120,0,30,252,0,63,252,0,63,252,0,
      63,252,0,63,252,0,63,252,0,63
10002 data 252,0,63,252,0,63,252,0,63,252,0,                6102
      63,252,0,63,126,0,126,126,0,126
10003 data 127,0,254,63,195,252,63,255,252,                  6651
      31,255,248,15,255,240,3,255,192
10004 data 0,126,0                                           808
10005 rem *** f ***
10006 data 255,255,254,255,255,255,255,255,                  7744
      255,255,255,255,255,255,255
10007 data 255,255,254,252,0,0,252,0,0,252,                  7178
      0,0,252,0,0,255,255,0,255,255,128
10008 data 255,255,128,255,255,0,252,0,0,                    6332
      252,0,0,252,0,0,252,0,0,252,0,0
10009 data 252,0,0,120,0,0                                   1529
10010 rem *** o ***
10011 data 0,126,0,3,255,192,15,255,240,31,                  6698
      255,248,63,255,252,63,195,252
10012 data 127,0,254,126,0,126,126,0,126,                    7037
      252,0,63,252,0,63,252,0,63,126,0,126
10013 data 126,0,126,127,0,254,63,195,252,                   7222
      63,255,252,31,255,248,15,255,240
10014 data 3,255,192,0,126,0                                 1967
10015 rem *** ufo ***
10016 data 0,60,0,0,66,0,0,129,0,3,0,192,28,                 6698
      0,56,224,0,7,144,0,9,142,0,113
10017 data 65,255,130,64,0,2,32,0,4,32,24,4,                 6269
      16,36,8,8,36,16,14,66,112
10018 data 25,129,152,48,255,12,32,0,4,32,0,                 5041
      4,112,0,14,248,0,31
10020 rem *** ufo + antrieb ***
10021 data 0,60,0,0,66,0,0,129,0,3,0,192,28,                 6698
      0,56,224,0,7,144,0,9,142,0,113
10022 data 65,255,130,64,0,2,32,0,4,32,24,4,                 6269
      16,36,8,8,36,16,14,66,112
10023 data 25,129,152,48,255,12,32,0,4,32,                   5768
      34,4,112,136,14,249,36,159
10025 rem *** explosion ***
10026 data 128,0,2,2,4,0,0,0,64,4,0,0,64,8,                  7054
      4,0,128,128,4,0,0,32,18,36,8,152,128
10027 data 0,126,0,2,255,65,8,126,144,65,26,                 6518
      0,8,72,136,2,2,0,0,0,65,16,128,0
10028 data 0,8,0,0,0,32,64,0,0,2,2,1                         2522
10030 rem *** zeichensatz ***
10031 data 128,224,192,240,252,248,250,255                   3654
10032 data 1,3,1,7,15,7,63,255                               2474
10033 data 255,248,192,224,240,224,192,128                   3688
10034 data 255,15,63,7,15,3,7,1                              2266
10035 data 0,0,16,20,28,95,255,255                           2910
10036 data 224,248,240,252,192,240,128,192                   3430
10037 data 3,31,7,15,7,63,15,7                               2591
10038 data 255,170,255,24,24,24,255,153                      3423
10039 data 129,66,60,252,63,60,66,129                        3263
10040 data 126,129,157,161,161,157,129,126                   4099
10041 data 255,255,255,255,255,255,255,255                   4167
20000 :
20005 rem
20010 rem ************************
20015 rem *** ufo - main program ***
20020 rem ************************
20025 rem
20030 :
20035 for a=1 to 15:poke 53287,a:poke 53288,                 5246
      a+1:poke 53289,a-1:for b=1 to 50:next
      :next
20040 for a=15 to 2 step -1:poke 53287,a:                    6012
      poke 53288,a+1:poke 53289,a-1:for b=1
      to 50:next :next
20045 l=1:g=0.1:print "[home,7spaces,wht]...                 4390
       [cyn]press [grn]f1[cyn] to start
       [wht]..."
20050 for a=1 to 55:read y:read x:read z                     2023
20055 si=54272:poke si+24,15:poke si+5,25:                   4001
      poke si+6,68
20060 poke si,x:poke si+1,y:poke si+4,17                     2834
20065 for t=1 to z*1.3:next                                  1380
20070 get a$:if a$=chr$ (133) then poke si+                  3888
      24,0:l=1:goto 20200
20075 poke si+4,0:poke si+5,0:poke si+6,0:                   2218
      next
20080 wait 198,133                                           1254
20090 data 52,39,133,49,58,133,52,39,133,52,                 5891
      39,200,43,219,600,0,0,200
20095 data 58,138,100,58,138,100,58,138,133,                 6880
      65,181,133,69,157,133,69,157,133
20100 data 65,181,133,52,39,133,39,18,400,0,                 5637
      0,200,52,39,100,52,39,100
20105 data 52,39,100,52,39,100,58,138,100,                   6719
      65,181,100,65,181,133,58,138,133
20110 data 65,181,133,58,138,400,0,0,200,49,                 5301
      58,100,49,58,100,49,58,100
20115 data 49,58,100,52,39,100,58,138,100,                   6637
      58,138,133,52,39,133,58,138,133
20120 data 52,39,400,0,0,200,43,219,100,43,                  5630
      219,100,43,219,100,43,219,100
20125 data 43,219,100,52,39,100,39,18,133,                   6862
      34,207,133,32,219,133,32,219,400
20130 data 0,0,200,58,138,100,58,138,100,58,                 6443
      138,133,58,138,133,52,39,133
20135 data 52,39,800                                         807
20150 wait 198,133                                           1254
20200 poke 53280,0:poke 53281,0:on l gosub                   5853
      50000,51000,52000:print "[home,yel]";
      tab(31);"level: ";l
20201 for a=0 to 11:poke v+a,0:next :poke v+                 3992
      29,0:poke v+23,0:x=140:y=50
20202 poke v,p:poke v+1,225:poke v+2,p+30:                   6897
      poke v+3,225:poke v+4,p+60:poke v+5,
      225:a=1:z=6
20210 hx=int (x/256):if hx=1 then poke v+16,                 2666
      56
20211 poke v+z,x-256*hx:poke v+z+1,y:poke                    4119
      53279,0
20212 if x<256 then poke v+16,0                              2127
20213 z=6:if peek (56321)=254 then m=m-1:z=                  5003
      8:poke v+6,0:poke v+7,0
20214 if peek (56321)=253 then m=m+1                         2430
20215 if peek (56321)=251 then n=n-1                         2465
20216 if peek (56321)=247 then n=n+1                         2551
20217 a=peek (53279):if a=15 or a=23 then                    3075
      gosub 40800
20218 if peek (56321)<>254 then poke v+8,0:                  3661
      poke v+9,0
20219 if y>210 and m<4 then l=l+1:if l=4                     3989
      then l=1:g=g+0.1:r=r+1
20220 if y>210 and m<4 goto 20200                            1661
20221 if x<24 then n=0:x=24                                  1792
20222 if x>320 then n=0:x=320                                1988
20223 if y<50 then m=0:y=50                                  1614
20225 m=m+g:x=x+n:y=y+m:goto 20210                           2008
40800 poke v+10,x-256*hx:poke v+11,y:poke v+                 4624
      z,0:poke v+z+1,0
40801 poke si+5,0:poke si+6,0:poke si+4,0                    2059
40802 si=54272:for x=15 to 0 step -0.3:poke                  5920
      si+24,x:poke si+5,15:poke si+6,0:poke
      si+1,40
40803 poke si,200:poke si+4,129:next :poke                   3327
      si+4,0:poke si+5,0
40804 for a=1 to 15:poke 53287,a:poke 53288,                 5246
      a+1:poke 53289,a-1:for b=1 to 50:next
      :next
40805 for a=15 to 2 step -1:poke 53287,a:                    6012
      poke 53288,a+1:poke 53289,a-1:for b=1
      to 50:next :next
40806 print "[home,3down,11right,wht]g a m e                5650
      [3spaces]o v e r[3spaces]!":poke 198,
      0:wait 198,1:poke 198,0:l=1
40807 poke v+21,0:print "[clr,3down]you have                 3552
       reached round ";r+1
40808 print "[down]and were destroyed at                    3004
       level ";l
40809 input "[3down]do you want to try again                 2919
      ";a$
40810 if a$<>"n" then m=0:n=0:poke v+21,63:                  2890
      goto 20200
40811 sys 64738                                              717
50000 print "[clr,grn,4down]"                               828
50001 print "#":print "-([4spaces,red]+[grn,                 2821
      32spaces]$";
50002 print "-#''[10spaces,red]+[grn,                       2304
      23spaces]$-";
50003 print "----#'[31spaces])--";                          2868
50004 print "------([8spaces]'[20spaces]$---                2771
      ";
50005 print "-----%[8spaces]$-#[14spaces,                   3128
      red]+[grn,3spaces])----";
50006 print "-----#''''''$---([17spaces])--                 3634
      --";
50007 print "----------------%[18spaces]$---                3767
      -";
50008 print "---------------%[17spaces]'$---                3884
      --";
50009 print "--------------%[3spaces,red]+                  4294
      [grn,13spaces]$-------";
50010 print "--------------([16spaces])-----                4131
      ---";
```


WWW.HOMECOMPUTERWORLD.COM

```
50011 print "-------------([16spaces])-----  4131
      ---";
50012 print "-------------([16spaces])-----  4131
      ---";
50013 print "-------------#'[16spaces]&----  4094
      ---";
50014 print "---------------#[16spaces])---  4072
      ---";
50015 print "----------------([15spaces]$--  4136
      ----";
50016 print "----------------([14spaces]$--  4224
      -----";
50017 print "----------------#''''[7spaces]  5023
      )#$--------";
50018 print "--------------------[wht]       5418
      *******[grn]$----------";
50019 print "------------------------------  5535
      --------";
50020 p=30:return                             886
51000 print "[grn,clr,3down]"                 633
51010 print "#''''"                           896
51020 print "----#'''''"                     1592
51030 print "----------#''''''[14spaces,pur]  3636
      +[grn,7spaces]$";
51040 print "----------------#'''[17spaces]   4014
      $-";
51050 print "--------------------#'           4341
      [13spaces]'$--";
51060 print "----------------------(          4502
      [11spaces]$----";
51070 print "----------------------(          4679
      [10spaces])-----";
51080 print "----------------------%          4575
      [11spaces]$-----";
51090 print "----------------------%          4742
      [11spaces]$------";
51100 print "----------------%&--%[9spaces]    4631
      ''$-------";
51130 print "---------------%[2spaces]&%       4425
      [9spaces]$----------";
51140 print "-------------%[12spaces]'$----    4508
      -------";
51150 print "---------%&--%[11spaces]'$-----   4446
      --------";
51160 print "--------%[2spaces]&%[11spaces]    4405
      )---------------";
51170 print "-------%[16spaces]$------------   4092
      ---";
51180 print "------%[15spaces]'$------------   4125
      ----";
51190 print "-----%[13spaces]''$------------   4559
      ------";
51200 print "----%[10spaces]'''$------------   4532
      ---------";
51210 print "----([9spaces]$----------------   4701
      ---------";
51220 print "----([8spaces])----------------   4842
      ----------";
51230 print "----#[wht]********[grn]$-------   5560
      --------------------";
51240 print "------------------------------   5535
      --------";
51250 p=160:return                            791
52000 print "[clr,grn,26spaces]'''$---------  3211
      -";
52010 print "#[24spaces]$--------------";      3267
52020 print "-([22spaces]$----------------";   3459
52030 print "-([10spaces]''''''''''$-------   4599
      ---------";
52040 print "-([9spaces]$--------------%&%&-   4398
      ---------";
52050 print "-([8spaces])--------%&%&%&%       4170
      [4spaces]&---------";
52060 print "-([8spaces])--------([11spaces]   3658
      )--------";
52070 print "-([8spaces])-------%[12spaces]    3529
      )--------";
52080 print "-([8spaces])------%[14spaces]&-   3339
      ------";
52090 print "-([9spaces]&--%&%[15spaces])---   3144
      ----";
52100 print "-([10spaces]&%[19spaces]&------   2881
      ";
52110 print "-([32spaces]&-----";             2462
52120 print "-([33spaces]&----";              2397
52130 print "-#[34spaces]&---";               2227
52140 print "--([13spaces,pur]+[grn,9spaces]  2729
      )#[8spaces])---";
52150 print "--#[23spaces]$-([8spaces]&--";   2322
52160 print "---#'[20spaces])--([8spaces])--  3116
      ";
52170 print "-----#[19spaces]$--([9spaces]&-  3005
      ";
52180 print "------#'[16spaces])---(          3138
      [9spaces])-";
52190 print "--------#'[14spaces]$---(         3185
      [10spaces]&";
52200 print "----------#'[11spaces])----(      3605
      [10spaces])";
52210 print "------------#''[8spaces]$----(    4012
      [10spaces]$";
52220 print "---------------#''#'''$-----#     4821
      [9spaces])-";
52230 print "----------------------------#      5377
      [wht]********[grn]$-";
52240 print "------------------------------   5535
      --------";
52250 p=30:return                             886
```

Wenn das Imperium zurückschlägt . . .

Wasser im Commodore 64

System: C64/Joystick
Typ: Spiel
Sprache: Basic/ASM

Schiffe versenken ist ein beliebtes Spiel. SVS ersetzt den Mitspieler und beweist: Der C64 ist nicht auf den Chip gefallen!

Ein starker Gegner stellt sich mit seiner kampfkräftigen Flotte: Es ist der C64. Eine Seeschlacht entbrennt. Sieger ist, wer alle gegnerischen Schiffe versenkt. Nach Start durch RUN fordert SVS auf, einen Joystick in Port 2 zu stecken. Drückt man den Feuerknopf des „Freudenknüppels", erscheint das Eingabemenü: Links das Spielerfeld (SF). Dort setzt man seine Schiffe (1×4, 2×3, 2×2 Felder groß). In dem rechten Feld (CF) sind die Schiffe des Rechners verteilt. Im SF steht oben ein weißes X, der Cursor. Das untere Drittel des Bildschirmes zeigt die zu setzenden Schiffe mit zugehörigem Quadrat (links davon). Mit dem Cursor fährt man das entsprechende Quadrat an und wählt mit Feuer das gewünschte Quadrat.

Treffer – Versenkt

Zur Bestätigung wird der Cursor schwarz. Er gibt die Lage des Schiffsbugs an, falls sich der Cursor bei erneutem Knopfdruck auf einem freien Feld im SF befindet. Wenn alle Schiffe gesetzt sind, ▶

WWW.HOMECOMPUTERWORLD.COM

wird das Programm nach Anwahl von EXIT fortgesetzt. Wenn im CF der Cursor erscheint, steuert man ihn mit dem Joystick. Mit dem Feuerknopf schießt man. Nach Fehlschüssen ist der Computer an der Reihe. Unterhalb der Felder steht die Anzahl der Schüsse und versenkten Schiffe. Unterbricht das Programm, startet man es mit GOTO 80 erneut.

Hat der C64 einen Treffer erzielt, errechnet er weitere Schüsse nach einem Algorithmus.

A. Buschhüter

```
 0 rem ############
 1 rem titelsprite
 2 rem ############
 3 data 16,[2spaces]0,[2spaces]8, 40,        1927
   [2spaces]0, 20, 16,[2spaces]0
 4 data [2spaces]8, 16,[2spaces]0,           2462
   [2spaces]8, 16, 56,[2spaces]8,146
 5 data 40, 73,124, 40, 62, 17,255,          2512
   [2spaces]8
 6 data [2spaces]1,[2spaces]1,[2spaces]0,    2051
   [2spaces]1,[2spaces]1,[2spaces]0, 15,
   255
 7 data 224,[2spaces]4, 16, 64,[2spaces]     2852
   2, 16,128,[2spaces]1
 8 data 17,[2spaces]0,[2spaces]1,147,        2581
   [2spaces]0, 15,147,224
 9 data 16, 84, 16,231, 57,207,[2spaces]     2881
   8,130
10 data 32, 16, 68, 16,224, 56, 15           2380
11 v=53248:poke v+21,6:poke 2042,13:poke     3841
   2041,13:for i=0 to 62
12 read a:poke 832+i,a:next :poke v+40,1:    3412
   poke v+41,1
20 print "[clr,4down,wht]" tab(              3207
   12)"schiffeversenken"
30 poke v+16,2:poke v+2,24:poke v+4,64       3034
40 poke v+3,73:poke v+5,73                   2593
50 print "[2down]" tab(12)"joystick port     2573
    ii"
55 for i=0 to 400:next :poke 646,7-peek      2469
   (646)
60 print "[2down]" tab(14)"press button      2852
   [3up]"
70 p=peek (56320):if (p and 16)<>0 then      2709
   55
80 clr :poke 65,peek (122):poke 66,peek      5274
   (123):poke 53265,239:poke 56333,31
90 rem   ###################
100 rem mp: chr-rom => ram
110 rem ###################
120 data 120,169, 51,133,[2spaces]1,169,     3056
   [2spaces]0,133
130 data 95,133, 90,133, 88,169,208,133      3064
140 data 96,169,240,133, 89,169,224,133      2942
150 data 91, 32,191,163,169, 55,133,         2947
   [2spaces]1
160 data 88, 96                               909
170 rem ##############################
180 rem mp: setcurs (sys866,x,y,text)
190 rem ##############################
200 data 32,253,174, 32,158,183,134,211      3112
210 data 32,253,174, 32,158,183,134,214      3208
215 data 32, 16,229, 32,253,174, 76,160      3177
220 data 170                                  578
221 rem ########################
222 rem mp: setmob0 (sys885,x,y)
223 rem ########################
224 data 32,253,174, 32,138,173, 32,170      3084
225 data 177,141, 16,208,140,[2spaces]0,     2798
   208, 32
226 data 253,174, 32,158,183,142,[2spaces]   3097
   1,208
227 data 96                                   461
230 rem #########
240 rem chr: heck
250 rem #########
260 data 81:rem <shift-q>                     648
270 data 252,[2spaces]6,227, 35, 35,227,     3189
   [2spaces]6,252
280 data 107:rem <c= -q>                      769
290 data 165,165,189,129,129,195,126, 60     4166
300 rem ### mitte ###
310 data 64:rem <shift-*>                     656
320 data 255,[2spaces]0,255,126,126,255,     3004
   [2spaces]0,255
330 data 95:rem <c= -*>                       666
340 data 165,189,189,189,189,189,189,165     4048
350 rem ### bug ###
360 data 62:rem < '>' >                       648
370 data [2spaces]3, 12, 49,198,198, 49,     2510
    12,[2spaces]3
380 data 30:rem < '↑' >                       634
390 data 24, 24, 36, 36, 66, 90,153,165      2921
400 rem ### explosion ###
410 data 65:rem <shift-a>                     660
420 data 126,255,255,219, 24, 24, 60,255     3366
460 rem ### welle ###
470 data 67:rem <shift-c>                     668
480 data [2spaces]0,[2spaces]0, 24, 60,      2222
   231,195,[2spaces]0,[2spaces]0
490 rem ### feld ###
500 data 35:rem < '#' >                       654
510 data 255,129,129,129,129,129,129,255     4320
520 data -1:rem ** chr ende **                626
530 rem ################
540 rem sprite: cursor
550 rem ################
560 data 16:rem block 16 (50176-50239)        654
570 data 195,[2spaces]0,[2spaces]0,231,      2300
   [2spaces]0,[2spaces]0,126,[2spaces]0
580 data [2spaces]0, 60,[2spaces]0,          1985
   [2spaces]0, 60,[2spaces]0,[2spaces]0,
   126
590 data [2spaces]0,[2spaces]0,231,          2061
   [2spaces]0,[2spaces]0,195
600 data -2:rem ** 'cursor' ende **           630
610 rem ### kanone ###
620 data 17:rem block 17 (50240-50303)        658
630 data 15,128,[2spaces]0, 15,152,          2341
   [2spaces]0,[2spaces]7, 48
640 data [2spaces]0,255,224,[2spaces]0,      2658
   255,224,[2spaces]0,[2spaces]7
650 data 48,[2spaces]0, 15,152,[2spaces]0,   2397
    15,128
660 data -2:rem ** 'kanone' ende **           630
670 rem ### schuss ###
680 data 18:rem block 18 (50304-50367)        662
690 data [2spaces]0,[2spaces]0,[2spaces]0,   1447
   [2spaces]0,[2spaces]0,[2spaces]0,
   [2spaces]0,[2spaces]0
700 data [2spaces]0, 63,[2spaces]0,          1303
   [2spaces]0, 63
710 data -2:rem ** 'schuss' ende **           630
720 data -3:rem ** sprites  ende **           634
730 rem ####
740 rem init
750 rem ####
760 v=53248:s=54272:jp=56320:e=65:sm=891:    4650
   rem vic,sid,controlport 2,'set mob'
765 poke jp+5,40:rem cursorgeschw.            760
770 ba=49152:fa=55296:sc=866:rem bs-addr.,    2292
   farb-addr.,'set cursor'
780 def fn t(x)=peek (po+x)<>e and peek       4750
   (po+x)<>160 and peek (po+x)<>67:rem fe
   ld frei ?
790 def fn f(x)=peek (f+x) and 15:rem farb    1977
   e
795 def fn j(x)=(o and x)=0:rem joystick      1228
800 fl=s:fh=s+1:w=s+4:ah=s+5:hk=s+6:la=s+     4381
   24:rem sid-addressen
810 poke v+32,6:poke v+33,6:print "[cbm 7]    2481
   ":rem rahmen-/grund-/schrift-farbe
820 def fn x(h)=int ((h-24)/8):def fn         4048
   y(h)=int ((h-50)/8):rem x/y-koordinate
830 rem schiffe: waagerecht
840 w$(1)=">[2shift *]Q":w$(2)=">[shift *]    6011
   Q":w$(3)=w$(2):w$(4)=">Q":w$(5)=">Q
850 rem senkrecht
860 s$(1)="↑[left,down,cbm *,left,down,      8259
   cbm *,left,down,cbm q]":s$(2)="↑[left,
   down,cbm *,left,down,cbm q]":s$(3)=
   s$(2):s$(4)="↑[left,down,cbm q]":
   s$(5)=s$(4)
```


WWW.HOMECOMPUTERWORLD.COM

```
870 rem x-koordinaten der menuefelder [x()
    ] und laenge der schiffe [l()]
880 x(1)=1:x(2)=10:x(3)=18:x(4)=26:x(5)=    7315
    33:l(1)=4:l(2)=3:l(3)=3:l(4)=2:l(5)=2
890 if peek (2)=1 then 1040:rem datas scho  1741
    n gelesen ?
900 for i=832 to 915:read a:poke i,a:next   2345
910 sys 832:poke 850,160:sys 832:poke       5530
    56576,peek (56576) and 252:poke v+24,
    10
920 poke 648,192:rem bs-> $c000             1053
940 read z:if z<0 then 970                  1504
950 for i=0 to 7:read a:poke 59392+8*z+i,a  3095
960 next :goto 940                           558
970 read z:if z=-3 then 1010                1874
980 for i=0 to 62:read a:if a>-2 then poke  3039
    ba+z*64+i,a:next
990 if a<0 then for i=i to 62:poke ba+z*    3296
    64+i,0:next
1000 goto 970                                600
1010 for i=0 to 2:poke 50168+i,16+i:next :  2723
    rem sprite-block-addressen
1020 for i=0 to 5:poke v+i,0:next           1730
1030 poke v+39,1:poke v+40,0:poke v+41,0:   3273
    poke 2,1
1040 poke 648,4:print "[clr]":poke 648,192: 2790
    rem sprite-zeiger setzen,bs (1024) loe
    schen
1045 poke v+24,10                            938
1050 rem #########
1060 rem bs-aufbau
1070 rem #########
1075 poke 53265,27:poke 56333,159           2454
1080 a$="CCCCCCCCCCCCCCCCC":a$=a$+"         4220
    [5spaces]"+a$:print "[clr]"a$
1090 for i=1 to 15:print "C################C 4709
    [5spaces]C################C":next
1100 print a$                                427
1110 for i=1 to 5:sys sc,x(i),18,"[cbm 7,   3544
    rvs] [off,cbm 8,right]"w$(i):print
1120 print tab(x(i))"[cbm 7,rvs] [off,      2250
    cbm 8,right]"s$(i):next
1130 sys sc,26,23,"[cbm 7,rvs] [off] EXIT   1986
1140 xk=32:yk=58:xg=164:yg=58:xs=xg:ys=yg:  6000
    poke v+21,1:rem sprites x/y
1150 poke v+2,xg:poke v+3,yg:poke v+4,xs:   4413
    poke v+5,ys
1160 sys sm,xk,yk:rem sprite setzen         1086
1170 rem ### joystick-abfrage ###
1180 o=peek (jp)                             504
1190 if fn j(1) and yk>58 then yk=yk-8      2432
1200 if fn j(2) and yk<242 then yk=yk+8     2485
1210 if fn j(4) and xk>32 then xk=xk-8      2263
1220 if fn j(8) and xk<320 then xk=xk+8     2343
1230 if fn j(16) then gosub 1380:goto 2960  2012
1250 goto 1160                               611
1260 rem ##################
1270 rem toene: explosion
1280 rem ##################
1290 for j=15 to 0 step -1:poke la,j:poke   5234
    ah,15:poke hk,0:poke fh,1:poke fl,20:
    poke w,129
1300 next :poke w,0:poke ah,0:return        1957
1310 rem ### pfeifen ###
1320 k=150-j                                 807
1330 poke la,15:poke ah,96:poke hk,55:poke  5234
    fh,k:poke fl,120:poke w,17:return
1340 rem ### 'wasser' ###
1350 for j=8 to 0 step -.8:poke la,j:poke   4919
    ah,15:poke hk,45:poke fh,100:poke fl,
    120
1360 poke w,129:next :poke la,0:poke w,0:   2627
    poke ah,0:return
1370 rem ### signal ###
1380 poke la,15:poke ah,21:poke hk,0:poke   4810
    fh,114:poke fl,162:poke w,17
1390 for j=1 to 15:next :poke w,0:poke ah,  3491
    0:poke la,0:return
1400 rem ######################
1410 rem schiffe setzen (comp)
1420 rem ######################
1430 for c=1 to 5                            842
1440 r=sgn ( int ( rnd (0)*3)-1):if r=0     2792
    then 1440:rem richtung des schiffes
1450 x=int ( rnd (0)*15)+1:y=int ( rnd (0)* 3353
    15)+1:rem position des sch.
1460 po=49174+x+40*y:pb=1046+x+40*y:rem neu 3772
    er bs/ alter bs
1470 for j=0 to l(c)-1:p=peek (pb+j*(-(r=-  7053
    1)*39+1)):q=peek (po+j*(-(r=-1)*39+1))
1480 if p<>32 or q<>35 then 1440:rem positi 1993
    on besetzt ?
1490 next                                    130
1500 for j=0 to l(c)-1:poke pb+j*(-(r=-1)*  4575
    39+1),48+c:next :rem sch.-nr. auf bs(1
    024)
1510 cx(c)=x:cy(c)=y:cr(c)=r                2287
1520 next :xk=208:yk=58                     1106
1530 sys sc,1,17,"[rvs]COMPUTER[off]:" spc( 4322
    19)":[rvs]SPIELER
1540 a=int ( rnd (ti)*5):if a/2=int (a/2)   3422
    then 1900:rem comp
1550 goto 1590:rem spieler                   722
1560 rem ###########
1570 rem spielerzug
1580 rem ###########
1590 poke v+21,1:gosub 2350                 1512
1600 sys sm,xk,yk                            970
1610 o=peek (jp):rem ### joystick ###        533
1620 if fn j(1) and yk>58 then yk=yk-8      2432
1630 if fn j(2) and yk<170 then yk=yk+8     2473
1640 if fn j(4) and xk>208 then xk=xk-8     2391
1650 if fn j(8) and xk<320 then xk=xk+8     2343
1660 if fn j(16) then 1690                  1336
1680 goto 1600                               583
1690 poke v+21,0:rem 'cursor' aus            831
1700 rem ### schuss auswerten ###
1710 b=41*5+fn x(xk)+40*fn y(yk)            2467
1720 p=b-41*5+ba:if peek (b)=32 and peek    3364
    (p)<>160 goto 1850
1730 if peek (p)>53 then 1590:rem 'daneben' 1744
1740 q=peek (b)-48:ct(q)=ct(q)+1            2113
1750 sys sc,fn x(xk),fn y(yk),"[wht]A       2392
    [cbm 8,left]";:gosub 1290
1760 print "A                              228
1770 if ct(q)<>l(q) then 1840               1475
1780 print "[blk]";:sys sc,cx(q)+22,cy(q),; 2483
1790 if cr(q)=1 then print w$(q):goto 1810  2011
1800 print s$(q)                             586
1810 print "[cbm 7]":cv=cv+1                1395
1820 sys sc,39-l(q),17+q,"[blk]"w$(q)"      2863
    " [cbm 7]":rem versenkt
1830 if cv=5 then 2190:rem spielende        1515
1840 ps=ps+1:goto 1590                      1472
1850 gosub 1350:ps=ps+1                     1394
1860 poke p,160:poke fa+p-ba,14             2087
1870 rem #########
1880 rem comp-zug
1890 rem #########
1900 gosub 2350                              617
1910 poke v+2,164:poke v+3,58:poke v+21,2:  2627
    poke v+4,164
1920 if (zt) or x then x=0:gosub 2390:if x  3085
    then 1990:rem war letzter schuss treff
    er ?
1930 x=int ( rnd (ti)*15)+1:y=int ( rnd     5327
    (ti)*15)+1:po=ba+x+40*y:p=peek (po)
1940 f=peek (fa+x+40*y) and 15               1794
1950 if p=e or p=160 or f=0 then 1930       2067
1960 q=peek (po-1):r=peek (po+1)            1929
1980 if int ( abs (x-y)/2)<>abs (x-y)/2     3218
    then 1930
1990 for j=0 to y*8-8:poke v+3,yg+j:next :  5850
    poke v+5,ys+y*8-8:poke v+21,6
2000 for j=0 to 8*(15-x)+18 step 8:gosub    5163
    1320:poke v+4,xs-j:next :poke la,0:
    poke w,0
2010 poke ah,0:po=ba+x+40*y:p=peek (po):cs= 3423
    cs+1
2020 if p<>35 then 2050                     1467
2030 poke po,160:poke fa+x+40*y,14:poke v+  3705
    21,2:gosub 1350
2040 poke v+21,0:goto 1590                  1482
2050 poke po,e:poke fa+x+40*y,1:gosub 1290  2181
2060 poke fa+x+40*y,15:cx=x:cy=y            2638
2070 pt=peek (41*5+x+40*y)-48               2623
2080 pt(pt)=pt(pt)+1                        1554
2090 if pt(pt)=l(pt) goto 2110              1766
2100 goto 1900                               595
2110 pv=pv+1:sys sc,px(pt),py(pt),"[blk]";: 3057
    x=0
2120 if r(pt)=1 then print w$(pt):goto 2140 2163
```


WWW.HOMECOMPUTERWORLD.COM

```
2130 print s$(pt)                                    745
2140 sys sc,0,17+pt,w$(pt)"[cbm 7]":gosub 2810      2697
2150 if pv<5 goto 1900                               1053
2160 rem ####
2170 rem ende
2180 rem ####
2190 print "[home,wht]":poke v+21,0:if cv=5 then 2240  2340
2200 for j=1 to 5:if ct(j)=l(j) then 2230            2551
2210 sys sc,cx(j)+22,cy(j),;:if cr(j)=1 then print w$(j)  3760
2220 if cr(j)=-1 then print s$(j)                    1746
2230 next :goto 2280                                  483
2240 for j=1 to 5:if pt(j)=l(j) then 2270            2692
2250 sys sc,px(j),py(j),;:if r(j)=1 then print w$(j)  2706
2260 if r(j)=-1 then print s$(j)                     1768
2270 next                                             130
2280 sys sc,5,18,;                                    916
2290 if pv=5 then print "Pech gehabt ! Sie verlieren !"  3446
2300 if cv=5 then print "Hurra ! Sie haben gewonnen !!"  3644
2310 print tab(12)"Nochmal (J/N) ?                   2322
2320 get a$:if a$="j" then run 80                    1532
2330 if a$<>"n" then 2320                            1448
2340 print "[clr,asc142,3down,3right]dann eben nicht !":end  2640
2350 sys sc,10,17,"[cbm 7]"cs tab(28-len ( str$ (ps)))ps:return  3096
2360 rem ############
2370 rem comp-taktik
2380 rem ############
2390 if zt>0 then cx=x(zt):cy=y(zt):zt=zt-1          3867
2395 x=cx:y=cy:po=ba+x+40*y:f=fa+x+40*y               3552
2400 if peek (po)<>e then x=0:return                 1926
2430 if peek (po-1)=e and fn t(1) and fn f(1) then x=x+1:return  3702
2450 if peek (po+1)=e and fn t(2) and fn f(2) then x=x+2:return  3758
2470 if peek (po+1)=e and peek (po+2)=e and fn t(3) and fn f(3) then x=x+3:return  4975
2500 if peek (po+1)=e and fn t(-1) and fn f(-1) then x=x-1:return  4109
2530 if peek (po-1)=e and fn t(-2) and fn f(-2) then x=x-2:return  4110
2550 if peek (po-1)=e and peek (po-2)=e and fn t(-3) and fn f(-3) then x=x-3:return  5174
2570 if peek (po-40)=e and fn t(40) and fn f(40) then y=y+1:return  2870
2590 if peek (po+40)=e and fn t(-40) and fn f(-40) then y=y-1:return  3747
2610 if peek (po-40)=e and fn t(-80) and fn f(-80) then y=y-2:return  3788
2630 if peek (po+40)=e and fn t(80) and fn f(80) then y=y+2:return  2905
2670 if peek (po+40)=e and peek (po+80)=e and fn t(120) and fn f(120) then y=y+3:return  5632
2690 if peek (po-40)=e and peek (po-80)=e and fn t(-120) and fn f(-120) then y=y-3:return  5550
2700 rr=int ( rnd (ti)*3)-1:if rr=0 then 2700        2987
2710 rs=int ( rnd (ti)*3)-1:if rs=0 then 2710        2994
2720 pp=rs*(-(rr=-1)*39+1)                           2441
2730 pq=41-abs (pp)                                  1338
2740 if fn t(pp) and fn f(pp) then b=pp: goto 2780   2682
2750 if fn t(-pp) and fn f(-pp) then b=-pp: goto 2780  2888
2760 if fn t(pq) and fn f(pq) then b=pq: goto 2780   2734
2770 if fn t(-pq) and fn f(-pq) then b=-pq           2197
2780 if abs (b)=1 then x=x+b:return                  1586
2790 y=y+b/40:return                                  717
2800 rem
2810 n=pt:r=r(n):l=l(n):x=px(n):y=py(n):p=ba+x+40*y  4737
2820 if peek (p+(-(r=-1)*39+1)*l)=e then cx=x-(r=1)*l:cy=y-(r=-1)*l:gosub 2921  7206
2830 if peek (p+(r=-1)*39-1)=e then cx=x+(r=1):cy=y+(r=-1):gosub 2921  5120
2840 for i=0 to l-1                                  1203
2850 if r=1 then 2890                                1052
2860 if peek (p+1+40*i)=e then cx=x+1:cy=y+i:gosub 2921  4043
2870 if peek (p-1+40*i)=e then cx=x-1:cy=y+i:gosub 2921  4067
2880 next :x=0:return                                 729
2890 if peek (p+i+40)=e then cx=x+i:cy=y+1: gosub 2921  3512
2900 if peek (p+i-40)=e then cx=x+i:cy=y-1: gosub 2921  3577
2910 next                                             130
2920 x=0:return                                       822
2921 zt=zt+1:x(zt)=cx:y(zt)=cy:return                3156
2930 rem ################
2940 rem schiffe set/del
2950 rem ################
2960 f=0:if yk<180 then on mo goto 3070              2737
2970 if fn f(v+39)=0 then 1160                       1467
2975 if yk<180 then 3055                             1296
2980 if fn y(yk)<>18 then 3020                       1706
2990 for i=1 to 5:if fn x(xk)=x(i) and r(i)=0 then r(i)=1:mo=1:goto 3320  5221
3010 next                                             130
3020 if fn y(yk)<>20 then 3270                       1805
3030 for i=1 to 5:if fn x(xk)=x(i) and r(i)=0 then r(i)=-1:mo=1:goto 3320  5613
3050 next :goto 1160                                  650
3055 for i=1 to 5:if peek (4↑5+fn x(xk)+40*fn y(yk))=48+i goto 3220  4805
3058 next :goto 1160                                  650
3060 rem ### set ###
3070 if xk>144 then 1160                             1370
3080 for j=0 to l(i)-1:po=ba+fn x(xk)+40*fn y(yk)    4001
3090 if r(i)=1 then if peek (po+j)<>35 then 1160     3203
3100 if r(i)=-1 then if peek (po+j*40)<>35 then 1160  3519
3110 next :poke 204,1                                 785
3120 px(i)=fn x(xk):py(i)=fn y(yk):sys sc, px(i),py(i),;  4007
3130 print "[cbm 8]";:if r(i)=1 then print w$(i):goto 3150  2843
3140 print s$(i)                                      713
3150 sys sc,x(i),18,"[rvs,blk] [cbm 7,off]" :print :print tab(x(i))"[rvs,blk] [cbm 7,off]  3343
3160 for j=0 to l(i)-1                               1317
3170 if r(i)=1 then poke 4↑5+40*fn y(yk)+fn x(xk)+j,48+i  4532
3180 if r(i)=-1 then poke 4↑5+40*fn y(yk)+fn x(xk)+j*40,48+i  4817
3190 next                                             130
3200 poke v+39,1:mo=0:goto 1160                      2187
3210 rem ### del ###
3220 a=px(i):b=py(i):sys sc,a,b,;                    2219
3230 if r(i)=1 then for j=0 to l(i)-1:print "#";:poke 4↑5+a+j+40*b,32:next  5550
3240 if r(i)=-1 then for j=0 to l(i)-1: print "#[left,down]";:poke 4↑5+a+(b+j)*40,32:next  5414
3250 print :r(i)=0:px(i)=0:py(i)=0:mo=0              2739
3260 sys sc,x(i),18,"[rvs] [off,down]": print tab(x(i))"[rvs] [off]":goto 1160  3473
3270 if fn x(xk)<>26 or fn y(yk)<>23 then mo=0:goto 1160  3808
3280 rem ### exit menue ###
3290 for j=1 to 5:if r(j)=0 then 1160                2254
3300 next :poke v+21,0                                904
3310 for j=18 to 23:poke 781,j:sys 59903: next :goto 1430  3529
3320 poke v+39,0:sys sc,fn x(xk),fn y(yk),; :poke 204,0  3282
3330 if ( peek (jp) and 16)=0 then 3330              1979
3340 goto 1160                                        611
3350 rem
3360 rem ### das war's ###
3370 rem
```

SVS glänzt durch schöne Grafik und Joystick-Steuerung

WWW.HOMECOMPUTERWORLD.COM

„Othello“: Wer hat den richtigen Dreh?

System: C16/116/Plus4
Typ: Spiel
Sprache: Basic/ASM

Dieses Programm simuliert Spielfeld und Gegner des gleichnamigen Denkspieles. In drei Stufen tritt man gegen den Computer oder einen weiteren Spieler an.

Je nach Schwierigkeitslevel benötigt der Commodore 2 bis 45 Sekunden Denkzeit. Die Denkzeit des Spielers ist unbegrenzt. Spielen zwei Spieler gegeneinander, prüft das Programm jede Eingabe auf Korrektheit ab.

Die Spielregeln: Man setzt einen Stein so, daß damit mindestens ein gegnerischer Stein zwischen diesem und einem eigenen auf dem Brett befindlichen Stein eingeschlossen wird. Das gilt sowohl in waagrechter und senkrechter, als auch in diagonaler Reihe. Alle eingeschlossenen Gegner wechseln ihre Farbe und werden somit zu eigenen Steinen. Die Eckfelder sind strategisch besonders wichtig, da sie ihre Farbe nicht mehr wechseln können. Kann ein Spieler nicht setzen, ist der andere Spieler am Zug. Freiwilliger Verzicht und Rücknahme eines Zuges ist nicht erlaubt! Sieger ist, wer zum Schluß die meisten Steine seiner Farbe besitzt. Eine Partie ist dann zu Ende, wenn alle 64 Felder besetzt sind oder beide Spieler keinen gültigen Zug mehr ausführen können.

Die Eingabe eines Zuges erfolgt durch Angabe der Feldnummer, ähnlich einem Schachbrett, zum Beispiel „D3“ oder „C6“. Unerlaubte Eingaben lehnt der Computer ab. Dafür zeigt er nach Druck auf die HELP-Taste alle derzeit möglichen Felder an. Ein vorzeitiger Spielabbruch ist nur durch Betätigung der 'F1'-Taste zu erreichen, da die RUN/STOP-Taste gesperrt ist.

Diese Version läuft auf allen Computern der 16er Serie. Die zeitkritische Zugsuche des Computers steht am Anfang des Programmes; der Aufbau des Spielfeldes und die Vorbesetzung der Variablen am Ende des Programmes.

1110 — 1560: Die Variablen werden durch die DIM-Anweisung in der Folge angelegt, die der Häufigkeit ihres Gebrauchs entspricht. Dadurch wird eine deutliche Beschleunigung des Programmablaufs erreicht. Das Array „A%(99)“ speichert die Belegung des Spielbretts und muß als erstes Feld dimensioniert sein, da es vom Maschinenprogramm nur an dieser Stelle gesucht wird.

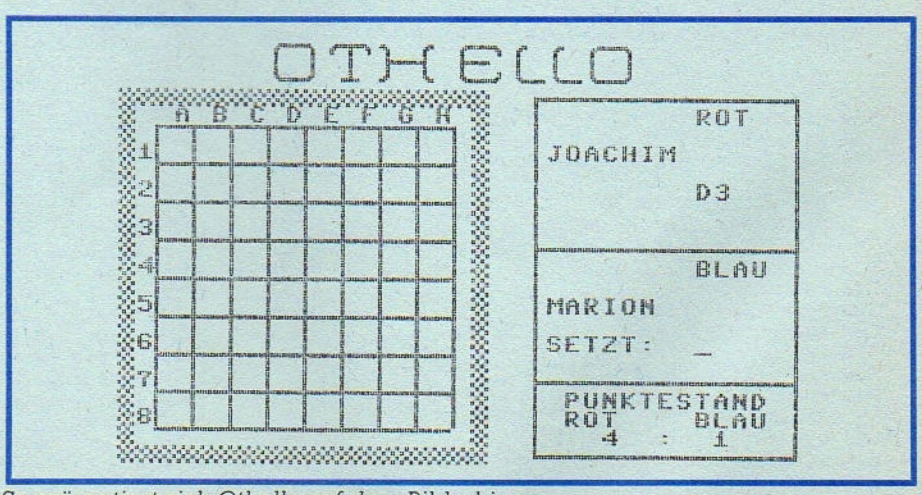
1570 — 1600: Es werden aus den DATA-Zeilen die Werte für die Richtung der Suche nach einem Zug und die Zahlen für die Feldbewertung in die Arrays eingelesen.

1610 — 1670: Das Maschinenprogramm wird aus den DATAs gelesen und in den Kassettenpuffer geschrieben.

1690 — 1960: Das Spielfeld und die Anzeigefelder werden aufgebaut und beschriftet. Dann wird die Grundstellung eingenommen: Je 2 Steine in der Mitte, diagonal versetzt.

90 — 170: Hier erfolgt die Auswahl, welcher Spieler am Zug ist und die verschiedenen Abfragen, ob das Spielende erreicht ist.

200 — 560: Der Computer errechnet sich seinen Zug. In Stufe eins reicht dafür der Bereich bis Zeile 280. Das hier benutzte Programm in Maschinencode belegt den Bereich ab $0332. Es benutzt das RAM ab $0660 und die Stellen $DA bis $DD der Zero Page. Der Aufruf SYS CY,E in Zeile 210 kopiert die Werte des Feldes „A%(99)“ in das RAM ab $0660 und nimmt den Wert der Variablen „E“ (aktuelle Spieler-Nummer) mit. Der in der FOR/NEXT-Schleife enthaltene USR-Aufruf übergibt der Variablen „VD“ die Anzahl der zu erobernden Steine des Feldes „P“. Ist das abgefragte Feld bereits besetzt oder kann nichts „gedreht“ werden, wird hier eine Null übergeben. Diese Abfrage in Basic zu gestalten wäre eine wahre Geduldsprobe. In den stärkeren Stufen wird in Zeilen 350, 440 und 500 mit SYS CD,x der gefundene Zug zur Probe gesetzt und aus der Sicht des Gegners mit dem USR-Befehl beurteilt. Aus dem Feld BW(xx) wird der strategische Wert des Zuges in die Berechnungen miteinbezogen. In



So präsentiert sich Othello auf dem Bildschirm

WWW.HOMECOMPUTERWORLD.COM

Zeile 560 wird mit dem erneuten Aufruf von SYS CY,E der alte Zustand wiederhergestellt.
590 — 700: Mit einem blinkenden Strich-Cursor wird der Spieler zur Eingabe aufgefordert. INSTR prüft auf gültige Zeichen. Die Koordinaten werden umgesetzt und an den folgenden Abschnitt übergeben. Mit „HELP" wird zur Zug-Übersicht verzweigt. Auch der Abbruch mit „F1" wird hier erkannt.
730 — 1000: Das vom Computer oder vom Spieler gewählte Feld wird hier geprüft, gegebenenfalls besetzt und die eingeschlossenen Steine „umgedreht". Nachdem der neue Spielstand errechnet und angezeigt wurde, erfolgt eine Anpassung der Bewertung für die Züge des Computers. 1030 — 1080: Dieses Unterprogramm testet ob der Spieler überhaupt setzen kann. Mit „HELP" aufgerufen, teilt es spielbare Felder mit.
1990 — 2200: Hier werden die Meldungen und Hinweise erzeugt, die im Spielablauf und am Ende angezeigt werden. Die RUN/STOP-Taste wird in Zeile 2090 ignoriert. Für den Fall, daß noch Basic-Fehler im Programm auftauchen, werden diese hier angezeigt, ohne das Bild zu zerstören.

Joachim Rieger

```
 10 rem **********************
 20 rem * o t h e l l o  c16 * 15.10.86
 30 rem *   joachim rieger   *
 40 rem *   4790 paderborn   *
 50 rem **********************
 60 scnclr :trap 2070:gosub 1110          1437
 70 :
 80 f=rnd (0):f=0                          965
 90 if e<>cp then f=1:gosub 1030:if f then 2091
    120
100 sound 1,930,3:char ,25,e*8,"setzt:     3032
    [7spaces]"
110 if e=cp then gosub 200:else gosub 590  1912
120 if f then gosub 1990                    705
130 if f>1 then 80                          755
140 if fe=2 then 2100                      1164
150 if z(0)=64 then 2110                   1610
160 if z(e)=0 or z(g)=0 then 2110          1771
170 e=3-e:g=3-g:goto 80                    1467
180 :
190 rem   computer zieht
200 char ,30,e*8,"[flashon]![flashoff]":   2222
    ti$="000000"
210 vp=0:vm=-1000:sys cy,e                 2304
220 for vs=1 to 60:p=f%(vs)                1775
230 vd=usr (p):if vd=n then 280            2021
240 if vd=z(g) then vp=p:vs=99:goto 280    2940
250 vn=vd+bw(p)+rnd (1)*(s+1)              2425
260 if s>1 then gosub 350:vn=vn-gm         2494
270 if vn>vm then vm=vn:vp=p               1990
280 next :if vp=n then i$="??":f=1:goto    2852
    310
290 p=vp:x=int (p/10):y=p-x*10             2795
300 i$=mid$ (p1$,x,1)+mid$ (p2$,y,1)       2556
310 char ,25,e*8,right$ (ti$,2)+" sec      3061
    [2spaces]"+i$
320 if f then return :else 700             1179
330 :
340 rem   gegenzuege testen
350 if z(0)=4 then return                  1206
360 sys cd,p:gm=-1000                      1456
370 for gs=1 to 60:gp=f%(gs)               1845
380 gd=usr (gp):if gd=n then 420           1921
390 if gd=z(e)+vd+1 then gd=gd+500         2319
400 gd=gd+bw(gp):if gd>gm then gm=gd:pg=gp 3429
410 if vn-gd<vm then gs=99                 1491
420 next :if s=2 or gm=-1000 then 560      1840
430 :
440 sys cd,pg:em=-1000                     1457
450 for es=1 to 60:et=f%(es)               1852
460 ed=usr (et):if ed=n then 480           1854
470 ed=ed+bw(et):if ed>em then em=ed:eg=et 3151
480 next :if em=-1000 then 550             1676
490 :
500 sys cd,eg:nm=-1000                     1413
510 for ns=1 to 60:nt=f%(ns)               1765
520 nd=usr (nt):if nd=n then 540           2056
530 nd=nd+bw(nt):if nd>nm then nm=nd       2928
540 next                                    130
550 gm=gm-(em-nm)*.6:nm=n                  2418
560 sys cy,e:return                         808
570 :
580 rem   eingabe
590 char ,33,e*8,"[flashon,cbm @]          1543
    [flashoff]"
600 get i$:x=instr (p1$,i$)                1725
610 if x then char ,33,e*8,i$+"[flashon,   2748
    cbm @,flashoff]":else 600
620 get i$:y=instr (p2$,i$)                1743
630 if i$=chr$ (20) then 590               1407
640 if x>8 then char ,34,e*8," ":else 680  2249
650 if x=9 then 2070                       1125
660 char ,0,23,"> ":gosub 1030             1704
670 print "<";:goto 590                    1133
680 if y then char ,34,e*8,i$:else 620     2294
690 p=x*10+y:char ,25,e*8,"[6spaces]"      2112
700 for i=23 to 24:char ,0,i,chr$ (27)+"q" 2851
    :next
710 :
720 rem   zug ausfuehren
730 f=3:if a%(p) then f=2:return           1597
740 for r=0 to 7:np=p:d=n                  1865
750 for j=1 to 7:np=np+rx(r)               1919
760 if a%(np)=n then j=8:goto 780          1837
770 if a%(np)=e then d=j-1:j=8             1982
780 next j:if d=n then 820                 1462
790 f=n:fe=n:gosub 960:np=p                2005
800 for i=1 to d:p=p+rx(r):gosub 950       2579
810 next i:p=np                             902
820 next r:if f then return                1125
830 z(e)=z(e)+1                             876
840 :
850 char ,26,21,"":z(0)=z(e)+z(g)          2617
860 print using "###";z(1),"[2spaces]:",   2672
    z(2);

870 :
880 if e<>cp then 930:rem bewertg.anpassen 1454
890 if p=11 then bw(22)=bw(33):bw(12)=     5077
    bw(13):bw(21)=bw(13)
900 if p=18 then bw(27)=bw(36):bw(17)=     5128
    bw(16):bw(28)=bw(16)
910 if p=81 then bw(72)=bw(63):bw(71)=     5199
    bw(83):bw(82)=bw(83)
920 if p=88 then bw(77)=bw(66):bw(78)=     5379
    bw(86):bw(87)=bw(86)
930 return                                  142
940 :
950 z(e)=z(e)+1:z(g)=z(g)-1                2527
960 color 1,c(e),1(e)                      1175
970 px=fn x(p):py=fn y(p):a%(p)=e          2035
980 sound 1,739+e*55,2:sound 1,739+g*55,2  3625
990 char ,px,py,a$:char ,px,py+1,b$        2347
1000 color 1,c(0),1(0):return              1556
1010 :
1020 rem   noch zug moeglich?
1030 sys cy,e:for x=10 to 80 step 10:for y= 2207
     1 to 8
1040 d=usr (x+y):if d=n then 1080          1879
1050 if f then f=0:x=90:y=9:goto 1080      2380
1060 z$=mid$ (p1$,int (x/10),1)+mid$ (p2$, 2883
     y,1)+" "
1070 print z$;:if pos (d)>36 then print "  2322
     [4spaces]"
1080 next :next :return                     697
1090 :
1100 rem   initialisierung/brett
1110 color 1,8,2:color 0,10,5:color 4,10,5 3378
1120 cy=818:cd=930:rem sys-adr             1593
1130 poke 1281,90:poke 1282,3:rem usr-adr  1765
1140 dim x,y,p,e,r,gp,gs,gd,gm,j,n,d,s     3366
1150 dim vs,vn,vd,vm,vp,ed,et,es,em,nd,nt  3330
1160 dim nd,nt,ns,nm,np,i,px,py,cp,g,f,fe  3293
1170 def fn x(p)=int (p/10)*2+2            1716
```


WWW.HOMECOMPUTERWORLD.COM

```
1180 def fn y(p)=(p-int (p/10)*10)*2+3          2383
1190 dim a%(99),f%(64),bw(99)                  2130
1200 t$(0)="[8spaces]j.r.[8spaces]"            1771
1210 t$(1)="U[shift *]IU[cbm r]II U U          2997
     [shift *]IU U U[shift *]I"
1220 t$(2)="[shift -,shift-space,shift -]      3376
     [shift -] [shift -,shift-space,
     shift -] [shift -,2spaces,shift -]
     [shift -] [shift -] [shift -]"
1230 t$(4)="[shift -] [shift -] [shift -]      2956
     [cbm q,shift *,cbm w] [cbm q,shift *]
     [shift -] [shift -] [shift -]
     [shift -]"
1240 t$(7)="J[shift *]K [cbm e] K J J          2957
     [shift *]KJ[shift *]J[shift *]J
     [shift *]K"
1250 t$(3)=t$(2):t$(5)=t$(2):t$(6)=t$(2)       3245
1260 t$(8)="[20cbm +]"                         3414
1270 print chr$ (142)+chr$ (8):vol 8           1765
1280 for r=0 to 7:char ,10,r,t$(r):next        2439
1290 char ,8,12,"wie ist dein name"            2239
1300 input s$(1)                                564
1310 char ,8,14,"willst du gegen den"          2374
1320 char ,8,15,"computer spielen?[3spaces]    2567
     "
1330 get i$:if i$<>"j" and i$<>"n" then        2815
     1330
1340 print i$:if i$="j" then cp=1:else 1420    2367
1350 char ,8,16,"welche stufe? (1-3)"          2713
1360 get i$:s=val (i$):if s<1 or s>3 then      3260
     1360
1370 print s                                    319
1380 char ,8,17,"willst du anfangen? "         2753
1390 get i$:if i$<>"j" and i$<>"n" then        2818
     1390
1400 print i$:if i$="j" then cp=2:else         2913
     s$(2)=s$(1)
1410 if cp then s$(cp)="computer[2spaces]"+    3079
     str$ (s):goto 1440
1420 char ,8,20,"name des gegners "            2459
1430 input s$(2)                                580
1440 for e=1 to 2                               750
1450 if len (s$(e))=0 then s$(e)="spieler"+    3001
     str$ (e)
1460 if len (s$(e))>13 then s$(e)=left$        2874
     (s$(e),13)
1470 next                                       130
1480 char ,0,23,"anzeige der moeglichen        4080
     felder[2spaces]:[2spaces]help"
1490 char ,0,24,"spiel vorzeitig abbrechen     3723
     [6spaces]:[3spaces]f1"
1500 f$(1)="nichts geht!!"                     2091
1510 f$(2)="schon belegt!"                     1898
1520 f$(3)="neu eingeben!"                     2295
1530 p1$="abcdefgh*?":p2$="12345678"           3029
1540 c(0)=2:c(1)=3:c(2)=7:c$(1)="rot"          3102
1550 l(0)=1:l(1)=4:l(2)=5:c$(2)="blau"         2917
1560 key 1,"*":key 8,"?"                       1727
1570 for x=0 to 7:read rx(x):next :x=0         2676
1580 for j=1 to 9:read d,vn                    1425
1590 for e=1 to d:read p:bw(p)=vn*.4*s         2808
1600 x=x+1:f%(x)=p:next :next                  1676
1610 if peek (cy)=32 then 1690                 1624
1620 for x=cy to 985 step 8:p=n                1743
1630 for y=0 to 7:read z$:e=dec (z$):p=p+e     2736
1640 poke x+y,e:next                           1268
1650 read r:if p=r then next :goto 1690        1644
1660 y=( peek (63)+peek (64)*256):poke cy,0    2779
1670 scnclr :print "data-fehler in";y:end      2390
1680 :
1690 for y=1 to 8:a$=a$+"O[cbm y]":b$=b$+"     2832
     [cbm h] ":next
1700 l$(0)="[shift -,13spaces,shift -]"        1848
1710 l$(1)="[cbm a]CCCCCCCCCCCCC[cbm s]"       2705
1720 l$(2)="[cbm q,13shift *,cbm w]"           2140
1730 l$(3)="[cbm z,13shift *,cbm x]"           2214
1740 scnclr :r=0:for x=1 to 7 step 3           1647
1750 char ,10,r,t$(x):r=r+1:next               2214
1760 char ,2,3,"[rvs]"+t$(8)                   1382
1770 char ,2,4,"[rvs,cbm +,2spaces]a b c d     2742
      e f g h [cbm +,off]"
1780 for y=1 to 8                               727
1790 char ,2,3+2*y,"[rvs,cbm +] "+a$+"         2551
     [cbm h,cbm +,off]"
1800 char ,2,4+2*y,"[rvs]"+str$ (y)            1803
1810 char ,2,4+2*y,"[rvs,cbm +,right]"+b$+"    2790
     [cbm h,cbm +,off]":next
1820 char ,2,21,"[rvs,cbm +] [16cbm y]         4318
      [cbm +,off]"
1830 char ,2,22,"[rvs]"+t$(8)+"[off]"          1722
1840 for y=4 to 21:char ,24,y,l$(0):next       2502
1850 char ,24,3,l$(1):char ,24,11,l$(2)        2595
1860 char ,24,18,l$(2):char ,24,22,l$(3)       2675
1870 for y=1 to 2:char ,33,y*8-4,c$(y)         2449
1880 char ,25,y*8-2,"[rvs]"+s$(y)+"[off]":     2665
     next
1890 char ,25,19,"[rvs] punktestand [off]"     2684
1900 char ,25,20," rot[4spaces]blau "          1817
1910 :
1920 rem   grundstellung
1930 a$="[rvs]O[cbm y]":b$="[cbm h] [off]"     1674
1940 e=2:p=44:gosub 960:p=55:gosub 960         2238
1950 e=1:p=45:gosub 960:p=54:gosub 960         2360
1960 g=2:z(e)=2:z(g)=2:gosub 850:return        2940
1970 :
1980 rem   fehler/ende
1990 sound 1,169,5:sound 2,378,6               1839
2000 sound 1,169,3:sound 2,499,3               1724
2010 char ,25,8*e+1,"[rvs]"+f$(f)+"[off]"      2487
2020 if f=1 then fe=fe+1                       1625
2030 for j=1 to 1000:next j                    1020
2040 char ,25,8*e+1,"[13spaces]"               1933
2050 return                                     142
2060 :
2070 if er=30 then resume :else trap           1220
2080 if er>0 then char ,0,24,err$ (er)+"       3916
      in"+str$ (el):goto 2120
2090 char ,9,24,"[rvs]abbruch![off]":goto      2246
     2120
2100 char ,0,24,"[rvs]keiner kann mehr         3020
      setzen![off]"
2110 for i=1 to 50:sound 3,1023-i,1:next       2624
2120 s=1:if z(2)>z(1) then s=2                 1929
2130 if z(1)=z(2) then s=0:goto 2170           2346
2140 color 4,c(s),1(s)                         1175
2150 char ,25,s*8-2,"[flashon]"+s$(s)+"        2517
     [flashoff]"
2160 char ,25,s*8,"hat gewonnen!"              2441
2170 if s=0 then char ,25,23,"[flashon]        3162
     unentschieden[flashoff]"
2180 char ,25,24,"nochmal?[2spaces]j/n"        2592
2190 get key i$:scnclr :if i$="j" then run     1903
2200 print chr$ (9):end                         906
2210 :
2220 rem   richtungen
2230 data -1,9,10,11,1,-9,-10,-11              2758
2240 rem   feldwerte
2250 data 4,50,11,18,81,88                     1671
2260 data 8,30,13,16,31,38,61,68,83,86         3196
2270 data 4,20,33,36,63,66                     1801
2280 data 8,16,14,15,41,48,51,58,84,85         2984
2290 data 8,14,34,35,43,46,53,56,64,65         3271
2300 data 8,10,24,25,42,47,52,57,74,75         3175
2310 data 8,06,23,26,32,37,62,67,73,76         3555
2320 data 8,00,12,17,21,28,71,78,82,87         2954
2330 data 4,-6,22,27,72,77                     1780
2340 rem   maschinencode
2350 data 20,de,9d,84,db,a0,00,b1,  1099       3540
2360 data 2f,c9,c1,d0,07,c8,b1,2f,  1080       2556
2370 data c9,80,f0,05,a2,12,6c,00,   862       2906
2380 data 03,a0,ce,a2,63,b1,2f,9d,  1011       2971
2390 data 60,06,88,88,ca,10,f6,60,   934       3020
2400 data 20,71,98,84,dd,85,df,85,  1139       3149
2410 data da,b9,60,06,d0,33,a9,07,   940       2632
2420 data 85,dc,a6,dc,bd,d2,03,85,  1274       2804
2430 data de,a4,dd,a2,00,98,18,65,  1046       2759
2440 data de,a8,b9,60,06,f0,12,c5,  1132       2844
2450 data db,f0,04,e8,4c,77,03,8a,  1031       2760
2460 data a4,da,d0,23,18,65,df,85,  1106       2813
2470 data df,c6,dc,10,d5,a5,da,d0,  1461       2714
2480 data 2b,a9,00,a4,df,4c,71,94,   936       2572
2490 data 20,de,9d,84,dd,84,da,a5,  1279       3264
2500 data db,99,60,06,4c,68,03,8a,   795       3040
2510 data f0,df,a4,dd,98,18,65,de,  1347       2990
2520 data a8,a5,db,99,60,06,ca,d0,  1217       2968
2530 data f3,4c,93,03,a9,00,85,da,   989       3217
2540 data a9,03,38,e5,db,85,db,60,  1124       2941
2550 data 01,ff,0a,f6,09,f7,0b,f5,  1024       3009
```

„Othello": Basic kombiniert mit Maschinensprache

WWW.HOMECOMPUTERWORLD.COM

Entwerfen und Spielen mit „Runman Construction Set“

System : C64/Laufwerk
Typ : Spiel
Sprache: ASM

Runman Construction Set ist ein Geschicklichkeitsspiel für maximal 9 Personen. Ziel des Spieles ist es, in der kürzesten Zeit alle Spielfelder zu durchlaufen und dabei möglichst wenig Figuren zu verlieren. Flammen und Gespenster hindern den Spieler daran, die notwendigen Punkte einzusammeln. Da im Spiel ein Editor integriert ist, mit dem man eigene Spielfelder entwerfen kann, wird das Spiel nicht langweilig. Nach dem Starten des Spiels werden die vorgegebenen Spielfelder in einen Zwischenspeicher kopiert. Anschließend erscheint das Hauptmenü.
F1 Start
F3 Editor
F8 Ende

Spiele spielend selbstgebaut

Durch Drücken der Taste F1 spielt man dann das vorgegebene Spiel, durch F8 beendet man das Programm. F3 führt zum Editor. Der Editor hat einen Speicher von 12 Spielfeldern. Diese kann man mit den Tasten (+/—) umschalten und bearbeiten. Das Spielfeld setzt sich aus verschiedenen Zeichen zusammen. Will man ein Zeichen an eine Position setzen, bewegt man den Pfeil am Bildschirm mittels den Cursortasten zu der Stelle hin und drückt die entsprechende Taste (siehe Tabelle 1.0). Fehler werden sofort angezeigt. Erst wenn alle Spielfelder fehlerfrei sind, gibt „RCS“ diese zum Spielen frei.
Der Editor besitzt eine Funktion, um ein Spielfeld in ein anderes zu kopieren. Man geht einfach mit (+/—) auf das Spielfeld, das kopiert werden soll, und drückt dann

Unendlich viele Spielfelder: Der eingebaute Editor macht's möglich. Selbst ist der Spieler und wählt, wieviele Leitern, Einbahnstraßen, Geister und Flammen den Schwierigkeitsgrad erhöhen.

gleichzeitig CTRL+C. Wenn der Pfeil verschwindet, sucht man sich mit (+/—) ein anderes Spielfeld aus und drückt dann RETURN, oder, wenn man den Modus verlassen will, RUN/STOP.
Die Anzahl der Gegner und Punkte ist nicht unbegrenzt. Es muß jeweils einer auf dem Bildschirm sein. Die maximale Anzahl der Gegner liegt bei 7, die der Punkte bei 255 innerhalb eines Spielfeldes. Auch sind in der obersten Spalte nicht alle Zeichen erlaubt, die entsprechenden Tasten sind gesperrt. Weitere Funktionen:

CLR	Pfeil links oben plazieren
SHIFT/CLR	Spielfeld im Speicher löschen
CR	Pfeil geht eine Spalte tiefer

Es werden immer der aktuelle Spielfeldspeicher (Screen), die Anzahl der Gegner und Punkte angezeigt. Ein Warnsignal ertönt, wenn der Pfeil an den Rand stößt. Spielfelder kann man auf einem Laufwerk abspeichern und laden, wobei es hierbei zwei Möglichkeiten gibt. Taste F1 drücken. Nun erscheint das Menü.
F1 Laden
F3 Speichern
F5 Directory
F7 Zurück zum Editor
Drückt man F1, wird ein Spielfeld, wenn es vorhanden ist, von der Diskette geladen. Dazu muß man vorher den Filenamen eingeben. Abspeichern geschieht analog. Mit F5 hat man die Gelegenheit, alle Spielfelder, die auf Diskette abgespeichert worden sind, hintereinander anzusehen und in einen der 12 Spielfeldspeicher zu kopieren. Das geschieht mit Taste F3. Hierbei wird das momentan sichtbare Spielfeld in den entsprechenden Spielfeldspeicher kopiert; es ertönt ein Signal. Auch hier läßt sich der Speicher mit den Tasten (+/—) umschalten. Wenn man wieder zum Editor will, betätigt man Taste F7.

Tabelle 1.0

Taste	Zeichen
A	Leiter
B	Wegstrecke
C	Wegende von links
D	Wegende von rechts
E	Sperre
F	Anfangszeichen für Tasten G bis J
G	Verzweigung zur Leiter
H	Endzeichen für Tasten G bis J
I	Verzweigung zur Stange
J	Wegsperre von rechts
K	Punkte
L	Schalter für J und T
N	Verzweigung des Leiterendes
O	Stange
P	Ende der Stange
Q	Spieler
T	Wegsperre von links

Input-Return-RUN

Bevor man sein Spiel beginnt, braucht das Programm ein paar Werte. Als erstes muß man die Anzahl der Spielfelder (maximal 12), sowie die Anzahl der Leben (maximal 9), eingeben. Danach werden die Spielfelder geprüft. Ist in einem der Spielfelder ein Fehler, wird dieser sofort angezeigt. Als nächstes muß für jedes

WWW.HOMECOMPUTERWORLD.COM

Spielfeld die Geschwindigkeit des Spielers und der Gegner eingegeben werden. Anschließend sucht man sich die Spielstärke der Gegner aus. Es gibt Flammen, die nach System umherlaufen, und Gespenster, die den Spieler aufsuchen. Weitere belegte Tasten:

Eintippen und Speichern

F1 Rücksprung zur letzten Einstellung
F3 Wiederholung der Parameterroutine
F5 Zurück zum Editor
Ist man damit fertig, beginnt das Spiel.
Zum Schluß muß man die Anzahl der Spieler und deren Namen eingeben. Joystick in Port 2 gesteckt und los geht's. Nun muß der Spieler die Punkte in jedem Spielfeld aufsammeln. Hat der Spieler das Bild geschafft, ist er wieder dran, wenn sämtliche anderen Spieler dieses auch geschafft haben oder ausgeschieden sind. Kollidiert der Spieler mit einem seiner Gegner, so muß er das ganze Bild von vorne beginnen. Nach Ablauf des Spiels erfolgt die Auswertung. Hierbei wird dann der Spielstand in einer Tabelle ausgegeben.
Nun kann man entscheiden, ob man wieder zum Editor geht (F1) oder ganz von vorne beginnt (F3).

Eintipphinweise

Aufgrund seiner Größe ist „Runman Construction Set" in 6 Teilen abgedruckt. Alle 6 Programme sind einzutippen und abzuspeichern. Danach lädt man den „Starterteil" und gibt RUN ein. Innerhalb der nächsten 11 Minuten schreibt sich das Programm als komplettes Maschinensprachenfile auf Diskette. Dieses Programm lädt man mit 'LOAD„name",8' und startet es mit RUN. Wichtig !!! Die Namen der Unterprogramme müssen die Titel der Bildunterschriften tragen. Sollte ein „?File not found-Error" erscheinen, wurde ein Teilprogramm unter falschem Namen gespeichert. „Runman Construction Set" ist, wie alle Programme dieses Heftes, auf einer Sammeldiskette erhältlich.
Die vorgedruckte Zahlkarte befindet sich auf Seite 3.

Christian Drapatz

```
1000 rem ------------------------------
1010 rem   runman construction set v2.0
1020 rem
1030 rem      by christian drapatz '86
1040 rem
1050 rem   programmteil 1 : editor
1060 rem ------------------------------
1070 :
1080 poke 53280,0:poke 53281,0                    1436
1090 poke 646,7:a$=chr$ (34)                      1862
1100 print "[clr,7right,5cbm +,5spaces,          3045
     cbm +,4spaces,cbm +,4spaces,2cbm +,
     3spaces,cbm +]
1110 print "[7right,cbm +,4spaces,cbm +,          2801
     4spaces,cbm +,4spaces,cbm +,4spaces,
     cbm +] [cbm +,2spaces,cbm +]
1120 print "[7right,cbm +,4spaces,cbm +,          2868
     4spaces,cbm +,4spaces,cbm +,4spaces,
     cbm +,2spaces,cbm +] [cbm +]
1130 print "[7right,cbm +,4spaces,cbm +,          2747
     4spaces,cbm +,4spaces,cbm +,4spaces,
     cbm +,3spaces,2cbm +]
1140 print "[7right,5cbm +,5spaces,cbm +,         2963
     4spaces,cbm +,4spaces,cbm +,4spaces,
     cbm +]
1150 print "[7right,cbm +] [cbm +,7spaces,        2625
     cbm +,4spaces,cbm +,4spaces,cbm +,
     4spaces,cbm +]
1160 print "[7right,cbm +,2spaces,cbm +,          2677
     6spaces,cbm +,4spaces,cbm +,4spaces,
     cbm +,4spaces,cbm +]
1170 print "[7right,cbm +,3spaces,cbm +,          2781
     5spaces,cbm +,4spaces,cbm +,4spaces,
     cbm +,4spaces,cbm +]
1180 print "[7right,cbm +,4spaces,cbm +,          3088
     5spaces,4cbm +,5spaces,cbm +,4spaces,
     cbm +]
1190 print "[down,13right]p r e s e n t           2920
1200 print "[down,6right,red]runman               4414
      construction set v2.0
1210 print "[yel,2down,3right]*** please          3537
      wait for 11 minutes ***
1220 sp(0)=32768:sp(1)=36864                      2679
1230 a(0)=3162:a(1)=3938                          1658
1240 n=0                                           371
1250 a=0                                           358
1260 read da:poke sp(n)+a,da                      1502
1270 if a<>a(n) then a=a+1:goto 1260              2485
1280 n=n+1                                         340
1290 if n=1 then 1250                             1136
1300 print "[2down,blk]load"a$"rcs b"a$",8,       2515
     1[3down]"
1310 print "[down]run[8up]                        1486
1320 poke 631,13:poke 632,13                      2109
1330 poke 198,2:end                               1140
1340 :
1350 data 15,128,9,128,195,194,205,56,48,         7075
     104,168,104,170,104,64,165,2,240,6,141
1360 data 4,128,76,226,252,142,22,208,32,         7603
     163,253,32,80,253,32,21,253,32,91,255
1370 data 88,76,0,144,147,17,17,29,144,71,        7725
     69,66,69,78,32,83,73,69,32,68,73,69
1380 data 32,65,78,90,65,72,76,32,68,69,82,       7760
     32,83,80,73,69,76,69,82,32,69,73,78
1390 data 58,255,13,13,29,144,83,73,78,68,        8081
     32,83,73,69,32,68,65,77,73,84,32,69
1400 data 73,78,86,69,82,83,84,65,78,68,69,       7251
     78,32,40,74,47,78,41,32,63,255,13
1410 data 13,29,29,29,29,29,29,29,29,31,1,        7320
     46,32,78,65,77,69,32,58,46,46,46,46
1420 data 46,46,46,46,46,46,46,46,46,46,46,       7939
     46,157,157,157,157,157,157,157,157
1430 data 157,157,157,157,157,157,157,157,        8391
     255,145,145,255,8,147,144,29,29,29
1440 data 29,29,29,29,29,29,29,29,29,32,32,       7346
     32,32,82,32,85,32,78,32,32,32,32,32
1450 data 32,13,13,13,29,29,29,29,29,29,29,       7305
     29,29,29,29,29,29,29,29,80,82,69,83
1460 data 69,78,84,13,13,13,29,29,29,29,29,       7752
     29,82,85,78,77,65,78,32,67,79,78,83
1470 data 84,82,85,67,84,73,79,78,32,83,69,       7198
     84,32,86,50,46,48,13,13,13,13,13,13
1480 data 13,13,13,13,13,13,13,13,13,13,13,       7435
     13,67,72,46,68,82,65,80,65,84,90,44
1490 data 66,69,82,76,73,78,69,82,83,84,82,       7414
     46,49,52,55,44,52,51,48,48,32,69,83
1500 data 83,69,78,32,49,255,145,145,145,145,     6421
     145,145,145,145,145,145,145,145,145
1510 data 145,145,145,28,13,29,29,29,29,29,       7489
     29,29,29,29,29,29,29,29,29,45,45
1520 data 32,77,69,78,85,69,32,45,45,13,13,       7378
     13,29,29,29,29,29,29,29,29,29,29,29
1530 data 29,29,29,29,29,70,49,32,83,84,65,       7144
     82,84,13,13,29,29,29,29,29,29,29,29
1540 data 29,29,29,29,29,29,29,29,70,51,32,       7708
     69,68,73,84,79,82,13,13,29,29,29,29
1550 data 29,29,29,29,29,29,29,29,29,29,29,       7071
     29,70,56,32,69,78,68,69,144,13,13
1560 data 13,29,29,29,29,40,68,82,85,69,67,       7105
     75,69,32,68,73,69,32,69,78,84,83,80
1570 data 82,69,67,72,69,78,68,69,32,84,65,       7478
     83,84,69,41,255,214,157,255,19,17
1580 data 17,17,17,17,17,17,17,17,17,17,17,       8183
     17,17,17,17,17,17,17,17,17,29,29
1590 data 29,29,29,29,29,29,28,83,73,78,68,       7278
     32,83,73,69,32,83,73,67,72,69,82,32
1600 data 40,74,47,78,41,32,63,255,147,29,        7624
     29,29,29,29,29,29,29,144,82,85,78,77
```


WWW.HOMECOMPUTERWORLD.COM

```
1610 data 65,78,32,67,79,78,83,84,82,85,67,  7846
     84,73,79,78,32,83,69,84,32,13,13,29
1620 data 29,29,29,29,29,29,29,29,29,29,29,  7298
     29,29,69,32,68,32,73,32,84,32,79,32
1630 data 82,13,215,215,215,215,215,215,     6237
     215,215,215,215,215,215,215,215,
     215
1640 data 215,215,215,215,215,215,215,215,   8620
     215,215,215,215,215,215,215,215
1650 data 215,215,215,215,215,215,215,17,    8182
     17,17,17,17,17,17,17,17,17,17,17,17
1660 data 17,17,17,13,215,215,215,215,215,   8471
     215,215,215,215,215,215,215,215
1670 data 215,215,215,215,215,215,215,215,   8620
     215,215,215,215,215,215,215,215
1680 data 215,215,215,215,215,215,215,215,   8291
     215,80,85,78,75,84,69,32,58,29,29,29
1690 data 29,29,29,29,71,69,71,78,69,82,32,  7705
     58,32,29,29,29,29,29,83,67,82,69,69
1700 data 78,32,58,13,13,29,29,40,70,49,61,  7366
     68,73,83,75,32,32,70,51,61,80,65,82
1710 data 65,77,69,84,69,82,32,32,70,55,61,  7688
     82,69,83,84,65,82,84,41,255,147,17
1720 data 29,29,29,29,29,144,68,73,83,75,    7359
     45,79,80,69,82,65,84,65,84,73,79,78
1730 data 13,13,13,29,29,29,70,49,32,76,65,  7379
     68,69,78,13,13,29,29,29,70,51,32,83
1740 data 80,69,73,67,72,69,82,78,13,13,29,  7890
     29,29,70,53,32,68,73,82,69,67,84,79
1750 data 82,89,13,13,29,29,29,70,55,32,90,  8031
     85,82,85,69,67,75,32,90,85,77,32,69
1760 data 68,73,84,79,82,255,147,17,17,29,   7462
     29,144,70,73,76,69,78,65,77,69,58,46
1770 data 46,46,46,46,46,46,46,46,46,46,46,  7972
     46,46,46,46,157,157,157,157,157,157
1780 data 157,157,157,157,157,157,157,157,   8160
     157,157,255,13,13,13,13,29,29,255,19
1790 data 29,29,29,29,29,29,29,32,32,32,32,  7017
     32,32,32,144,68,73,82,69,67,84,79
1800 data 82,89,32,32,32,32,32,32,32,32,32,  7501
     13,13,70,73,76,69,78,65,77,69,58,32
1810 data 32,32,32,32,32,32,32,32,32,32,32,  7630
     32,32,32,32,17,17,17,17,17,17,17,17
1820 data 17,17,17,17,17,17,17,17,17,13,13,  7966
     13,13,29,40,70,49,61,87,69,73,84,69
1830 data 82,32,32,70,51,61,85,69,66,69,82,  7908
     78,69,72,77,69,78,32,32,70,55,61,69
1840 data 68,73,84,79,82,41,19,255,19,17,    7659
     17,29,29,29,29,29,29,29,29,29,32,32
1850 data 32,32,32,32,32,32,32,32,32,32,32,  7943
     32,32,32,157,157,157,157,157,157,157
1860 data 157,157,157,157,157,157,157,157,   8100
     157,255,19,13,29,29,29,29,29,29,29
1870 data 29,29,29,29,29,29,29,29,80,65,82,  8013
     65,77,69,84,69,82,13,13,29,71,69,66
1880 data 69,78,32,83,73,69,32,66,73,84,84,  8111
     69,32,68,73,69,32,65,78,90,65,72,76
1890 data 32,68,69,82,32,83,80,73,69,76,45,  7464
     13,29,70,69,76,68,69,82,32,40,77,65
1900 data 88,46,32,49,50,32,83,67,82,69,69,  7736
     78,83,41,32,69,73,78,13,13,29,65,78
1910 data 90,65,72,76,32,58,255,19,17,17,    8165
     17,17,17,17,17,17,17,17,29,87,73,69
1920 data 86,73,69,76,32,76,69,66,69,78,32,  7864
     87,79,76,76,69,78,32,83,73,69,32,73
1930 data 77,32,83,80,73,69,76,13,29,72,65,  7822
     66,69,78,32,63,13,13,29,76,69,66,69
1940 data 78,32,58,255,19,13,29,83,67,82,    7196
     69,69,78,32,58,48,48,13,13,32,65,78
1950 data 90,65,72,76,32,68,69,82,32,80,85,  7731
     78,75,84,69,32,58,48,48,48,13,29,29
1960 data 29,29,29,29,29,29,29,29,29,29,71,  7324
     69,71,78,69,82,32,58,32,32,48,13,13
1970 data 29,71,69,83,67,72,87,73,78,68,73,  7870
     71,75,69,73,84,13,29,40,83,67,72,78
1980 data 69,76,76,32,49,46,46,46,46,57,32,  7913
     76,65,78,71,83,65,77,41,13,13,29,83
1990 data 80,73,69,76,69,82,32,58,255,13,    7659
     29,71,69,71,78,69,82,32,32,58,255,13
2000 data 13,32,71,69,83,84,65,72,76,84,32,  8041
     68,69,82,32,71,69,71,78,69,82,13,29
2010 data 40,71,61,71,69,83,80,69,78,83,84,  7692
     32,32,70,61,70,76,65,77,77,69,41,13
2020 data 255,13,32,48,46,32,71,69,71,78,    7615
     69,82,32,58,255,147,17,17,17,17,17,17
2030 data 17,17,17,17,17,17,17,17,17,17,17,  7988
     17,17,17,17,17,17,17,17,29,29,40,70
2040 data 49,61,90,85,82,85,69,67,75,32,32,  7744
     70,51,61,80,65,82,65,77,69,84,69,82
2050 data 32,32,70,53,61,69,68,73,84,79,82,  7765
     41,255,157,32,157,255,70,76,65,77
2060 data 77,69,255,71,69,83,80,69,78,83,    7307
     84,255,19,29,29,29,29,29,29,29,29,29
2070 data 29,29,29,29,29,29,29,5,78,65,77,   7455
     69,32,32,58,255,19,29,29,29,29,29,29
2080 data 29,29,29,29,29,29,29,29,29,29,17,  7161
     5,80,85,78,75,84,69,58,32,32,32,32
2090 data 48,48,255,19,29,29,29,29,29,29,29, 7121
     29,29,29,29,29,29,29,29,29,29,17,29
2100 data 29,29,29,29,29,29,255,147,17,17,   7522
     17,17,17,17,17,17,17,17,17,17,29,29
2110 data 29,29,29,29,29,29,29,29,29,29,5,   7329
     71,32,65,32,77,32,69,32,32,32,79,32
2120 data 86,32,69,32,82,255,5,77,65,67,72,  7849
     32,68,73,67,72,32,66,69,82,69,73,84
2130 data 44,255,5,68,65,83,32,87,65,82,32,  7615
     87,79,72,76,32,78,73,67,72,84,83,44
2140 data 255,13,13,29,29,29,5,65,66,69,82,  7749
     32,68,85,32,72,65,83,84,32,74,65,32
2150 data 78,79,67,72,32,48,32,77,65,69,78,  7592
     78,67,72,69,78,32,33,255,13,13,29
2160 data 29,29,29,29,5,68,65,77,73,84,32,   8025
     83,67,72,69,73,68,69,83,84,32,68,85
2170 data 32,74,69,84,90,84,32,65,85,83,32,  7636
     33,33,255,147,17,17,17,17,17,17,17
2180 data 17,17,17,17,17,17,5,29,29,66,82,   7836
     65,86,79,44,68,65,83,32,66,73,76,68
2190 data 32,72,65,83,84,32,68,85,32,71,69,  7783
     83,67,72,65,70,70,84,32,33,33,33,255
2200 data 19,17,17,17,17,17,17,17,17,17,17,  8146
     17,17,17,17,17,17,17,17,17,17,17,17
2210 data 17,29,29,29,29,29,29,29,29,29,28,  7655
     40,32,68,82,85,69,67,75,69,32,70,69
2220 data 85,69,82,75,78,79,80,70,32,41,     7353
     255,19,17,17,17,17,17,17,17,17,17,17
2230 data 17,17,17,17,17,17,17,17,17,17,17,  7928
     17,17,29,29,29,29,29,29,29,29,29,29
2240 data 29,29,29,29,29,29,29,29,29,29,29,  7383
     29,29,29,5,68,82,85,69,67,75,69,32
2250 data 39,83,84,79,80,39,157,157,157,     8129
     157,157,157,157,157,157,157,157,157,
     157
2260 data 157,17,70,85,69,82,32,83,69,76,    7866
     66,83,84,77,79,82,68,255,147,29,29,29
2270 data 29,29,29,29,29,29,29,29,29,29,29,  7008
     29,158,17,17,65,85,83,87,69,82,84
2280 data 85,78,71,255,19,17,17,17,17,17,    7766
     17,17,17,17,17,17,17,17,17,17,17,17
2290 data 17,17,17,17,17,17,17,29,29,29,29,  7359
     158,40,70,49,61,69,68,73,84,79,82
2300 data 32,70,51,61,82,69,83,84,65,82,84,  7134
     32,70,53,61,69,78,68,69,41,255,1,6
2310 data 12,18,24,35,85,49,58,49,51,32,48,  6923
     48,32,49,56,32,48,49,66,45,80,32,49
2320 data 51,32,48,48,48,2,34,66,98,130,     7213
     162,194,226,32,32,32,32,32,32,32,32,32
2330 data 32,32,32,32,32,32,32,160,32,32,    7227
     32,32,32,32,32,32,32,32,32,32,32,32
2340 data 32,160,32,32,32,32,32,32,32,32,    7035
     32,32,32,32,32,32,32,160,32,32,32,32
2350 data 32,32,32,32,32,32,32,32,32,32,32,  7137
     160,32,32,32,32,32,32,32,32,32,32,32
2360 data 32,32,32,32,32,160,32,32,32,32,    7350
     32,32,32,32,32,32,32,32,32,32,32,160
2370 data 32,32,32,32,32,32,32,32,32,32,32,  7227
     32,32,32,32,160,32,32,32,32,32,32,32
2380 data 32,32,32,32,32,32,32,32,32,32,160, 7104
     32,32,32,32,32,32,32,32,32,32,32,32
2390 data 32,32,32,32,32,160,32,32,32,32,    7209
     32,32,32,32,32,32,32,32,32,32,32,32
2400 data 32,32,160,44,83,44,0,104,136,120,  7317
     136,136,136,152,136,168,136,184,136
2410 data 200,136,216,136,232,136,250,136,   7755
     0,176,168,178,80,181,248,183,160,186
2420 data 72,189,240,191,152,194,64,197,     6177
     232,199,144,202,56,205,10,28,20,18,14
2430 data 21,26,29,33,34,37,42,39,38,41,62,  7653
     22,60,65,66,67,68,69,70,71,72,73,74
2440 data 75,76,78,79,80,81,84,32,0,4,40,4,  7196
     80,4,120,4,160,4,200,4,240,4,24,5
```


WWW.HOMECOMPUTERWORLD.COM

```
2450 data 64,5,104,5,144,5,184,5,224,5,8,6,      6756
     48,6,88,6,128,6,168,6,208,6,248,6
2460 data 32,7,72,7,112,7,152,7,192,7,32,        8196
     65,67,69,75,76,79,81,0,66,68,70,74,78
2470 data 80,84,0,71,73,0,71,72,73,0,67,68,      7418
     32,69,75,76,78,79,80,81,87,0,148,137
2480 data 157,137,157,137,148,137,148,137,       9686
     165,137,168,137,157,137,168,137,157
2490 data 137,148,137,148,137,148,137,157,       7432
     137,148,137,157,137,148,137,148,137
2500 data 148,137,157,137,174,137,172,137,       8460
     20,38,1,2,4,8,16,32,64,128,44,128,87
2510 data 128,128,128,181,128,184,128,78,        7632
     129,242,129,245,129,45,130,25,131,124
2520 data 131,172,131,200,129,179,131,46,        8306
     132,91,132,199,132,8,133,127,133,139
2530 data 133,187,133,201,133,11,134,15,         7455
     134,22,134,31,134,57,134,90,134,116,
     134
2540 data 160,134,179,134,201,134,242,134,       7571
     26,135,80,135,137,135,231,135,5,136
2550 data 201,129,0,0,0,0,0,0,0,0,0,0,0,0,       7263
     0,1,1,3,3,7,7,15,15,31,31,63,12,6,6
2560 data 6,6,6,6,6,6,6,5,5,5,5,4,4,4,4,4,       8117
     4,4,4,9,8,7,6,5,5,5,5,4,4,4,4,4,5,5
2570 data 5,6,4,5,5,7,7,6,7,40,25,43,22,20,      7367
     21,31,40,50,30,21,40,197,206,206,206
2580 data 206,206,206,206,197,212,206,206,       7336
     206,206,206,206,197,212,206,206,206
2590 data 206,206,206,197,212,212,206,206,       7520
     206,206,206,197,212,212,206,206,206
2600 data 206,206,197,212,212,212,206,206,       7404
     206,206,197,212,212,212,206,206,206
2610 data 206,197,212,212,212,212,206,206,       7588
     206,197,212,212,212,212,206,206,206
2620 data 197,212,212,212,212,212,206,206,       7514
     197,212,212,212,212,212,206,206,197
2630 data 212,212,212,212,212,212,206,206,       7678
     207,208,209,210,227,228,229,230,214
2640 data 215,216,217,0,207,208,209,210,         6036
     206,228,229,230,230,215,216,217,217,
     199
2650 data 200,194,205,198,197,203,204,201,       7664
     192,193,195,196,243,244,245,246,247
2660 data 0,200,199,198,198,200,194,205,         6974
     205,205,194,194,194,196,244,245,246,
     247
2670 data 243,198,197,195,193,192,205,194,       7969
     201,203,204,199,200,196,0,195,195,193
2680 data 192,195,197,193,205,205,205,194,       7860
     194,194,198,197,201,204,203,194,205
2690 data 192,193,195,199,200,196,0,205,         7218
     205,204,203,201,197,204,194,194,194,
     205
2700 data 205,205,248,249,250,197,194,126,       7589
     125,0,0,0,0,123,119,0,0,119,0,0,0,0
2710 data 123,0,0,0,0,0,0,123,119,119,123,       7713
     126,125,0,0,123,119,0,0,123,119,0,0
2720 data 123,123,0,0,0,0,0,0,0,0,0,0,0,0,       6365
     0,123,126,119,0,0,0,0,0,0,123,119,0
2730 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,119,       6304
     119,20,36,52,68,84,100,116,132,30,60
2740 data 90,120,150,180,210,240,3,12,8,4,       5812
     0,12,15,12,14,4,0,0,0,13,2,12,0,0,0
2750 data 8,56,142,60,142,64,142,68,142,72,      7056
     142,76,142,80,142,84,142,27,51,77
2760 data 102,126,153,178,205,255,211,218,       7280
     212,219,213,220,0,218,211,219,212,220
2770 data 213,211,212,213,218,219,220,221,       7367
     223,224,0,221,221,221,221,221,221,223
2780 data 224,223,211,212,213,218,219,220,       7425
     222,225,226,0,222,222,222,222,222,222
2790 data 225,226,225,223,224,221,0,221,         7182
     221,212,225,226,222,0,222,222,212,48
2800 data 140,52,140,41,140,45,140,22,140,       7213
     32,140,3,140,13,140,246,139,253,139
2810 data 238,138,252,138,73,139,87,139,46,      7420
     139,60,139,9,139,28,139,32,21,155
2820 data 32,195,154,169,0,141,100,140,133,      6644
     2,141,21,208,169,4,32,10,150,32,216
2830 data 150,169,5,32,10,150,32,233,150,        7178
     201,140,208,3,76,130,144,201,134,208
2840 data 3,76,150,144,201,133,208,235,32,       7231
     246,150,169,0,32,10,150,32,211,150
2850 data 32,233,150,201,140,240,188,201,        6441
     49,144,245,201,58,176,241,32,210,255
2860 data 162,48,142,139,128,56,233,49,141,      7149
     91,140,169,255,141,92,140,238,92,140
2870 data 238,139,128,169,2,32,10,150,32,        7664
     74,150,173,92,140,205,91,140,144,234
2880 data 169,1,32,10,150,32,59,150,144,         6939
     182,76,0,160,169,7,32,10,150,32,59,150
2890 data 176,3,76,6,144,169,255,133,2,76,       7179
     226,252,162,0,142,104,140,142,112,140
2900 data 142,3,142,232,142,113,140,32,246,      6746
     150,169,8,32,10,150,169,0,32,255,150
2910 data 169,1,141,102,140,169,4,141,103,       6724
     140,32,179,154,32,108,152,32,40,152
2920 data 169,1,141,21,208,173,141,2,166,        7534
     197,41,1,208,13,224,7,240,35,224,2,240
2930 data 19,32,127,145,208,11,224,7,240,        6681
     16,224,2,208,243,32,0,145,76,144,145
2940 data 32,38,145,76,144,145,32,77,145,        7177
     76,144,145,32,99,145,76,144,145,173
2950 data 102,140,201,1,240,115,201,30,240,      6795
     15,173,0,208,56,233,8,141,0,208,206
2960 data 102,140,76,127,145,173,16,208,41,      7232
     254,141,16,208,169,248,208,235,173
2970 data 102,140,205,229,137,176,76,201,        7152
     29,240,15,173,0,208,24,105,8,141,0,208
2980 data 238,102,140,76,127,145,173,16,         7146
     208,9,1,141,16,208,169,1,208,235,173
2990 data 103,140,201,4,240,38,173,1,208,        6735
     56,233,8,141,1,208,206,103,140,76,127
3000 data 145,173,103,140,205,228,137,240,       7718
     15,173,1,208,24,105,8,141,1,208,238
3010 data 103,140,76,127,145,169,17,141,4,       6979
     212,173,3,142,74,176,10,169,250,32
3020 data 218,150,169,16,141,4,212,96,166,       7575
     197,173,141,2,201,4,208,14,224,22,208
3030 data 3,76,45,146,224,20,208,3,76,41,        7438
     146,138,162,0,221,62,137,208,3,76,153
3040 data 146,232,224,18,144,243,201,3,208,      7097
     3,76,0,144,201,51,208,37,173,141,2
3050 data 201,1,208,14,169,1,32,255,150,32,      6509
     27,152,32,40,152,76,202,144,169,1
3060 data 141,102,140,169,4,141,103,140,32,      7117
     179,154,76,202,144,201,40,208,14,32
3070 data 194,147,169,0,32,255,150,32,108,       7337
     152,76,202,144,201,43,208,6,32,242
3080 data 147,76,238,145,201,4,208,3,76,31,      7299
     148,201,5,208,3,76,74,155,201,1,208
3090 data 17,162,1,142,102,140,202,142,16,       7256
     208,169,24,141,0,208,32,99,145,76,202
3100 data 144,169,128,48,2,165,0,133,2,169,      6811
     0,141,21,208,173,104,140,141,204,142
3110 data 165,197,201,40,240,14,201,43,240,      7462
     29,201,1,240,31,201,63,208,238,240
3120 data 47,32,194,147,169,0,32,255,150,        6836
     32,108,152,169,248,32,218,150,76,60
3130 data 146,32,242,147,76,83,146,173,204,      7981
     142,32,122,152,165,167,133,179,165
3140 data 168,133,180,169,130,32,255,150,        7599
     32,108,152,169,1,141,21,208,76,202,144
3150 data 104,72,201,65,240,4,201,78,208,3,      7047
     76,15,147,173,103,140,208,11,189,80
3160 data 137,72,173,103,140,201,4,240,227,      7056
     10,170,189,98,137,133,171,133,177
3170 data 189,99,137,133,172,24,105,212,         7279
     133,178,32,119,152,165,167,24,101,171
3180 data 133,167,165,168,101,172,133,168,       7720
     165,167,56,233,160,133,167,165,168
3190 data 233,4,133,168,172,102,140,104,72,      7217
     201,69,240,54,201,75,240,18,201,81
3200 data 240,114,208,79,104,145,167,145,        7543
     171,169,8,145,177,76,202,144,177,171
3210 data 201,75,240,22,201,69,208,3,32,         6993
     127,147,201,81,208,5,169,0,141,106,140
3220 data 32,140,147,76,135,152,104,76,202,      7302
     144,177,171,201,69,240,246,201,75
3230 data 208,3,32,168,147,201,81,208,5,         8046
     169,0,141,106,140,173,107,140,201,7,
     176
3240 data 223,32,114,147,76,135,152,177,         7387
     171,201,69,240,15,201,75,240,17,201,81
3250 data 208,239,169,0,141,106,140,240,         7474
     232,32,127,147,76,51,147,32,168,147,76
3260 data 135,152,173,106,140,208,179,238,       7852
     106,140,177,171,201,69,208,3,32,127
3270 data 147,201,75,208,3,32,168,147,76,        7265
```


WWW.HOMECOMPUTERWORLD.COM

```
      135,152,238,107,140,169,1,24,109,111
3280 data 140,141,111,140,96,206,107,140,      6386
      173,111,140,56,233,1,141,111,140,96
3290 data 238,105,140,162,2,189,108,140,       6989
      201,9,240,7,24,105,1,157,108,140,96,
      169
3300 data 0,157,108,140,202,16,234,96,206,     6864
      105,140,162,2,189,108,140,240,7,56
3310 data 233,1,157,108,140,96,169,9,157,      6972
      108,140,202,16,236,96,173,104,140,201
3320 data 11,144,13,162,0,142,104,140,142,     6164
      112,140,232,142,113,140,96,238,104
3330 data 140,162,1,189,112,140,201,9,240,     6589
      7,24,105,1,157,112,140,96,169,0,157
3340 data 112,140,202,16,234,96,173,104,       7341
      140,208,16,169,11,141,104,140,162,1,
      142
3350 data 112,140,232,142,113,140,208,22,      6466
      206,104,140,248,162,1,189,112,140,56
3360 data 233,1,157,112,140,41,240,240,3,      7303
      202,16,240,216,96,169,9,32,10,150,169
3370 data 12,32,10,150,169,0,141,21,208,       6955
      133,198,32,233,150,201,136,208,3,76,
      168
3380 data 144,201,134,240,38,201,135,240,      6002
      48,201,133,208,234,169,0,32,141,152
3390 data 162,2,32,198,255,169,8,32,88,148,    7677
      76,168,144,32,255,150,32,204,255,169
3400 data 2,32,195,255,96,169,128,32,141,      7134
      152,162,2,32,201,255,169,16,208,224
3410 data 169,8,32,10,150,169,13,32,10,150,    7845
      162,255,232,189,72,136,157,117,140
3420 dAta 224,22,144,245,169,15,162,8,168,     7145
      32,186,255,169,0,32,189,255,32,192
3430 data 255,169,13,162,8,168,32,186,255,     8089
      169,1,162,71,160,136,32,189,255,32
3440 data 192,255,169,0,141,115,140,169,32,    7667
      32,255,150,162,15,32,201,255,162,255
3450 data 232,189,117,140,32,210,255,224,      6745
      13,144,245,32,204,255,32,224,152,162
3460 data 13,32,198,255,32,207,255,141,141,    7559
      140,32,207,255,141,142,140,174,115
3470 data 140,189,96,136,224,4,144,18,224,     6822
      7,240,7,56,233,100,162,49,208,9,56
3480 data 233,200,162,50,208,2,162,48,142,     6901
      138,140,32,23,153,142,139,140,141,140
3490 data 140,32,204,255,162,15,32,201,255,    7355
      162,255,232,189,131,140,32,210,255
3500 data 224,9,144,245,32,204,255,169,253,    7879
      141,93,140,162,13,32,198,255,32,207
3510 data 255,41,3,201,1,240,6,32,204,255,     7466
      76,202,149,162,0,238,93,140,32,207
3520 data 255,201,160,240,8,157,248,136,       7081
      232,224,18,144,238,169,0,157,248,136
3530 data 32,204,255,169,14,32,10,150,162,     7324
      255,232,189,250,136,240,7,32,210,255
3540 data 224,16,144,243,169,0,141,114,140,    7469
      32,167,152,162,2,32,198,255,169,64
3550 data 32,88,148,32,27,152,169,4,32,9,      6877
      152,32,40,152,169,0,141,101,140,32,233
3560 data 150,201,133,240,54,201,134,240,      6796
      24,201,136,240,84,201,43,208,6,32,194
3570 data 147,76,133,149,201,45,208,227,32,    7683
      242,147,76,133,149,173,101,140,48
3580 data 216,169,17,141,4,212,169,128,141,    7369
      101,140,32,255,150,169,16,141,4,212
3590 data 76,141,149,169,0,141,101,140,173,    7165
      115,140,201,7,176,6,238,115,140,76
3600 data 224,148,173,141,140,240,15,173,      7000
      142,140,32,23,153,142,129,140,141,130
3610 data 140,76,173,148,169,13,32,195,255,    7586
      169,15,32,195,255,76,168,144,120,141
3620 data 20,3,142,21,3,96,169,16,76,10,       7647
      150,10,170,189,238,137,133,173,189,239
3630 data 137,133,174,160,255,200,177,173,     8183
      201,255,240,6,32,210,255,76,24,150
3640 data 96,32,253,149,140,18,208,173,17,     6894
      208,41,127,141,17,208,169,129,141,26
3650 data 208,88,96,32,233,150,201,74,240,     7425
      6,201,78,208,245,24,96,56,96,173,92
3660 data 140,10,170,189,18,137,133,179,       7072
      189,19,137,133,180,169,0,141,93,140,32
3670 data 211,150,32,228,255,141,94,140,       6245
      201,13,240,62,201,19,240,76,201,20,240
3680 data 29,201,31,144,234,201,97,176,230,    6796
      32,210,255,172,93,140,145,179,238
3690 data 93,140,172,93,140,192,16,208,209,    6911
      136,16,36,173,93,140,240,204,206,93
3700 data 140,169,46,32,210,255,169,157,32,    7666
      210,255,169,157,32,210,255,208,181
3710 data 172,93,140,240,179,136,169,46,32,    7305
      210,255,177,179,9,128,145,179,96,169
3720 data 3,32,10,150,173,92,140,201,9,240,    6918
      4,169,2,208,2,169,10,32,10,150,76
3730 data 89,150,169,6,76,10,150,169,170,      6754
      162,0,160,0,32,219,255,32,222,255,201
3740 data 0,208,249,96,169,0,133,198,165,      6340
      197,201,64,240,250,76,228,255,169,1
3750 data 141,32,208,141,33,208,96,141,94,     7187
      140,32,119,152,169,160,162,4,133,169
3760 data 134,170,169,160,162,216,133,177,     8126
      134,178,160,0,173,94,140,208,3,76,211
3770 data 151,201,255,208,39,177,169,201,      6925
      74,240,15,201,84,240,15,201,76,240,15
3780 data 201,77,240,15,76,219,151,169,84,     7634
      208,10,169,74,208,6,169,77,208,2,169
3790 data 76,145,169,208,235,201,129,208,3,    6167
      76,36,153,201,130,208,19,177,167,72
3800 data 177,179,145,167,145,169,104,166,     8221
      2,48,2,145,179,76,215,151,74,176,44
3810 data 74,176,49,74,176,62,74,176,88,74,    7837
      176,25,74,176,16,74,176,6,177,169
3820 data 145,167,208,88,32,207,255,145,       7678
      169,208,77,169,32,145,169,208,75,177
3830 data 167,32,210,255,208,68,169,32,145,    7570
      167,145,169,208,60,177,169,201,81
3840 data 240,4,201,69,208,4,169,32,145,       7111
      169,208,44,177,167,201,69,240,13,201
3850 data 75,240,14,201,81,208,30,238,106,     6721
      140,208,25,32,114,147,208,20,32,140
3860 data 147,208,15,32,207,255,145,167,       8020
      208,8,177,167,145,169,169,8,145,177,
      165
3870 data 170,201,7,208,6,165,169,201,70,      7628
      240,64,230,167,208,2,230,168,230,169
3880 data 208,2,230,170,230,177,208,2,230,     7392
      178,173,94,140,201,130,208,6,230,179
3890 data 208,2,230,180,76,21,151,141,94,      7074
      140,169,160,162,4,133,167,133,169,134
3900 data 168,134,170,76,21,151,162,255,       7781
      169,0,232,157,105,140,224,6,208,248,96
3910 data 162,2,189,108,140,41,15,157,108,     6942
      140,24,105,48,157,120,7,169,0,157,120
3920 data 219,202,16,234,173,111,140,41,15,    6751
      141,111,140,24,105,48,141,135,7,169
3930 data 0,141,135,219,162,1,189,112,140,     6971
      41,15,157,112,140,24,105,48,157,149
3940 data 7,169,0,157,149,219,202,16,234,      7274
      96,32,27,152,169,4,32,255,150,76,40
3950 data 152,173,104,140,10,170,189,38,       7160
      137,133,167,189,39,137,133,168,96,32
3960 data 40,152,76,231,146,141,114,140,       6535
      169,10,32,10,150,169,9,141,92,140,32
3970 data 74,150,172,93,140,136,177,179,41,    6863
      127,145,179,173,114,140,240,4,169
3980 data 87,208,2,169,82,141,17,137,162,      6770
      255,172,93,140,136,200,232,189,14,137
3990 data 153,250,136,224,3,208,244,200,       6948
      140,93,140,169,2,162,8,168,32,186,255
4000 data 173,93,140,162,250,160,136,32,       7080
      189,255,32,192,255,169,0,133,144,169
4010 data 11,32,10,150,169,8,32,180,255,       7158
      169,111,32,150,255,32,165,255,201,48
4020 data 240,26,32,165,255,32,210,255,201,    7177
      13,208,246,32,171,255,169,200,32,218
4030 data 150,104,104,32,91,148,76,168,144,    7265
      76,171,255,162,48,56,233,10,144,3
4040 data 232,176,248,105,58,96,177,169,       7852
      141,4,142,201,32,208,2,169,65,32,57,
      154
4050 data 32,73,154,32,41,154,144,45,104,      6314
      104,104,104,169,8,32,10,150,169,0,32
4060 data 255,150,32,122,145,173,17,208,9,     7021
      16,141,17,208,169,38,141,229,137,173
4070 data 3,142,41,254,141,3,142,76,191,       6847
      144,162,42,208,24,32,77,154,173,4,142
4080 data 201,69,240,12,201,75,240,238,201,    6876
      76,240,234,201,81,208,21,162,40,32
```


WWW.HOMECOMPUTERWORLD.COM

```
4090 data 62,154,32,81,154,32,41,154,176,5,      6668
     32,85,154,208,171,32,85,154,173,4
4100 data 142,201,69,240,38,201,81,240,34,       6589
     173,102,140,201,39,144,21,169,0,141
4110 data 16,208,169,16,141,0,208,169,0,         7213
     141,102,140,32,99,145,76,219,151,32,38
4120 data 145,76,219,151,173,103,140,10,10,      7224
     10,24,105,48,56,32,131,154,169,16
4130 data 141,116,140,173,102,140,201,30,        6912
     176,37,10,10,10,24,109,116,140,24,32
4140 data 131,154,173,4,142,201,81,240,9,        7290
     238,97,140,238,98,140,76,157,153,173
4150 data 98,140,9,128,141,98,140,48,243,        7003
     56,233,30,72,173,4,142,201,81,208,4
4160 data 162,0,240,3,174,97,140,189,230,        6543
     137,141,99,140,174,104,140,189,247,141
4170 data 13,99,140,157,247,141,169,0,141,       6614
     116,140,104,16,175,160,255,200,177
4180 data 175,240,7,205,143,140,208,246,24,      6920
     96,56,96,56,233,65,10,170,189,184
4190 data 137,133,175,189,185,137,133,176,       8252
     96,169,1,208,29,169,1,208,6,169,40
4200 data 208,21,169,40,141,144,140,165,         7529
     169,56,237,144,140,133,169,165,170,233
4210 data 0,133,170,208,17,141,144,140,165,      7402
     169,24,109,144,140,133,169,165,170
4220 data 105,0,133,170,160,0,177,169,141,       7164
     143,140,96,72,8,173,104,140,10,10,10
4230 data 10,141,138,140,173,4,142,201,81,       6702
     208,4,169,0,240,4,173,97,140,10,24
4240 data 109,138,140,141,138,140,40,144,3,      7518
     238,138,140,174,138,140,104,157,55
4250 data 141,96,169,24,141,0,208,169,0,         7300
     141,16,208,169,88,141,1,208,96,169,15
4260 data 141,24,212,169,9,141,5,212,141,1,      6551
     212,169,200,141,0,212,169,16,141,4
4270 data 212,234,234,234,234,234,169,25,        7411
     141,24,208,173,22,208,9,16,141,22,208
4280 data 162,1,142,28,208,202,142,21,208,       7594
     142,37,208,162,13,142,35,208,232,142
4290 data 32,208,142,33,208,142,39,208,232,      7272
     142,34,208,169,54,133,1,169,202,141
4300 data 248,7,96,32,195,154,169,0,162,80,      7502
     133,167,134,168,169,0,162,176,133
4310 data 169,134,170,162,32,160,0,177,167,      7816
     145,169,200,208,249,230,168,230,170
4320 data 202,208,242,162,255,232,189,72,        7592
     138,157,145,140,224,165,208,245,76,66
4330 data 157,169,0,162,255,232,157,145,         7486
     140,224,12,144,248,141,21,208,169,48
4340 data 141,19,133,141,20,133,32,61,157,       7386
     169,15,32,10,150,169,1,32,126,156,141
4350 data 157,140,169,16,32,19,157,162,255,      7332
     232,157,158,140,224,8,144,248,169
4360 data 240,32,218,150,32,66,157,173,20,       7082
     133,201,57,144,10,169,47,141,20,133
4370 data 169,49,141,19,133,238,20,133,32,       7177
     27,152,169,4,32,255,150,173,107,140
4380 data 174,104,140,157,191,140,173,105,       7633
     140,157,203,140,162,255,232,189,108
4390 data 140,24,105,48,224,3,176,5,157,43,      7347
     133,208,240,141,69,133,32,61,157,173
4400 data 17,208,9,16,141,17,208,169,17,32,      7075
     19,157,174,104,140,157,167,140,169
4410 data 18,32,19,157,174,104,140,157,179,      7565
     140,169,19,32,10,150,169,48,141,189
4420 data 133,169,0,141,143,140,238,189,         6468
     133,169,20,32,10,150,32,211,150,32,233
4430 data 150,201,134,208,3,76,165,156,201,      7268
     135,208,3,76,156,156,201,133,208,3
4440 data 76,29,157,201,70,240,32,201,71,        6495
     208,224,174,143,140,189,230,137,141
4450 data 144,140,174,104,140,189,145,140,       6794
     13,144,140,157,145,140,169,24,72,160
4460 data 212,208,5,169,23,72,160,206,173,       7063
     104,140,10,10,10,141,129,140,174,143
4470 data 140,232,138,24,109,129,140,170,        7005
     152,157,215,140,104,32,10,150,238,143
4480 data 140,173,143,140,205,107,140,176,       7770
     3,76,248,155,238,104,140,173,104,140
4490 data 205,157,140,176,3,76,136,155,76,       7235
     50,144,141,4,142,169,32,32,210,255
4500 data 162,0,142,8,142,169,157,32,210,        7004
     255,32,211,150,32,233,150,201,135,208
4510 data 5,104,104,76,150,144,201,134,208,      7041
     5,104,104,76,74,155,201,133,208,3
4520 data 76,27,157,201,48,144,224,201,58,       6656
     176,220,32,210,255,174,8,142,157,5
4530 data 142,160,0,173,4,142,240,32,174,8,      6479
     142,224,1,208,20,32,12,157,240,12
4540 data 201,2,176,38,169,10,141,7,142,         6887
     200,208,18,200,208,5,238,8,142,208,169
4550 data 32,12,157,240,25,169,0,141,7,142,      6783
     32,12,157,24,109,7,142,201,13,176
4560 data 1,96,169,22,32,10,150,76,134,156,      7230
     192,1,240,244,76,134,156,185,5,142
4570 data 56,233,48,96,32,10,150,169,0,76,       7209
     126,156,104,104,173,104,140,208,3,76
4580 data 74,155,206,104,140,173,20,133,         6678
     201,48,208,8,141,19,133,169,58,141,20
4590 data 133,206,20,133,76,200,155,169,21,      6607
     76,10,150,173,17,208,41,239,141,17
4600 data 208,162,1,142,21,208,142,113,140,      6838
     202,142,112,140,142,104,140,138,202
4610 data 232,157,247,141,224,12,144,248,        6805
     173,3,142,9,1,141,3,142,238,229,137
4620 data 169,0,32,255,150,162,0,142,102,        6519
     140,142,98,140,232,142,97,140,169,4
4630 data 141,103,140,169,16,32,179,154,         7509
     169,129,32,255,150,173,98,140,201,129
4640 data 144,44,32,214,147,173,104,140,         6514
     205,157,140,208,206,169,0,141,104,140
4650 data 141,21,208,173,3,142,41,254,141,       7281
     3,142,206,229,137,169,147,32,210,255
4660 data 173,17,208,9,16,141,17,208,96,76,      7297
     58,153,173,25,208,141,25,208,41,1
4670 data 208,3,76,89,158,174,18,208,173,        7318
     93,140,74,74,74,74,144,10,224,0,144
4680 data 3,76,80,159,76,89,158,224,222,         7295
     144,3,76,197,158,224,68,144,3,76,255
4690 data 158,169,68,141,18,208,169,6,141,       6626
     33,208,141,32,208,169,0,162,2,160,1
4700 data 32,112,158,173,11,221,160,3,162,       6902
     255,136,232,224,2,240,68,224,5,240
4710 data 64,224,6,240,11,32,171,158,74,74,      7324
     74,74,32,99,158,232,32,171,158,41
4720 data 15,32,99,158,224,6,208,219,162,        6669
     255,160,0,232,192,15,176,15,189,185
4730 data 139,153,0,208,200,169,49,153,0,        6236
     208,200,208,236,169,127,141,21,208,169
4740 data 0,141,16,208,76,126,234,169,241,       7270
     157,248,7,208,176,141,95,140,169,231
4750 data 24,109,95,140,157,248,7,96,141,        6532
     37,208,142,38,208,192,128,176,7,152
4760 data 160,1,162,255,48,18,174,111,140,       7260
     189,145,140,141,99,140,169,1,141,39
4770 data 208,160,0,162,0,232,192,1,240,11,      6832
     78,99,140,144,4,169,3,208,2,169,7
4780 data 157,39,208,224,8,144,233,96,173,       7576
     93,140,10,10,10,144,7,173,219,142,192
4790 data 0,240,10,185,8,221,192,0,208,3,        6551
     141,219,142,96,169,0,141,18,208,169
4800 data 6,141,33,208,141,32,208,169,10,        6624
     162,0,160,7,32,112,158,173,108,140,141
4810 data 21,208,162,255,160,0,232,189,193,      6976
     139,153,0,208,200,169,231,153,0,208
4820 data 200,169,242,157,248,7,224,7,144,       7959
     233,76,84,158,169,222,141,18,208,169
4830 data 14,141,33,208,169,6,141,32,208,        6403
     169,10,162,0,160,128,32,112,158,160
4840 data 255,162,255,232,200,189,161,142,       7325
     157,0,208,232,189,161,142,157,0,208
4850 data 185,178,142,153,248,7,192,7,144,       8154
     231,173,177,142,141,16,208,162,1,173
4860 data 93,140,74,74,74,176,6,32,135,234,      7570
     174,96,140,142,21,208,76,89,158,169
4870 data 0,141,18,208,169,0,141,32,208,         6466
     141,33,208,141,21,208,76,123,234
```

„RUNMAN Construction Set": Starterteil

WWW.HOMECOMPUTERWORLD.COM

```
1000 rem --------------------------------
1010 rem   runman construction set v2.0
1020 rem
1030 rem      by christian drapatz '86
1040 rem
1050 rem   programmteil 2 : game
1060 rem --------------------------------
1070 :
1080 a=0:a$=chr$ (34)                                  1541
1090 print "[clr,cyn,2down,10right]***                 3457
      please wait ***"
1100 read da:poke 40960+a,da                           1648
1110 if a<3744 then a=a+1:goto 1100                    2034
1120 print "[2down,blk]load"a$"rcs c"a$",8,              2517
     1[3down]"
1130 print "[down]run[8up]                              1486
1140 poke 631,13:poke 632,13                           2109
1150 poke 198,2:end                                    1140
1160 :
1170 data 32,1,172,169,196,162,157,160,0,              7074
     32,38,150,169,0,141,104,140,141,92,140
1180 data 162,255,142,28,208,232,157,65,               6895
     142,224,129,208,248,169,0,141,141,140
1190 data 162,255,232,157,131,140,224,8,               7064
     144,248,173,14,221,9,128,141,14,221,
     162
1200 data 255,232,189,131,140,201,1,240,5,             7853
     169,0,157,131,140,224,8,144,239,174
1210 data 104,140,189,167,140,141,100,140,             7887
     189,179,140,141,102,140,160,255,189
1220 data 191,140,201,7,240,7,170,232,189,             7893
     230,137,168,136,140,96,140,169,23,32
1230 data 6,172,169,29,32,173,171,32,249,              7667
     171,169,3,32,103,169,120,173,17,208
1240 data 41,239,141,17,208,174,92,140,173,            6997
     104,140,157,186,142,189,158,140,201
1250 data 6,144,2,169,6,170,189,230,137,               7005
     168,136,140,108,140,173,104,140,10,10
1260 data 10,10,170,160,0,189,55,141,153,              6772
     161,142,232,200,192,16,144,244,174,104
1270 data 140,189,247,141,141,177,142,169,             7727
     0,170,157,197,142,232,224,8,144,248
1280 data 174,104,140,189,203,140,141,107,             6701
     140,138,10,10,10,170,160,0,140,207
1290 data 142,132,171,189,215,140,153,178,             7489
     142,169,123,153,9,142,232,200,192,8
1300 data 144,239,169,255,141,111,140,238,             7855
     111,140,32,85,166,138,174,104,140,221
1310 data 191,140,208,241,32,68,229,169,25,            7144
     32,173,171,169,26,32,10,150,32,86
1320 data 165,162,255,232,224,8,176,4,169,             7771
     2,208,2,169,5,157,17,142,224,15,144
1330 data 238,169,0,141,93,140,32,255,150,             7916
     169,2,32,255,150,173,17,208,9,16,141
1340 data 17,208,169,35,32,10,150,160,0,32,            6706
     41,150,162,1,160,34,24,32,240,255
1350 data 169,30,32,210,255,174,104,140,               6954
     232,138,32,15,172,169,47,32,210,255,
     173
1360 data 157,140,32,15,172,32,120,165,173,            7206
     30,208,74,176,250,169,0,141,30,208
1370 data 173,30,208,74,176,38,32,173,165,             7353
     32,144,170,173,107,140,240,91,169,6
1380 data 141,105,140,206,105,140,169,1,32,            7196
     163,165,173,105,140,208,243,165,203
1390 data 201,63,240,3,76,115,161,169,125,             8018
     141,9,142,169,243,32,158,164,174,92
1400 data 140,222,158,140,189,158,140,170,             6984
     24,105,48,141,228,134,138,208,12,169
1410 data 32,32,213,161,238,141,140,169,1,             6806
     208,43,169,31,32,213,161,76,254,161
1420 data 72,169,20,32,6,172,169,30,32,173,            7591
     171,104,76,241,171,169,125,141,9,142
1430 data 169,196,32,158,164,32,1,172,169,             6920
     33,32,241,171,169,128,174,92,140,157
1440 data 131,140,169,255,141,142,140,238,             7067
     142,140,173,142,140,205,91,140,240
1450 data 2,176,37,173,92,140,205,91,140,              7048
     144,5,169,255,141,92,140,238,92,140
1460 data 173,141,140,205,91,140,240,2,176,            6594
     43,174,92,140,189,131,140,208,209
1470 data 76,109,160,238,104,140,173,104,              5867
     140,205,157,140,176,21,169,255,141,92
1480 data 140,238,92,140,174,92,140,189,               7230
     131,140,240,2,16,243,76,56,160,32,1,
     172
1490 data 169,28,32,241,171,169,240,141,26,            7014
     208,169,49,162,234,32,253,149,88,169
1500 data 36,32,10,150,169,1,141,134,2,169,            6282
     0,162,255,232,157,108,140,224,8,208
1510 data 248,169,255,141,92,140,169,1,141,            7568
     208,142,238,92,140,169,255,141,103
1520 data 140,238,103,140,174,103,140,189,             7213
     108,140,48,15,173,208,142,208,79,189
1530 data 186,142,205,107,140,240,12,176,              7129
     69,173,103,140,205,91,140,144,222,176
1540 data 123,189,158,140,205,118,140,240,             7514
     4,176,49,144,234,32,42,163,162,3,200
1550 data 200,200,185,125,142,221,119,140,             7785
     240,4,144,216,176,27,136,202,16,240
1560 data 32,42,163,162,2,200,200,185,90,              6330
     142,221,104,140,240,4,144,6,176,191
1570 data 136,202,16,240,174,103,140,142,              6106
     117,140,189,186,142,141,107,140,189
1580 data 158,140,141,118,140,162,0,142,               7579
     208,142,32,42,163,185,125,142,157,119
1590 data 140,232,200,224,4,208,244,32,42,             7115
     163,162,0,185,90,142,157,104,140,200
1600 data 232,224,3,208,244,240,130,173,               6428
     103,140,10,10,168,96,174,117,140,169
1610 data 128,157,108,140,172,92,140,138,              7491
     153,94,140,204,91,140,176,8,76,133,162
1620 data 206,142,140,16,29,169,7,24,109,              6709
     91,140,141,142,140,201,7,208,3,206,142
1630 data 140,173,91,140,170,24,105,49,141,            7065
     146,140,142,144,140,189,94,140,141
1640 data 143,140,32,246,163,173,146,140,              6931
     32,210,255,169,46,32,210,255,32,11,164
1650 data 32,255,163,32,16,164,169,255,32,             7350
     163,165,169,255,32,163,165,206,142
1660 data 140,206,146,140,206,144,140,174,             6908
     144,140,240,169,16,201,238,142,140
1670 data 169,37,32,10,150,169,1,141,134,2,            7375
     169,255,141,103,140,162,70,202,208
1680 data 253,206,103,140,208,29,169,255,              7025
     141,103,140,173,134,2,201,1,240,4,169
1690 data 1,208,2,169,0,141,134,2,160,3,32,            7495
     248,163,32,255,163,165,203,201,4,240
1700 data 13,201,5,240,15,201,6,208,203,               7554
     133,2,76,226,252,32,195,154,76,150,144
1710 data 76,0,144,160,0,174,142,140,24,76,            7139
     240,255,173,143,140,32,158,171,76
1720 data 179,171,32,11,164,169,32,76,210,             7421
     255,160,21,32,248,163,174,143,140,254
1730 data 186,142,189,186,142,32,23,153,               7325
     168,138,32,210,255,152,32,210,255,32
1740 data 11,164,174,143,140,189,158,140,              6726
     32,152,164,32,11,164,173,143,140,10
1750 data 10,168,200,200,200,162,255,136,              6596
     232,224,2,240,71,224,5,240,67,224,6
1760 data 240,11,185,89,142,74,74,74,74,32,            7220
     152,164,232,185,89,142,41,15,32,152
1770 data 164,224,6,208,219,32,11,164,173,             7182
     143,140,10,10,168,162,0,185,125,142
1780 data 72,200,232,224,4,208,246,162,0,              6485
     104,32,152,164,232,224,4,208,247,169
1790 data 48,32,210,255,76,210,255,169,58,             6742
     32,210,255,208,173,24,105,48,76,210
1800 data 255,141,178,142,32,114,170,173,              6602
     167,140,141,129,140,169,4,141,167,140
1810 data 169,7,32,90,169,174,104,140,189,             7782
     145,140,141,105,140,162,0,232,78,105
1820 data 140,176,4,169,227,208,2,169,214,             7452
     157,178,142,236,191,140,144,236,169
1830 data 255,141,111,140,238,111,140,173,             7776
     111,140,240,43,173,93,140,74,74,74
1840 data 176,25,32,170,169,174,191,140,               6736
     189,178,142,201,217,240,4,201,230,208
1850 data 8,169,2,32,90,169,76,211,164,173,            7271
     111,140,205,191,140,240,202,208,205
1860 data 169,4,32,240,167,169,45,32,163,              7418
     165,174,162,142,173,93,140,74,74,74
1870 data 144,15,224,200,144,231,173,129,              6638
     140,141,167,140,169,7,76,103,169,224
1880 data 200,144,165,176,239,173,92,140,              7648
     10,10,170,96,32,53,165,160,0,254,125
1890 data 142,189,125,142,201,10,208,11,               7279
     169,0,157,125,142,232,200,192,4,208,
     235
```


WWW.HOMECOMPUTERWORLD.COM

```
1900 data 169,27,32,10,150,169,1,141,134,2,      7297
     32,53,165,232,232,232,160,0,189,125
1910 data 142,24,105,48,32,210,255,202,200,      6929
     192,4,144,241,96,173,15,221,41,127
1920 data 141,15,221,32,53,165,169,0,141,        6635
     11,221,189,91,142,141,10,221,189,90
1930 data 142,141,9,221,189,89,142,141,8,        6727
     221,173,93,140,41,223,141,93,140,96
1940 data 170,160,0,200,208,253,202,208,         5734
     250,96,169,0,141,111,140,206,100,140
1950 data 208,74,174,104,140,189,167,140,        6349
     141,100,140,173,9,142,201,111,240,12
1960 data 165,171,240,8,198,171,173,9,142,       7084
     76,221,165,169,3,133,171,173,0,220
1970 data 141,9,142,201,126,240,43,201,125,      6480
     240,42,201,123,240,41,201,119,240
1980 data 40,201,111,240,39,169,0,141,101,       6959
     140,173,178,142,201,196,144,5,201,201
1990 data 176,5,96,169,194,208,2,169,205,        7633
     141,178,142,96,76,117,167,76,192,167
2000 data 76,15,168,76,139,168,76,7,169,         7227
     174,111,140,173,177,142,74,202,16,252
2010 data 169,0,42,96,173,111,140,10,170,        7378
     189,161,142,96,169,1,141,112,140,173
2020 data 111,140,10,10,24,109,112,140,168,      7256
     96,174,111,140,189,230,137,141,112
2030 data 140,173,177,142,77,112,140,141,        7134
     177,142,96,32,27,166,141,112,140,32
2040 data 41,166,56,233,16,141,113,140,173,      6825
     112,140,233,0,141,112,140,160,2,173
2050 data 112,140,74,141,112,140,173,113,        6330
     140,106,141,113,140,136,16,239,173,113
2060 data 140,174,111,140,157,73,142,173,        7239
     111,140,10,170,232,189,161,142,56,233
2070 data 48,74,74,74,174,111,140,157,81,        7356
     142,160,0,140,112,140,174,111,140,189
2080 data 81,142,170,232,138,10,170,189,98,      7481
     137,133,167,189,99,137,133,168,174
2090 data 111,140,189,73,142,168,177,167,        7465
     72,32,55,166,104,153,33,142,174,111
2100 data 140,96,201,32,208,4,169,0,240,7,       6985
     56,233,65,170,189,201,139,174,111,140
2110 data 157,65,142,96,174,111,140,189,9,       7064
     142,201,126,240,24,201,125,240,10,201
2120 data 123,240,36,201,119,32,14,167,96,       6723
     254,81,142,32,34,167,222,81,142,96
2130 data 222,81,142,32,34,167,254,81,142,       7437
     96,254,73,142,32,34,167,222,73,142
2140 data 96,222,73,142,32,34,167,254,73,        7626
     142,96,160,1,76,159,166,174,111,140
2150 data 189,9,142,168,189,17,142,201,2,        7119
     208,16,189,25,142,201,5,208,34,32,85
2160 data 166,32,204,166,76,226,166,192,         7392
     123,240,13,201,6,208,5,169,254,157,17
2170 data 142,189,65,142,96,201,0,208,248,       6550
     169,8,208,241,192,126,240,11,201,1
2180 data 208,236,169,9,157,25,142,208,229,      7559
     201,7,208,225,169,255,208,243,32,39
2190 data 167,174,111,140,189,65,142,74,         7735
     144,62,32,253,168,32,50,166,185,33,142
2200 data 201,65,240,8,201,71,240,4,201,78,      7151
     208,41,173,111,140,10,170,232,189
2210 data 161,142,56,233,2,157,161,142,174,      7148
     111,140,254,25,142,254,25,142,174
2220 data 111,140,208,9,173,93,140,10,176,       7809
     3,32,77,170,32,129,169,96,32,39,167
2230 data 174,111,140,189,65,142,74,74,144,      7519
     242,32,253,168,32,50,166,185,33,142
2240 data 201,65,240,20,201,71,240,16,201,       6443
     73,240,12,201,79,240,8,201,80,240,4
2250 data 201,78,208,209,169,2,141,112,140,      7350
     173,111,140,10,170,232,189,161,142
2260 data 24,109,112,140,157,161,142,174,        6286
     111,140,222,25,142,222,25,142,76,174
2270 data 167,32,39,167,174,111,140,189,17,      6817
     142,201,2,208,22,189,65,142,74,74
2280 data 74,144,102,32,50,166,185,33,142,       6884
     201,84,240,92,201,67,240,88,174,111
2290 data 140,208,37,169,0,32,52,166,185,        7025
     33,142,201,74,240,22,170,173,93,140
2300 data 74,144,4,224,84,240,11,169,1,32,       7576
     103,169,32,253,168,76,92,168,32,2,169
2310 data 32,27,166,240,12,32,41,166,201,0,      6998
     240,2,176,3,32,66,166,32,41,166,56
2320 data 233,2,157,161,142,174,111,140,         7163
     222,17,142,222,17,142,174,111,140,208
2330 data 3,32,36,170,32,129,169,96,32,39,       7686
     167,174,111,140,189,17,142,201,2,208
2340 data 23,189,65,142,74,74,74,74,144,         6901
     233,32,50,166,185,33,142,201,74,240,
     223
2350 data 201,68,240,219,174,111,140,208,        6434
     37,169,0,32,52,166,185,33,142,201,84
2360 data 240,22,170,173,93,140,74,144,4,        6861
     224,74,240,11,169,1,32,103,169,32,253
2370 data 168,76,217,168,32,2,169,32,27,         7017
     166,208,10,32,41,166,201,254,208,3,32
2380 data 66,166,32,41,166,24,105,2,157,         7298
     161,142,174,111,140,254,17,142,254,17
2390 data 142,76,127,168,169,0,76,103,169,       7771
     169,0,76,90,169,173,101,140,208,38
2400 data 32,36,169,201,76,240,4,201,77,         7239
     208,27,169,255,32,255,150,169,1,141,
     101
2410 data 140,76,12,170,174,111,140,222,81,      6826
     142,160,3,32,159,166,254,81,142,96
2420 data 32,36,169,201,75,208,31,173,81,        7389
     142,10,170,189,98,137,133,167,189,99
2430 data 137,133,168,172,73,142,169,32,         8009
     145,167,206,107,140,32,53,170,32,60,
     165
2440 data 96,32,116,169,173,93,140,13,112,       7209
     140,141,93,140,96,32,116,169,173,93
2450 data 140,45,113,140,141,93,140,96,170,      7638
     189,230,137,141,112,140,73,255,141
2460 data 113,140,96,174,111,140,240,40,         6723
     172,104,140,185,145,140,202,61,230,137
2470 data 240,24,160,255,32,203,172,74,176,      7416
     10,74,176,8,74,176,6,74,176,4,200
2480 data 200,200,200,200,16,23,160,20,16,       6989
     23,32,51,169,160,5,173,9,142,201,123
2490 data 240,5,201,119,240,2,200,200,200,       6755
     152,10,10,168,185,55,140,133,167,185
2500 data 56,140,133,168,185,57,140,133,         7184
     169,185,58,140,133,170,174,111,140,208
2510 data 25,173,9,142,201,125,208,18,173,       7615
     178,142,201,198,208,11,173,33,142,201
2520 data 73,208,4,169,196,208,17,160,255,       7522
     174,111,140,200,177,167,240,10,221
2530 data 178,142,208,246,177,169,157,178,       7131
     142,96,169,8,141,4,212,169,100,141
2540 data 0,212,141,1,212,169,6,141,5,212,       7630
     169,129,141,4,212,96,169,8,141,4,212
2550 data 169,20,141,0,212,141,1,212,169,2,      6861
     208,230,169,8,141,11,212,169,20,141
2560 data 7,212,141,8,212,169,96,141,12,         7119
     212,169,17,141,4,212,96,169,8,141,4,
     212
2570 data 173,33,142,201,73,240,12,201,79,       6968
     240,8,201,80,240,4,169,24,208,198,169
2580 data 33,141,0,212,141,1,212,169,22,         6687
     208,170,96,169,5,32,90,169,32,53,165
2590 data 173,11,221,173,10,221,157,91,142,      7391
     173,9,221,157,90,142,173,8,221,157
2600 data 89,142,96,206,102,140,240,1,96,        5901
     174,104,140,189,179,140,141,102,140
2610 data 174,104,140,189,145,140,141,105,       6918
     140,238,111,140,78,105,140,144,3,76
2620 data 240,172,32,39,167,174,111,140,         7201
     189,9,142,168,192,125,176,69,189,17,
     142
2630 data 201,2,208,15,192,123,240,50,32,        7206
     73,171,185,108,139,32,95,171,176,95
2640 data 174,111,140,189,9,142,74,144,18,       6856
     74,144,21,74,144,6,32,217,168,76,141
2650 data 171,32,92,168,76,141,171,32,150,       7499
     167,76,141,171,32,238,167,76,141,171
2660 data 32,73,171,185,107,139,16,204,189,      6961
     25,142,201,5,208,202,192,126,240,8
2670 data 32,73,171,185,106,139,16,6,32,73,      6840
     171,185,105,139,32,95,171,144,179
2680 data 174,111,140,189,9,142,201,126,         7190
     240,4,169,126,208,18,169,125,208,14,
     174
2690 data 111,140,189,9,142,240,4,169,123,       7419
     208,2,169,119,32,88,171,16,142,169
2700 data 0,32,52,166,185,33,142,56,233,65,      7042
     10,10,168,96,174,111,140,157,9,142
2710 data 96,32,88,171,32,226,166,32,50,         7150
     166,174,111,140,189,9,142,201,125,144
```


WWW.HOMECOMPUTERWORLD.COM

```
2720 data 26,185,33,142,201,65,240,18,201,      6872
     71,240,14,168,189,9,142,201,125,208
2730 data 4,192,78,240,2,56,96,24,96,173,       6401
     111,140,174,104,140,221,191,140,240
2740 data 244,76,168,170,173,92,140,10,170,     7760
     189,18,137,133,167,189,19,137,133
2750 data 168,160,255,96,32,10,150,32,155,      7864
     171,200,177,167,72,41,127,32,210,255
2760 data 104,16,244,96,141,106,140,32,68,      8215
     229,32,1,172,32,155,171,200,177,167
2770 data 16,251,140,105,140,173,106,140,       6644
     56,237,105,140,74,168,96,32,233,171
2780 data 208,251,32,233,171,240,246,96,        7167
     173,0,220,41,16,201,16,96,72,32,1,172
2790 data 104,32,10,150,169,34,32,10,150,       7064
     76,222,171,169,3,76,90,169,32,192,171
2800 data 162,12,24,76,240,255,162,48,56,       7526
     233,10,144,3,232,176,248,105,58,168
2810 data 138,32,210,255,152,76,210,255,        6834
     174,111,140,189,81,142,205,81,142,96
2820 data 169,0,32,52,166,185,33,142,96,32,     7803
     88,171,32,226,166,185,33,142,96,169
2830 data 0,32,229,172,169,2,32,211,172,76,     7553
     18,173,169,1,32,229,172,169,3,208
2840 data 241,32,222,172,201,212,208,237,       7258
     169,2,32,229,172,76,141,171,32,222,172
2850 data 201,212,208,222,169,3,208,239,        7414
     169,123,208,2,169,119,141,206,142,174
2860 data 111,140,189,73,142,72,173,206,        7317
     142,201,123,240,5,222,73,142,208,3,254
2870 data 73,142,173,206,142,201,123,240,       6481
     30,254,73,142,160,1,32,159,166,201,74
2880 data 240,34,201,68,240,30,189,73,142,      6634
     205,73,142,208,225,104,157,73,142,56
2890 data 96,222,73,142,160,1,32,159,166,       6473
     201,84,240,4,201,67,208,226,104,157
2900 data 73,142,24,96,174,111,140,202,189,     7295
     197,142,96,168,32,203,172,25,230,137
2910 data 157,197,142,96,174,111,140,189,       7180
     178,142,96,168,32,203,172,89,230,137
2920 data 157,197,142,96,32,39,167,174,111,     7379
     140,189,17,142,201,2,208,7,189,25
2930 data 142,201,5,240,3,76,215,170,32,        6444
     222,172,201,221,240,4,201,222,208,6,32
2940 data 146,169,76,141,171,32,36,172,208,     7373
     61,173,93,140,74,144,31,74,176,28
2950 data 32,4,167,32,50,166,185,33,142,        6458
     201,76,240,4,201,77,208,11,32,23,169
2960 data 169,1,32,90,169,76,141,171,32,46,     6841
     172,201,65,240,67,174,111,140,189
2970 data 73,142,205,73,142,176,3,76,68,        7365
     174,76,14,174,32,203,172,240,21,74,176
2980 data 9,74,176,9,74,176,9,76,102,172,       7945
     76,65,172,76,78,172,76,87,172,32,36
2990 data 172,144,4,169,213,208,2,169,211,      7208
     174,111,140,157,178,142,32,46,172,201
3000 data 65,208,24,32,98,171,144,43,174,       6159
     111,140,189,9,142,201,126,240,4,169
3010 data 126,208,2,169,125,76,68,171,201,      6952
     78,240,23,201,71,240,68,201,74,240
3020 data 58,201,84,240,57,201,67,240,53,       7365
     201,68,240,46,76,215,170,174,111,140
3030 data 189,9,142,201,125,240,53,32,36,       6998
     172,176,11,32,46,172,201,71,208,11,169
3040 data 125,208,2,169,126,32,95,171,144,      7294
     220,173,207,142,73,1,141,207,142,208
3050 data 3,76,14,174,76,68,174,174,111,        7322
     140,189,9,142,201,126,240,4,201,125,
     208
3060 data 203,32,98,171,144,185,176,219,        7006
     174,111,140,169,212,157,178,142,96,32
3070 data 36,172,208,38,32,113,172,144,25,      7170
     32,203,172,74,176,27,169,0,32,211,172
3080 data 169,221,174,111,140,157,178,142,      8018
     76,141,171,76,95,173,32,203,172,208
3090 data 248,32,5,174,169,123,32,55,172,       7549
     201,74,240,92,208,43,32,36,172,208,29
3100 data 32,117,172,144,16,32,203,172,74,      6576
     74,176,17,169,1,32,211,172,169,222
3110 data 208,199,32,203,172,208,203,32,5,      7210
     174,169,119,32,55,172,201,84,240,47
3120 data 201,66,240,43,201,71,240,39,201,      7378
     78,240,35,201,70,240,31,201,73,240
3130 data 27,201,80,240,23,201,72,240,19,       6213
     174,111,140,189,9,142,201,119,240,4
3140 data 169,119,208,2,169,123,32,88,171,      4982
     76,215,170
```

Dieser Programmteil heißt „rcs b“

```
1000 rem ---------------------------------
1010 rem    runman construction set v2.0
1020 rem
1030 rem    programmteil : screens [nr 1]
1040 rem ---------------------------------
1050 a=0:a$=chr$ (34)                           1541
1060 print "[clr,pur,2down,10right]***         3433
      please wait ***"
1070 read da:poke 20480+a,da                    1456
1080 if a<4095 then a=a+1:goto 1070             1912
1090 print "[2down,blk]load"a$"rcs d"a$",8,     2519
     1[3down]"
1100 print "[down]run[8up]                      1486
1110 poke 631,13:poke 632,13                    2109
1120 poke 198,2:end                             1140
1130 :
1140 data 32,32,32,32,75,32,32,69,32,32,75,     7513
     32,32,32,32,32,32,32,32,32,32,32,32
1150 data 32,32,32,32,32,32,32,32,32,32,76,     7428
     32,32,75,75,32,32,32,67,66,70,71,72
1160 data 66,70,73,72,66,70,71,72,68,32,32,     7560
     32,32,32,32,32,32,32,32,32,32,32,32
1170 data 32,32,32,67,70,71,72,66,66,68,32,     7545
     32,32,32,32,65,32,32,32,79,32,32,32
1180 data 65,32,32,32,75,32,69,32,32,75,32,     8243
     32,76,32,75,32,32,75,32,76,32,32,65
1190 data 32,32,32,32,32,32,32,75,32,65,32,     8241
     75,32,79,32,75,32,65,32,32,67,66,66
1200 data 70,71,72,66,70,71,72,66,84,84,84,     7799
     70,73,72,68,32,65,32,32,32,32,32,32
1210 data 67,66,70,71,72,66,66,80,66,66,70,     7919
     71,72,68,32,32,32,32,65,32,32,32,65
1220 data 32,32,32,32,32,32,79,32,32,32,65,     7429
     32,32,32,32,32,32,32,32,32,65,32,32
1230 data 32,32,32,32,32,65,32,32,32,32,75,     7774
     32,65,32,32,75,65,76,32,32,32,76,32
1240 data 79,32,75,32,65,32,32,75,32,32,32,     7723
     32,32,32,65,32,32,32,32,32,32,32,65
1250 data 32,67,70,71,72,66,78,66,66,70,71,     8103
     72,74,74,74,66,70,73,72,66,66,78,70
1260 data 71,72,68,32,32,32,32,75,65,32,32,     7587
     32,75,32,32,32,65,32,32,32,65,32,32
1270 data 32,32,32,32,65,32,32,32,32,32,32,     7728
     79,32,32,32,32,32,65,32,32,32,32,67
1280 data 66,70,71,72,70,71,72,70,71,72,78,     7705
     68,32,32,65,32,32,32,32,75,32,65,32
1290 data 32,69,75,32,32,79,75,32,32,75,32,     7799
     65,32,32,32,32,32,32,65,32,32,65
1300 data 32,32,65,32,32,32,32,32,32,65,32,67, 7483
     70,71,72,66,78,70,73,72,70,71,72,80
1310 data 66,70,71,72,66,78,66,68,32,32,32,     7789
     32,75,65,32,32,65,32,32,65,32,75,76
1320 data 32,32,65,32,32,32,32,65,32,32,32,     7504
     79,32,32,65,32,32,32,32,65,32,32,32
1330 data 32,32,32,32,67,66,70,71,72,70,71,     7908
     72,66,78,66,66,70,71,72,78,68,32,32
1340 data 65,32,32,32,32,79,32,32,65,32,32,     7438
     32,32,65,32,32,32,32,32,32,32,32,32
1350 data 32,65,32,32,65,32,32,32,32,32,32,     7946
     65,32,32,32,32,32,65,75,32,75,32,79
1360 data 75,32,65,32,75,32,32,65,32,75,32,     7766
     32,32,32,32,32,32,75,65,81,75,65,75
1370 data 32,32,32,32,32,65,32,32,67,66,70,     7925
     71,72,70,73,72,80,66,66,78,70,71,72
1380 data 66,78,66,70,71,72,68,32,32,67,66,     7778
     66,78,66,66,78,66,68,32,32,32,32,65
1390 data 32,32,32,32,32,65,32,32,79,32,32,     7495
```


WWW.HOMECOMPUTERWORLD.COM

```
      32,32,32,32,65,32,32,32,32,32,65,32
1400 data 32,32,32,32,32,32,32,32,32,32,32,  7715
      32,32,32,75,75,65,32,32,75,32,32,65
1410 data 75,32,79,32,75,32,32,75,32,65,32,  7913
      75,69,75,32,65,75,32,32,32,32,32,32
1420 data 32,32,32,32,32,32,67,66,66,66,78,  7744
      84,84,84,84,66,78,66,66,80,66,66,66
1430 data 66,66,66,78,66,66,66,66,66,78,66,  7525
      68,32,32,32,32,32,32,81,75,32,32,32
1440 data 32,76,32,32,32,32,32,76,32,32,32,  7672
      32,32,69,32,32,32,32,75,32,32,32,75
1450 data 32,32,32,32,32,32,32,32,67,70,71,  7455
      72,66,66,66,70,71,72,66,84,84,84,84
1460 data 84,66,70,73,72,66,66,66,66,66,66,  8278
      66,66,66,66,66,66,66,66,70,71,72,68
1470 data 32,32,32,32,65,32,32,32,32,32,65,  7351
      32,32,32,32,32,32,32,32,32,79,32,32
1480 data 32,32,32,32,32,32,32,32,32,32,32,  7456
      32,32,32,65,32,32,32,32,32,32,65,32
1490 data 32,32,75,32,65,32,75,32,32,32,75,  7727
      32,32,32,79,32,32,32,75,32,32,32,75
1500 data 32,32,32,32,69,32,32,32,65,32,32,  7617
      32,32,32,32,65,32,32,67,66,66,78,66
1510 data 66,66,66,66,66,66,66,70,73,72,66,  8289
      66,66,66,66,66,66,66,70,71,72,66,68
1520 data 32,32,65,32,32,32,32,32,32,65,32,  7396
      32,32,32,32,32,32,32,32,32,32,32,32
1530 data 32,32,79,32,32,32,32,32,32,32,32,  7429
      32,32,65,32,32,32,32,32,65,32,32,32
1540 data 32,32,32,65,32,32,32,32,32,32,32,  7774
      32,75,32,32,75,32,32,32,79,32,32,76
1550 data 32,32,32,75,76,32,32,65,32,32,32,  7652
      32,32,65,32,32,32,32,32,32,65,32,32
1560 data 67,66,70,71,72,66,66,66,66,66,66,  8134
      66,70,73,72,66,66,74,74,74,74,66,66
1570 data 66,78,66,66,68,32,32,65,32,32,32,  7461
      32,32,32,65,32,32,32,32,32,65,32,32
1580 data 32,32,32,32,32,32,32,32,79,32,32,  7507
      32,32,32,32,32,32,32,32,32,32,32,32
1590 data 32,65,32,32,32,32,32,32,65,32,32,  7849
      32,32,32,65,32,32,75,32,32,32,32,75
1600 data 32,79,32,32,75,32,69,32,32,75,32,  7726
      32,32,32,75,32,32,32,65,32,32,32,32
1610 data 32,32,65,32,32,67,66,66,78,66,66,  7794
      66,66,66,66,66,66,70,73,72,66,66,66
1620 data 66,66,66,66,66,66,70,71,72,66,68,32,  7685
      32,65,32,32,32,32,32,32,65,32,32,32
1630 data 32,32,32,32,32,32,32,32,32,32,32,  7441
      32,79,32,32,32,32,32,32,32,32,32,32
1640 data 65,32,32,32,32,32,65,32,32,32,32,  7575
      32,32,65,32,32,32,32,75,32,32,32,76
1650 data 32,75,32,32,76,32,79,32,75,32,32,  8052
      32,32,32,75,32,32,65,32,32,32,32,32
1660 data 65,32,32,32,32,32,32,65,32,32,67,  7223
      66,70,71,72,66,66,84,84,84,84,66,70
1670 data 73,72,66,66,66,66,66,66,66,66,66,  8020
      78,66,66,68,32,32,65,32,32,32,32,32
1680 data 32,65,32,32,32,32,32,65,32,32,32,  7618
      32,32,32,32,32,32,79,32,32,32,32,32
1690 data 32,32,32,32,32,32,32,32,32,32,32,  7298
      65,32,32,32,32,32,32,65,32,32,75,32
1700 data 32,65,32,32,75,32,32,69,32,75,32,  8206
      79,32,32,32,75,32,76,32,32,32,32,32
1710 data 32,76,69,75,32,65,32,32,32,32,67,  7906
      66,78,66,66,66,66,66,78,66,66,66,66
1720 data 66,66,66,66,66,66,80,66,66,66,66,66,  8117
      66,74,74,74,74,74,74,66,66,66,66,78
1730 data 66,68,32,32,32,32,81,32,32,75,32,  7726
      32,32,32,32,32,32,75,32,32,32,32,32
1740 data 32,32,32,32,32,75,32,32,32,32,32,  7782
      32,32,32,69,32,32,75,32,32,32,67,66
1750 data 70,73,72,66,66,70,71,72,66,66,70,  8116
      73,72,66,66,70,73,72,66,66,70,73,72
1760 data 66,66,70,71,72,66,70,73,72,70,71,  7853
      72,68,32,32,32,32,32,79,32,32,32,32
1770 data 65,32,32,32,32,79,32,32,32,32,79,  7585
      32,32,32,32,79,32,32,32,32,65,32,32
1780 data 32,79,32,32,65,32,32,32,32,32,75,  7995
      32,79,32,32,75,32,65,32,32,75,32,79
1790 data 32,69,32,32,79,32,32,32,75,79,32,  7686
      32,75,32,65,32,75,32,79,32,32,65,32
1800 data 32,32,32,67,66,70,73,72,66,66,66,  7945
      78,66,66,66,70,73,72,66,66,70,73,72
1810 data 66,66,66,80,66,66,66,70,71,72,66,  7914
      70,73,72,70,71,72,68,32,32,32,32,32
1820 data 79,32,32,32,32,32,32,32,32,32,79,  7552
      32,32,32,32,79,32,32,32,32,32,32,32
1830 data 32,32,65,32,32,32,79,32,32,65,32,  7836
      32,32,32,32,75,32,79,32,69,75,32,32
1840 data 32,32,32,32,79,75,32,32,32,79,75,  7610
      32,32,32,32,32,75,32,32,65,75,32,32
1850 data 79,75,32,65,75,32,32,32,67,66,70,  8222
      73,72,66,66,66,66,66,66,66,70,73,72
1860 data 66,66,70,73,72,66,66,70,71,72,66,  8240
      66,66,78,66,66,70,73,72,70,71,72,68
1870 data 32,32,32,32,32,79,32,32,32,32,32,  7591
      32,32,32,32,79,32,32,32,32,79,32,32
1880 data 32,32,65,32,32,32,32,32,32,32,32,  7550
      79,32,32,65,32,32,32,32,32,32,75,79
1890 data 32,32,32,32,32,32,32,75,32,79,32,  8023
      32,75,32,79,32,32,75,32,65,32,32,75
1900 data 32,32,32,75,32,79,32,32,65,75,32,  7744
      32,32,67,66,70,73,72,66,66,70,71,72
1910 data 66,66,70,73,72,66,66,66,80,66,66,  8189
      66,66,78,66,66,66,70,73,72,66,70,73
1920 data 72,70,71,72,68,32,32,32,32,32,79,  7511
      32,32,32,32,65,32,32,32,32,79,32,32
1930 data 32,32,32,32,32,32,32,32,32,32,32,  7513
      32,79,32,32,32,79,32,32,65,32,32,32
1940 data 32,32,32,75,79,32,75,32,32,65,32,  8084
      75,32,32,79,75,32,32,32,32,75,32,32
1950 data 75,32,32,69,75,32,79,75,32,32,79,  7821
      75,32,65,32,32,32,32,67,66,70,73,72
1960 data 66,66,70,71,72,66,66,70,73,72,66,  7928
      66,70,71,72,66,66,70,73,72,66,66,70
1970 data 73,72,66,70,73,72,70,71,72,68,32,  7811
      32,32,32,32,79,32,32,32,32,65,32,32
1980 data 32,32,79,32,32,32,32,65,32,32,32,  7567
      32,79,32,32,32,32,79,32,32,32,79,32
1990 data 32,65,32,32,32,32,32,32,75,79,32,  8038
      32,75,32,65,32,32,75,32,79,32,69,75
2000 data 32,65,32,32,75,32,79,32,32,75,32,  8249
      79,32,32,75,79,32,75,65,75,32,32,32
2010 data 67,66,66,80,66,66,66,66,78,66,66,  8295
      66,66,80,66,66,66,66,78,66,66,66,66
2020 data 80,66,66,66,66,80,66,66,66,80,66,  7785
      66,78,66,68,32,32,32,32,81,32,32,75
2030 data 32,32,32,32,32,32,32,32,32,32,75,  7380
      32,32,32,32,32,32,32,32,32,32,32,32
2040 data 32,32,32,32,32,32,75,32,32,32,32,  8102
      67,66,70,73,72,66,66,70,73,72,66,66
2050 data 70,73,72,66,66,70,73,71,72,66,70,  8297
      73,72,70,71,72,66,66,66,66,70,71,72
2060 data 66,66,68,32,32,32,32,32,79,32,32,  7516
      32,32,79,32,32,32,32,79,32,32,32,32
2070 data 79,65,32,32,32,79,32,32,65,32,32,  7657
      32,32,32,32,65,32,32,32,32,32,32,32
2080 data 32,32,79,75,32,32,32,79,32,32,32,  7686
      32,79,75,32,32,32,79,65,32,32,32,79
2090 data 32,32,65,32,32,32,32,32,32,65,32,  7909
      75,32,32,32,32,67,66,70,73,72,68,67
2100 data 70,73,72,66,66,70,73,72,66,66,70,  8172
      73,71,72,66,70,73,72,66,78,66,70,71
2110 data 72,66,70,71,72,66,66,68,32,32,32,  7761
      32,32,79,32,32,32,32,79,32,32,32,32
2120 data 79,32,32,32,32,79,65,32,32,32,79,  7561
      32,32,32,32,32,65,32,32,32,65,32,32
2130 data 32,32,32,32,32,32,32,79,32,75,32,  8012
      69,79,32,32,32,32,79,32,32,69,32,79
2140 data 65,75,32,32,79,32,32,32,32,32,65,  7601
      32,32,32,65,32,75,32,32,32,32,67,70
2150 data 71,73,72,66,66,70,73,72,68,67,70,  8284
      73,72,66,66,70,73,71,72,66,70,73,72
2160 data 66,66,66,66,78,66,66,70,71,72,66,  7709
      66,68,32,32,32,32,65,79,32,32,32,32
2170 data 79,32,32,32,32,79,32,32,32,32,79,  7606
      65,32,32,32,79,32,32,32,32,32,32,32
2180 data 32,32,65,32,32,32,32,32,32,32,32,  7628
      65,79,32,75,32,32,79,32,32,32,69,79
2190 data 32,32,32,32,79,65,32,75,32,79,32,  7727
      32,32,32,32,69,32,32,32,65,32,75,32
2200 data 32,32,32,67,70,71,73,72,66,70,71,  7782
      73,72,66,66,70,73,72,68,67,70,73,71
2210 data 72,66,70,73,72,66,70,71,72,66,66,  8008
      66,70,71,72,66,66,68,32,32,32,32,65
2220 data 79,32,32,32,65,79,32,32,32,32,79,  7687
      32,32,32,32,79,65,32,32,32,79,32,32
2230 data 32,65,32,32,32,32,32,65,32,32,32,  7794
```


WWW.HOMECOMPUTERWORLD.COM

```
     32,32,32,32,32,65,79,75,32,32,65,79
2240 data 32,75,32,32,79,32,32,32,32,79,65,   7572
     75,32,32,79,32,32,32,65,32,32,32,32
2250 data 32,65,32,75,32,32,32,32,67,70,71,   7897
     73,72,66,70,71,73,72,66,66,70,73,71
2260 data 72,66,70,73,71,72,66,70,73,72,66,   8131
     66,78,66,70,71,72,70,71,72,66,66,68
2270 data 32,32,32,32,65,79,32,32,32,65,79,   7804
     32,32,32,32,79,65,32,32,32,79,65,32
2280 data 32,32,79,32,32,32,32,32,32,65,32,   7779
     32,65,32,32,32,32,32,32,75,65,79
2290 data 32,32,75,65,79,32,69,75,32,79,65,   8100
     32,32,32,79,65,32,75,32,79,32,32,32
2300 data 75,32,32,65,32,32,65,32,75,32,32,   7819
     32,32,67,66,78,80,66,66,66,78,80,66
2310 data 66,66,66,80,78,66,66,66,80,78,66,   7766
     66,66,80,66,66,66,66,66,66,78,66,66
2320 data 78,66,66,66,68,32,32,32,32,32,32,   7549
     75,32,32,32,32,32,32,32,32,69,32,32
2330 data 75,32,32,32,75,32,32,32,76,32,32,   7513
     32,32,76,32,32,32,69,32,32,32,32,32
2340 data 32,67,70,71,72,66,66,70,71,72,68,   8082
     32,32,67,66,70,71,72,70,71,72,66,70
2350 data 71,72,66,74,74,74,74,70,71,72,66,   7423
     66,70,71,72,68,32,32,32,32,65,32,32
2360 data 32,32,65,32,32,32,32,32,32,32,65,   7282
     32,32,65,32,32,32,65,32,32,32,32,32
2370 data 32,32,65,32,32,32,32,32,65,32,32,   7426
     32,32,32,65,32,32,32,32,65,32,32,32
2380 data 32,32,75,32,65,32,32,65,32,32,32,   7631
     65,32,32,75,32,32,32,32,65,32,32,32
2390 data 32,65,32,32,32,32,32,32,32,65,32,69,  7449
     32,75,65,32,32,32,32,67,66,70,71,72
2400 data 70,71,72,66,70,71,72,66,66,68,32,   7449
     32,32,65,32,75,32,32,65,32,32,32,32
2410 data 67,70,71,72,66,66,66,78,70,71,72,   7727
     68,32,32,32,65,32,32,65,32,32,32,65
2420 data 32,32,32,67,70,71,72,78,66,70,71,   7673
     72,78,66,68,32,32,32,32,65,32,32,32
2430 data 32,32,32,65,32,32,32,32,32,65,32,   7678
     32,65,32,32,32,65,32,32,32,32,32,65
2440 data 32,32,32,32,65,32,32,32,32,32,32,   7450
     32,32,65,32,32,32,32,32,32,65,32,32
2450 data 32,75,32,65,32,32,65,32,69,32,65,   7581
     32,32,32,32,32,65,32,32,32,32,65,32
2460 data 32,32,32,32,32,32,32,65,75,32,32,   7626
     32,32,32,65,32,67,70,71,72,78,70,73
2470 data 71,72,66,66,78,70,71,72,68,32,65,   7461
     32,32,32,32,65,32,32,32,32,32,32,67
2480 data 70,71,72,70,71,72,68,32,65,32,32,   7828
     32,65,32,32,32,79,65,32,32,32,32,32
2490 data 65,32,32,32,65,32,75,32,81,65,32,   7607
     32,75,32,32,32,32,32,65,32,32,65,32
2500 data 32,32,65,32,32,32,65,32,32,32,79,   7599
     65,32,32,32,32,32,65,32,67,70,71,72
2510 data 70,73,72,78,70,71,72,68,32,32,32,   7651
     32,65,32,32,65,32,32,75,65,32,32,32
2520 data 65,32,32,32,79,65,75,32,32,32,75,   7736
     65,32,32,32,65,32,32,79,32,32,32,65
2530 data 32,32,32,32,32,32,65,32,67,78,70,   7707
     71,72,78,70,71,72,78 66,66,70,73,71
2540 data 72,66,70,71,72,78,68,32,32,65,75,   8144
     32,79,32,75,32,65,32,32,32,32,32,32
2550 data 65,32,32,32,32,65,32,32,32,65,32,   7729
     32,32,32,32,79,65,32,32,32,65,32,32
2560 data 32,67,70,71,72,70,73,72,66,70,71,   7993
     72,68,32,32,32,32,65,32,32,32,32,65
2570 data 32,32,32,65,32,32,32,32,32,79,65,   7426
     32,32,32,65,32,32,32,32,32,65,32,32
2580 data 79,32,32,32,65,32,32,32,32,32,32,   7434
     65,32,75,32,32,65,32,69,32,65,76,32
2590 data 32,32,76,79,65,32,75,32,65,32,32,   7914
     75,32,32,65,32,32,79,32,75,32,65,69
2600 data 32,32,32,67,66,78,66,66,66,66,78,   7956
     66,66,66,78,66,84,84,84,66,80,78,66
2610 data 66,66,78,66,66,66,66,66,78,66,66,   8057
     80,66,66,66,78,66,68,32,32,32,32,32
2620 data 32,32,32,32,32,32,32,32,32,32,32,   7412
     32,32,69,32,32,75,32,32,32,32,32,32
2630 data 75,32,32,32,32,32,76,32,32,32,32,   7491
     32,32,32,32,32,32,32,32,32,32,32,32
2640 data 32,32,32,32,32,67,66,66,66,66,66,   7970
     70,71,72,66,70,71,72,70,71,72,68,67
2650 data 66,70,71,72,66,68,32,32,32,32,75,   7760
     32,69,32,32,75,32,32,32,32,75,32,32
2660 data 32,32,32,32,32,32,65,32,32,32,65,   7369
     32,32,65,32,32,32,32,32,65,32,32,32
2670 data 32,32,67,70,73,72,66,70,71,72,66,   7582
     74,74,74,66,66,68,32,32,32,32,32,32
2680 data 65,32,32,32,65,32,32,65,32,75,32,   7679
     32,32,65,32,32,32,32,32,32,32,79,32
2690 data 32,32,65,32,32,32,32,32,32,32,32,   7535
     32,32,32,32,75,32,65,32,32,32,65,32
2700 data 67,78,66,66,66,66,70,71,72,66,68,   7792
     32,32,32,32,79,32,32,32,65,32,32,32
2710 data 32,32,32,32,67,70,71,72,66,66,66,   7729
     78,68,32,32,65,32,32,32,32,32,32,32
2720 data 32,65,32,32,32,32,32,32,32,79,32,   7561
     32,32,65,32,32,32,32,32,32,32,32,32
2730 data 65,32,32,32,32,32,32,32,32,65,32,   7482
     32,32,32,32,69,32,32,65,32,32,32,32
2740 data 32,32,32,79,32,32,32,65,32,32,32,   7807
     32,75,32,32,32,75,65,32,32,75,32,32
2750 data 32,32,32,65,32,67,66,70,71,72,66,   7713
     66,78,66,66,68,32,32,32,32,79,32,67
2760 data 66,78,66,70,71,72,70,73,72,66,70,   8029
     71,72,66,70,71,72,66,68,32,65,32,32
2770 data 32,32,65,32,32,32,32,32,32,32,32,   7405
     32,32,32,79,32,32,32,32,32,32,65,32
2780 data 32,79,32,32,32,65,32,32,32,65,32,   7669
     32,32,32,65,32,32,32,32,65,32,32,32
2790 data 32,32,32,32,32,32,32,32,79,32,32,   7558
     32,32,32,32,65,32,32,79,32,32,32,65
2800 data 32,32,32,65,32,32,32,32,65,32,32,   7492
     32,69,65,32,32,75,32,32,32,32,32,32
2810 data 32,75,79,32,32,32,75,32,32,65,32,   8050
     32,79,32,75,32,65,32,75,32,65,32,75
2820 data 32,32,65,32,32,67,66,78,66,66,70,   7830
     71,72,68,32,32,32,67,66,80,70,71,72
2830 data 66,66,66,78,66,70,73,72,66,70,71,   8163
     72,66,70,71,72,66,66,70,71,72,68,32
2840 data 32,32,32,32,32,65,32,32,32,32,32,   7450
     32,32,32,32,65,32,32,32,32,32,32,32
2850 data 79,32,32,32,65,32,32,32,65,32,32,   7543
     32,32,65,32,32,32,32,32,32,32,32,65
2860 data 32,32,32,32,32,32,32,32,32,65,32,   7561
     32,32,32,32,32,32,79,32,32,32,65,32
2870 data 32,32,65,32,32,32,32,65,32,32,32,   7336
     32,32,32,32,32,65,32,32,32,32,32,32
2880 data 81,32,75,65,76,75,32,32,75,32,76,   7771
     79,32,75,32,65,32,32,32,65,32,32,32
2890 data 32,65,32,76,32,75,32,32,76,32,65,   7687
     32,32,32,32,32,67,66,66,66,78,66,84
2900 data 84,84,84,84,66,80,66,66,66,78,66,   8099
     66,66,78,66,68,32,67,78,66,66,84,84
2910 data 84,84,66,66,78,66,66,68,32,32,32,   7597
     32,32,32,75,32,32,32,32,32,32,69,32
2920 data 75,32                               671
```

„rcs c" ist der Titel dieses Abschnitts

```
1000 rem --------------------------------
1010 rem   runman construction set v2.0
1020 rem
1030 rem   programmteil : screens [nr 2]
1040 rem --------------------------------
1050 :
1060 a=0:a$=chr$ (34)                        1541
1070 print "[clr,grn,2down,10right]***       3445
     please wait ***"
1080 read da:poke 24576+a,da                 1596
1090 if a<4063 then a=a+1:goto 1080          2200
1100 print "[2down,blk]load"a$"rcs e"a$",8,   2521
     1[3down]"
1110 print "[down]run[8up]                   1486
1120 poke 631,13:poke 632,13                 2109
1130 poke 198,2:end                          1140
1140 :
1150 data 32,32,75,32,32,32,32,32,32,75,32,  7560
     32,32,32,32,32,75,32,32,32,32,32,32
1160 data 32,32,67,70,71,72,66,66,66,70,71,  7755
     72,66,66,66,66,66,66,66,70,71,72,66
1170 data 66,66,66,66,70,71,72,70,73,72,66,  7539
     66,70,71,72,66,68,32,32,32,32,65,32
1180 data 32,32,32,32,65,32,32,32,32,32,32,  7552
     32,32,32,65,32,32,32,32,32,32,32,65
1190 data 32,32,79,32,32,32,32,65,32,32,32,  7565
     32,32,32,32,65,32,32,32,75,32,65,32
1200 data 32,75,32,32,75,32,32,32,65,75,32,  7958
```


WWW.HOMECOMPUTERWORLD.COM

```
      32,32,75,32,32,65,32,32,79,32,32,75
1210 data 32,65,32,75,32,32,32,32,32,65,32,  8312
      32,67,66,66,78,66,66,70,71,72,66,70
1220 data 71,72,78,66,66,70,71,72,68,32,65,  7963
      32,32,79,32,67,66,66,78,66,66,68,32
1230 data 32,32,32,65,32,32,32,32,32,32,32,  7564
      32,32,65,32,32,32,65,32,32,32,32
1240 data 65,32,32,32,65,32,32,79,32,32,32,  7739
      32,32,32,32,32,32,32,32,69,65,32,75
1250 data 32,32,75,32,32,32,32,65,32,81,32,  7733
      65,32,75,32,32,32,65,32,75,32,65,32
1260 data 32,79,75,32,32,75,32,32,32,32,32,  7791
      32,67,70,71,72,70,71,72,70,71,72,66
1270 data 70,71,72,66,70,71,72,70,71,72,70,  7943
      71,72,66,66,78,66,66,80,66,70,71,72
1280 data 68,32,32,32,32,32,32,32,65,32,32,  7480
      65,32,32,65,32,32,32,65,32,32,32,65
1290 data 32,32,65,32,32,65,32,32,32,32,32,  7327
      32,32,32,32,65,32,32,32,32,32,32,32
1300 data 32,32,65,32,32,65,32,75,65,32,32,  7568
      32,65,32,32,32,65,32,32,65,32,32,65
1310 data 32,32,32,32,32,32,32,69,75,32,65,32,  8088
      75,32,32,32,32,32,67,66,78,66,70,71
1320 data 72,70,71,72,66,66,78,68,32,32,65,  7722
      32,75,65,32,32,65,32,32,32,67,70,71
1330 data 72,66,66,78,66,70,71,72,68,32,32,  7699
      32,32,32,32,32,65,32,32,65,32,32,32
1340 data 32,32,67,70,71,72,66,78,68,32,65,  7339
      32,32,32,32,32,65,32,32,32,32,32,32
1350 data 65,32,32,32,32,32,75,32,32,75,65,  7545
      32,32,65,32,32,32,69,32,32,32,65,32
1360 data 32,32,32,32,65,32,32,75,32,32,65,  7743
      32,32,32,75,32,32,65,32,32,32,32,67
1370 data 70,71,72,66,78,66,70,71,72,66,66,  7822
      66,68,32,32,65,32,32,32,32,67,70,71,72
1380 data 66,66,66,70,71,72,66,66,66,66,70,  7772
      71,72,68,32,32,32,32,65,32,32,32,32
1390 data 32,65,32,32,32,32,32,32,32,65,32,  7738
      32,32,32,32,65,32,32,32,32,32,65,32
1400 data 32,32,32,32,32,65,32,32,32,32,32,  7503
      32,65,32,75,32,32,32,65,32,76,32,32
1410 data 32,76,32,65,32,75,32,32,32,65,32,  8089
      75,32,32,75,65,32,32,32,75,69,32,65
1420 data 32,32,32,32,32,67,66,78,66,66,66,66,  8004
      66,78,66,66,74,74,74,66,66,78,66,66
1430 data 66,66,66,78,66,66,66,66,66,78,66,  7818
      66,66,66,66,66,78,66,68,32,32,32,32
1440 data 32,32,76,32,75,32,76,32,75,32,32,  7642
      32,32,32,32,32,32,32,32,32,32,75
1450 data 32,32,75,32,32,69,32,32,75,32,32,  7461
      75,32,32,32,67,70,71,72,66,84,84,84
1460 data 66,70,71,72,66,68,32,32,32,32,32,  7705
      32,32,32,32,67,70,71,72,66,70,71,72
1470 data 68,67,70,71,72,66,68,32,32,32,32,  7952
      65,32,32,32,32,32,32,32,65,32,32,32
1480 data 32,32,75,32,32,75,32,32,32,32,32,  7525
      65,32,32,32,65,32,32,32,32,65,32,32
1490 data 32,32,32,32,32,65,32,32,32,32,32,  7474
      32,32,65,32,32,67,70,71,72,70,73,72
1500 data 70,71,72,68,32,65,32,32,32,65,32,  7796
      32,32,75,65,32,75,32,32,32,32,32,65
1510 data 32,75,32,32,69,75,32,65,32,76,32,  7951
      32,65,32,32,79,32,32,65,32,32,32,65
1520 data 32,32,67,78,70,71,72,66,78,66,66,  7722
      68,32,32,67,66,78,66,66,70,71,72,66
1530 data 66,78,66,66,68,32,65,32,32,79,32,  7691
      32,65,75,32,76,65,75,32,32,32,32,65
1540 data 32,32,32,32,32,32,32,32,32,32,32,  7447
      32,32,32,65,32,32,32,32,32,32,32,32
1550 data 65,32,32,79,67,66,78,66,66,66,78,  7959
      66,68,32,32,32,65,32,75,32,32,76,32
1560 data 32,32,32,32,32,32,32,32,65,32,32,  7603
      32,32,32,32,32,32,65,32,32,79,32,32
1570 data 32,32,32,32,32,32,32,32,67,66,78,  7778
      66,70,71,72,66,68,32,32,32,32,75,69
1580 data 32,32,65,32,32,75,32,32,32,75,32,  7786
      65,75,32,79,32,69,32,75,32,32,32,75
1590 data 32,75,32,32,32,32,32,65,32,32,32,  8064
      32,32,67,70,71,72,66,66,78,66,66,66
1600 data 70,71,72,66,66,78,66,70,73,72,70,  7998
      71,72,66,66,66,70,71,72,68,32,32,32
1610 data 32,65,32,32,32,32,32,32,32,65,32,  7540
      32,32,32,32,32,32,32,65,32,32,32,32
1620 data 32,32,79,32,32,65,32,32,32,32,32,  7306
      65,32,32,32,32,32,32,65,32,32,32,32
1630 data 32,32,32,65,32,32,32,32,32,32,32,  7647
      75,65,32,75,32,32,32,81,79,75,32,65
1640 data 75,32,32,32,32,65,32,32,32,32,32,  7590
      32,65,32,32,32,32,32,32,75,65,32,75
1650 data 32,32,75,32,67,66,78,66,66,70,71,  7774
      72,70,73,72,70,71,72,66,68,32,32,65
1660 data 32,32,75,32,75,32,65,32,75,32,32,  7763
      32,67,70,71,72,70,71,72,66,68,32,32
1670 data 32,32,32,32,65,32,32,79,32,32,65,  7705
      32,32,32,67,66,78,70,71,72,66,66,66
1680 data 78,66,66,68,32,32,32,32,65,32,32,  7584
      65,32,32,32,32,32,32,32,32,32,65,32
1690 data 32,79,32,32,65,32,32,32,32,32,32,  7474
      32,65,32,32,32,32,32,32,32,32,32,32
1700 data 32,75,65,32,75,65,32,76,32,75,32,  7921
      75,32,76,32,65,32,32,79,75,32,65,76
1710 data 32,75,32,32,76,32,65,32,32,75,32,  8360
      69,32,75,32,32,32,67,66,78,66,66,78
1720 data 66,66,74,74,74,74,74,66,66,78,66,  7402
      66,80,66,66,78,66,84,84,84,84,66,66
1730 data 78,66,66,66,66,66,66,66,68,32,32,  7906
      32,32,32,32,75,76,32,32,76,69,32,32
1740 data 75,32,32,32,32,32,75,32,32,32,32,  7712
      32,75,32,32,76,32,32,32,75,76,32,32
1750 data 32,75,32,32,32,67,66,70,71,72,66,  7598
      84,84,66,66,70,71,72,66,66,66,70,71
1760 data 72,70,71,72,66,66,66,70,71,72,66,  7677
      74,74,74,66,70,71,72,66,68,32,32,32
1770 data 32,32,65,32,32,32,32,32,32,32,65,  7183
      32,32,32,32,32,65,32,32,65,32,32,32
1780 data 32,32,65,32,32,32,32,32,32,32,65,  7390
      32,32,32,32,32,32,75,32,65,75,32,32
1790 data 32,75,32,32,65,32,75,32,32,75,65,  8091
      32,75,65,32,32,32,75,32,65,32,32,69
1800 data 32,32,75,32,65,32,75,32,32,32,67,  7668
      66,70,71,72,66,70,71,72,66,66,78,66
1810 data 70,71,72,70,71,72,70,71,72,70,71,  7882
      72,66,78,66,66,70,71,72,66,70,71,72
1820 data 66,68,32,32,32,32,32,65,32,32,32,  7370
      65,32,32,32,32,32,32,65,32,32,65,32
1830 data 32,65,32,32,65,32,32,32,32,32,32,  7734
      65,32,32,32,65,32,32,32,32,32,32,75
1840 data 32,65,32,75,32,65,32,69,32,75,32,  7946
      32,65,32,75,65,75,32,65,32,75,65,32
1850 data 75,32,32,32,75,65,75,32,32,65,32,  7952
      75,32,32,32,67,66,70,71,72,66,70,71
1860 data 72,66,70,71,72,66,78,66,70,71,72,  8119
      70,71,72,66,78,66,70,71,72,66,70,71
1870 data 72,66,70,71,72,66,68,32,32,32,32,  7910
      32,65,32,32,32,65,32,32,32,65,32,32
1880 data 32,32,32,65,32,32,65,32,32,32,32,  7567
      32,65,32,32,32,65,32,32,32,65,32,32
1890 data 32,32,32,32,32,75,65,32,75,32,65,  7704
      32,75,32,65,75,32,32,32,75,65,75,32
1900 data 65,81,32,75,32,32,65,32,69,32,65,  7768
      75,32,32,65,32,75,32,32,32,67,66,70
1910 data 71,72,66,70,71,72,66,66,78,66,70,  7967
      71,72,70,71,72,70,71,72,70,71,72,66
1920 data 78,66,66,70,71,72,66,70,71,72,66,  7776
      68,32,32,32,32,32,65,32,32,32,65,32
1930 data 32,32,32,32,32,65,32,32,65,32,32,  7519
      65,32,32,65,32,32,32,32,32,32,65,32
1940 data 32,32,65,32,32,32,32,32,32,32,75,  7608
      65,32,69,32,65,32,75,32,32,32,32,65
1950 data 75,32,65,75,32,65,75,32,65,32,32,  7731
      75,32,32,32,65,32,75,32,65,32,75,32
1960 data 32,32,67,66,70,71,72,66,66,78,66,  7549
      66,70,71,72,66,78,66,70,71,72,70,71
1970 data 72,66,78,66,70,71,72,66,66,78,66,  7604
      66,70,71,72,66,68,32,32,32,32,32,65
1980 data 32,32,32,32,32,32,32,65,32,32,32,  7540
      32,32,65,32,32,65,32,32,32,32,32,65
1990 data 32,32,32,32,32,32,32,65,32,32,32,  7624
      32,32,32,75,32,65,32,75,32,32,32,75
2000 data 32,65,32,32,32,69,75,32,65,32,75,65,  7827
      76,32,75,32,76,65,32,32,32,75,69,32
2010 data 32,65,32,75,32,32,32,67,66,66,78,  8457
      66,66,66,66,66,66,66,78,66,66,66,66
2020 data 66,78,66,66,78,66,84,84,84,66,78,  7959
      66,66,66,66,66,66,66,78,66,66,66,68,32
2030 data 32,32,32,75,32,32,69,32,75,32,76,  7616
      32,32,75,32,32,76,32,75,32,32,32,32
2040 data 32,32,32,32,75,32,32,32,32,32,75,  7617
      32,32,32,69,32,32,32,67,66,66,70,71
2050 data 72,66,70,71,72,68,67,70,71,72,70,  8239
      71,72,70,73,72,66,68,32,67,66,70,71
2060 data 72,66,66,66,70,71,72,66,66,68,32,  7512
      32,32,32,32,32,65,32,32,32,65,32,32
```


WWW.HOMECOMPUTERWORLD.COM

```
2070 data 32,32,65,32,32,65,32,32,79,32,32,  7387
     32,32,32,32,32,65,32,32,32,32,32,65
2080 data 32,32,32,32,32,32,32,32,32,32,65,  7432
     32,32,32,65,32,32,32,32,65,32,32,65
2090 data 32,32,79,32,32,32,32,75,32,32,65,  7431
     81,32,75,32,32,65,32,32,32,32,32,32
2100 data 32,75,32,32,65,32,75,32,65,32,32,  7906
     75,32,65,32,32,65,32,32,79,32,67,66
2110 data 66,70,71,72,78,70,71,72,68,32,65,  7788
     32,32,32,32,32,32,67,70,71,72,78,66
2120 data 66,70,71,72,84,84,66,78,68,32,65,  7254
     32,32,79,32,32,32,32,32,65,32,32,32
2130 data 65,32,32,32,65,32,32,32,32,32,32,  7471
     32,32,65,32,32,32,32,32,65,32,32,32
2140 data 32,32,32,32,65,32,32,79,32,32,69,  7833
     32,32,65,32,75,32,65,32,75,32,65,75
2150 data 32,32,32,32,32,32,32,65,32,32,32,  7587
     69,32,65,32,32,75,32,32,75,32,65,32
2160 data 32,79,32,67,70,71,72,78,66,66,70,  8133
     71,72,66,66,78,70,71,72,68,32,32,67
2170 data 70,71,72,70,71,72,66,78,66,66,70,  8133
     73,72,66,66,78,68,32,79,32,32,65
2180 data 32,32,32,32,32,65,32,32,32,32,32,  7306
     65,32,32,32,32,32,32,65,32,32,65,32
2190 data 32,32,32,32,32,32,79,32,32,32,32,  7639
     69,79,75,32,32,65,32,32,32,32,76,65
2200 data 32,32,69,32,75,65,32,32,32,32,32,  7573
     32,65,32,32,65,32,32,32,32,32,32,79
2210 data 32,32,32,32,32,67,66,80,70,71,72,78,  7852
     66,68,32,67,70,71,72,66,66,66,70,71
2220 data 72,68,32,32,32,75,65,75,32,65,76,  7766
     32,32,32,32,32,79,32,32,32,32,32,32
2230 data 32,32,65,32,32,32,32,32,32,32,65,  7160
     32,32,32,32,32,65,32,32,32,32,67,70
2240 data 71,72,70,71,72,68,32,32,75,32,79,  7903
     76,32,32,76,32,75,32,32,65,32,32,75
2250 data 32,32,32,76,65,32,32,32,32,75,65,  7774
     32,32,32,32,32,32,65,32,32,65,32,67
2260 data 70,71,72,66,80,66,84,84,70,71,72,  7518
     66,70,71,72,66,70,73,72,66,66,78,66
2270 data 68,32,67,70,71,72,68,32,32,32,32,  7457
     65,32,32,65,32,32,32,65,32,32,32,32
2280 data 32,32,32,65,32,32,32,65,32,32,32,  7558
     79,32,32,32,32,32,32,32,32,32,65,32
2290 data 32,32,32,32,32,65,75,32,65,32,75,  7923
     32,65,32,32,75,32,75,32,32,65,32,75
2300 data 32,65,32,69,32,79,32,75,76,32,32,  8137
     32,32,76,32,65,32,32,32,32,67,66,78
2310 data 66,66,78,66,66,66,78,66,66,66,66,  8461
     66,66,66,78,66,66,66,78,66,66,66,80
2320 data 66,66,66,84,84,84,84,66,66,78,66,  7334
     68,32,32,32,32,75,32,32,69,32,32,32
2330 data 32,75,32,32,32,32,32,32,32,32,75,  7540
     32,32,32,32,32,32,32,32,32,32,32,32
2340 data 32,32,32,32,32,32,32,32,32,67,66,70,  8133
     71,72,66,68,67,70,71,72,70,71,72,68
2350 data 67,70,71,72,70,71,72,66,66,70,71,  7400
     72,68,32,32,32,32,32,32,32,32,32,32
2360 data 32,32,32,32,32,65,32,32,32,32,32,  7600
     65,32,32,65,32,32,32,32,65,32,32,65
2370 data 32,32,32,32,65,32,32,32,32,32,69,  7371
     32,32,75,32,32,32,32,32,32,32,32,65
2380 data 32,32,32,32,32,65,32,32,65,32,32,  7438
     32,32,65,32,32,65,32,32,32,32,65,32
2390 data 32,67,66,66,66,70,71,72,66,66,68,  7852
     32,32,32,32,32,65,32,32,32,75,32,65
2400 data 69,32,65,32,32,32,75,65,32,32,65,  7385
     32,32,75,32,65,32,32,32,32,32,32,32
2410 data 65,32,32,32,32,32,32,32,32,32,65,  7911
     32,67,70,71,72,78,66,70,71,72,68,67
2420 data 70,71,72,66,78,70,71,72,70,71,72,  7710
     68,32,32,32,32,32,65,32,32,32,32,32
2430 data 32,32,32,32,65,32,32,32,65,32,32,  7312
     32,32,65,32,32,32,32,65,32,32,32,32
2440 data 65,32,32,65,32,32,32,75,32,32,32,  7749
     65,32,32,75,32,32,32,32,32,75,65,32
2450 data 32,32,65,32,32,75,32,65,32,32,81,  7412
     32,65,32,32,32,32,65,32,32,65,32,32
2460 data 67,66,70,71,72,78,70,71,72,68,32,  8194
     32,67,66,70,71,71,72,66,78,70,71,72
2470 data 66,78,66,70,71,72,78,68,32,32,32,  7553
     65,32,32,65,32,32,32,32,32,65,32,32
2480 data 32,65,32,32,32,32,32,32,32,65,65,  7615
     32,32,32,32,65,32,32,32,32,32,65,32
2490 data 32,32,32,69,32,65,32,32,65,75,32,  7579
     32,32,32,65,32,32,32,65,32,32,32,32
2500 data 32,32,75,65,65,32,32,32,32,65,32,  7695
     32,32,32,32,65,32,32,67,66,66,70,71
2510 data 72,66,78,66,70,71,72,66,78,66,66,  8042
     70,71,72,68,32,32,67,66,70,71,71,72
2520 data 68,32,32,65,32,32,32,32,32,65,32,  7675
     32,32,32,32,32,65,32,32,32,32,32,65
2530 data 32,32,32,32,32,32,65,32,32,32,32,  7628
     32,32,32,65,65,32,32,32,32,65,75,32
2540 data 32,32,32,65,32,75,32,32,32,32,65,  7498
     32,32,32,75,32,65,32,76,32,32,32,76
2550 data 65,75,32,32,32,32,32,32,65,65,32,  8018
     32,67,66,78,70,71,72,66,66,78,66,70
2560 data 71,72,68,32,65,32,67,66,66,70,71,  7521
     72,66,84,84,84,70,71,72,68,32,32,32
2570 data 32,32,65,65,32,32,32,32,32,32,65,  7318
     32,32,32,32,32,32,65,32,32,32,32,65,32
2580 data 32,32,32,32,65,32,32,32,32,32,32,  7530
     65,32,32,32,32,32,32,75,65,65,76,32
2590 data 75,32,32,76,65,32,32,32,32,32,32,  7669
     65,32,69,32,65,32,32,32,32,75,65,32
2600 data 32,69,32,75,32,65,32,32,32,32,67,  7780
     66,66,78,66,66,84,84,84,84,66,78,66
2610 data 68,32,67,66,66,78,66,66,66,78,66,  8049
     68,67,66,66,78,66,66,66,66,66,66,78
2620 data 66,68,32,32,32,32,32,75,32,75,32,  7882
     32,32,32,32,75,32,32,32,32,32,32,75
2630 data 32,32,32,32,32,32,75,32,32,32,32,  7624
     32,32,32,69,32,75,32,32,32,32,67,70
2640 data 71,72,70,71,72,68,32,67,70,71,72,  8042
     66,66,66,66,70,71,72,66,68,32,67,66
2650 data 70,71,72,66,66,66,70,71,72,70,71,  7861
     72,68,32,32,32,32,65,32,32,65,32,32
2660 data 32,32,32,65,32,32,32,32,32,32,65,  7165
     32,32,32,32,32,32,32,65,32,32,32,32
2670 data 32,65,32,32,65,32,32,32,32,32,75,  7733
     65,32,32,65,75,32,32,69,32,65,32,81
2680 data 32,32,75,32,65,32,75,32,32,32,75,  7627
     32,65,32,32,75,32,32,65,32,32,65,75
2690 data 32,32,32,67,66,78,68,67,78,70,71,  8230
     72,66,66,78,66,70,71,72,66,66,78,66
2700 data 66,70,71,72,66,66,78,66,66,70,71,  7760
     72,78,68,67,78,66,68,32,32,32,32,32
2710 data 32,32,32,32,65,32,32,32,32,32,32,  7480
     65,32,32,32,32,32,32,32,65,32,32,32
2720 data 32,32,32,32,65,32,32,32,32,32,32,  7587
     32,32,32,32,75,32,32,32,75,32,65,32
2730 data 32,32,75,32,32,65,32,32,75,32,32,  7625
     69,32,65,32,75,32,32,32,32,32,65,32
2740 data 69,32,75,32,32,32,32,32,67,66,70,  7960
     71,72,66,66,78,68,32,67,70,71,72,78
2750 data 66,66,70,71,72,66,66,78,66,70,71,  8516
     72,66,66,66,78,66,66,70,71,72,66,68
2760 data 32,32,32,32,32,65,32,32,32,32,32,  7489
     32,32,32,65,32,32,32,32,32,65,32,32
2770 data 32,32,32,32,65,32,32,32,32,32,32,  7523
     32,32,65,32,32,32,32,32,32,69,32,65
2780 data 32,32,75,32,32,32,75,32,65,32,32,  7644
     32,75,32,65,32,75,32,32,32,32,65,32
2790 data 75,32,32,32,75,32,32,65,75,32,32,  7941
     32,32,67,66,66,78,66,70,71,72,66,70
2800 data 71,72,78,68,32,67,66,66,78,70,71,  7976
     72,66,66,66,78,70,73,72,66,70,71,72
2810 data 66,78,66,66,68,32,32,32,32,32,32,  7443
     32,32,65,32,32,32,65,32,32,32,32,32
2820 data 32,32,32,32,65,32,32,32,32,32,32,  7546
     79,32,32,32,65,32,32,32,32,32,32,32
2830 data 32,32,32,75,32,32,32,65,32,32,32,  7832
     65,32,32,69,75,32,32,32,75,32,65,32
2840 data 32,75,32,32,32,79,32,32,75,65,69,  7635
     32,32,32,75,32,32,32,67,66,70,71,72
2850 data 66,78,66,66,66,78,70,71,72,66,66,  8276
     70,71,72,66,78,66,66,70,71,72,68,79
2860 data 67,66,66,78,66,70,71,72,66,68,32,  7706
     32,32,32,32,65,32,32,32,32,32,32,32
2870 data 32,65,32,32,32,32,65,32,32,32,32,  7636
     32,32,65,32,32,79,32,32,32,32,32,32
2880 data 65,32,32,32,32,32,32,32,75,65,32,  7494
     75,32,32,32,32,75,32,65,32,32,32,32
2890 data 65,32,32,75,32,32,32,65,32,75,79,  8022
     75,32,32,32,75,32,65,32,75,32,32,32
2900 data 67,66,66,78,66,66,66,66,66,66,66,  7871
     66,78,66,66,66,66,78,66,66,66,68,32
2910 data 67,78,66,66,80,66,66,66,66,66,66,  5319
     78,66,66,68,32
```

In „rcs d" sind die Spielfelder untergebracht

WWW.HOMECOMPUTERWORLD.COM

```
1000 rem ------------------------------
1010 rem   runman construction set v2.0
1020 rem
1030 rem      by christian drapatz '86
1040 rem
1050 rem   programmteil 4 : sprite's
1060 rem ------------------------------
1070 :
1080 a=0:a$=chr$ (34)                                  1541
1090 print "[clr,blu,2down,10right]***                 3453
      please wait ***"
1100 read da:poke 12288+a,da                           1567
1110 if a<3583 then a=a+1:goto 1100                    2066
1120 print "[2down,blk]load"a$"rcs f"a$",8,             2523
     1[3down]"
1130 print "[down]run[8up]                              1486
1140 poke 631,13:poke 632,13                           2109
1150 poke 198,2:end                                    1140
1160 :
1170 data 0,255,0,0,87,0,1,117,0,1,85,0,0,             6229
     87,0,0,20,0,0,40,0,0,40,0,0,104,0,0
1180 data 124,0,0,168,0,0,160,0,0,40,0,0,              6130
     44,0,0,240,0,0,0,0,0,0,0,0,0,0,0,0
1190 data 0,0,0,0,0,0,0,0,255,0,0,87,0,1,              6185
     117,0,1,85,0,0,87,0,0,20,0,0,40,0,0
1200 data 40,0,0,104,0,0,124,0,0,168,0,0,              6050
     138,0,15,138,0,3,195,0,0,15,0,0,0,0
1210 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6236
     0,255,0,0,87,0,1,117,0,1,85,0,0,87
1220 data 0,0,20,0,0,40,0,0,40,0,0,40,0,0,             5758
     60,0,0,24,0,0,40,0,0,40,0,0,40,0,0
1230 data 252,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6074
     0,0,0,0,0,0,0,0,255,0,0,87,0,1,117,0
1240 data 1,85,0,0,87,0,0,20,0,0,40,0,0,40,            6050
     0,0,104,0,0,124,0,0,168,0,0,138,0
1250 data 0,138,0,0,130,0,3,207,0,0,0,0,0,             5967
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,255
1260 data 0,3,255,0,3,255,0,3,255,0,3,255,             7892
     0,2,222,0,2,170,0,0,168,0,0,168,0,0
1270 data 252,0,0,168,0,0,136,0,0,136,0,0,             6420
     136,0,0,204,0,0,0,0,0,0,0,0,0,0,0,0
1280 data 0,0,0,0,0,0,0,0,0,3,255,0,3,87,0,1,          6343
     221,0,1,85,0,3,119,0,0,84,0,0,168
1290 data 0,2,170,0,4,168,64,0,252,0,0,168,            6944
     0,0,136,0,0,136,0,0,136,0,3,207,0
1300 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6808
     0,0,0,3,255,0,3,255,0,3,255,0,3,255
1310 data 0,3,255,0,0,220,0,0,168,0,2,170,             7255
     0,4,168,64,0,252,0,0,168,0,0,136,0
1320 data 0,136,0,0,136,0,0,204,0,0,0,0,0,             6248
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,255
1330 data 0,3,255,0,3,255,0,3,255,0,3,255,             7951
     0,0,220,0,2,170,0,2,170,0,0,168,0,0
1340 data 252,0,0,168,0,0,136,0,0,140,0,0,             6406
     128,0,0,192,0,0,0,0,0,0,0,0,0,0,0,0
1350 data 0,0,0,0,0,0,0,0,3,255,0,3,255,0,             6317
     3,255,0,3,255,0,3,255,0,0,220,0,2,170
1360 data 0,2,170,0,0,168,0,0,252,0,0,168,             6829
     0,0,136,0,0,200,0,0,8,0,0,12,0,0,0
1370 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6815
     0,0,255,0,0,213,0,0,93,64,0,85,64,0
1380 data 213,0,0,20,0,0,40,0,0,40,0,0,41,             6190
     0,0,61,0,0,42,0,0,162,0,0,162,0,0,130
1390 data 0,0,243,192,0,0,0,0,0,0,0,0,0,0,             6503
     0,0,0,0,0,0,0,0,0,0,0,0,0,36,0,0,169
1400 data 0,2,170,64,10,170,144,10,170,144,            6544
     0,37,80,0,36,0,0,36,0,0,36,0,0,36
1410 data 0,0,36,0,0,36,0,0,36,0,0,32,0,0,             6320
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1420 data 0,255,0,0,213,0,0,93,64,0,85,64,             6227
     0,213,0,0,20,0,0,40,0,0,40,0,0,41,0
1430 data 0,61,0,0,42,0,0,42,0,0,8,0,0,40,             6129
     0,0,56,0,0,15,0,0,0,0,0,0,0,0,0,0,0
1440 data 0,0,0,0,0,0,0,0,255,0,0,213,0,0,93,          6215
     64,0,85,64,0,213,0,0,20,0,0,40,0,0
1450 data 40,0,0,41,0,0,61,0,0,42,0,0,162,             6406
     0,0,162,240,0,195,192,0,240,0,0,0,0
1460 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6511
     0,255,0,0,213,0,0,93,64,0,85,64,0,213
1470 data 0,0,20,0,0,40,0,0,40,0,0,40,0,0,             5763
     60,0,0,36,0,0,40,0,0,40,0,0,40,0,0
1480 data 63,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,            5456
     0,0,0,0,0,0,128,0,0,160,0,2,168,0
1490 data 2,170,0,2,170,0,10,106,128,10,86,            6266
     128,41,85,160,43,223,160,43,93,104
1500 data 37,255,88,21,255,104,11,255,224,             7182
     2,255,128,0,126,0,0,0,0,0,0,0,0,0,0
1510 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,40,0,           6542
     0,170,0,0,170,0,2,170,128,2,150,128
1520 data 10,90,160,9,213,160,43,85,168,41,            7398
     95,104,37,253,88,21,255,88,11,255
1530 data 224,3,255,192,0,254,0,0,0,0,0,0,            6259
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,32,0
1540 data 0,128,0,0,160,0,2,170,0,2,170,               6195
     128,10,165,128,10,90,160,41,245,160,41
1550 data 87,168,39,95,104,37,253,88,21,.              6998
     255,88,11,255,224,1,255,64,0,254,0,0
1560 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             5776
     0,0,2,40,0,0,8,0,0,138,0,0,170,0,2
1570 data 170,128,2,150,128,10,90,160,9,               7231
     217,160,43,87,168,41,95,104,38,253,88
1580 data 21,255,88,11,255,240,3,255,128,0,            6057
     254,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1590 data 0,0,0,0,0,8,32,0,0,40,0,0,168,0,             6183
     2,170,0,2,166,0,10,150,128,10,90,128
1600 data 43,217,160,43,93,160,37,95,104,              6550
     37,221,88,23,255,88,11,255,240,3,255
1610 data 128,0,254,0,0,0,0,0,0,0,0,0,0,0,0,          6228
     0,0,0,0,0,0,0,0,0,0,0,0,0,84,0,1,85
1620 data 0,1,85,0,1,153,0,1,153,0,1,221,0,            6406
     1,85,0,1,101,0,1,85,0,1,153,0,1,101
1630 data 0,1,85,0,1,85,0,1,17,0,0,0,0,0,0,            5641
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1640 data 0,84,0,1,85,0,1,85,0,1,153,0,1,              6396
     221,0,1,153,0,1,85,0,1,101,0,1,85,0
1650 data 1,153,0,1,101,0,1,85,0,1,85,0,1,             5689
     17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1660 data 0,0,0,0,0,0,0,0,84,0,1,85,0,1,85,            6446
     0,1,221,0,1,153,0,1,153,0,1,85,0,1
1670 data 101,0,1,85,0,1,153,0,1,101,0,1,              6338
     85,0,1,85,0,1,17,0,0,0,0,0,0,0,0,0,0
1680 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,84,0,            6118
     1,81,0,1,21,0,1,137,0,1,17,0,0,153
1690 data 0,1,85,0,0,85,0,1,69,0,1,152,0,1,            6462
     101,0,1,81,0,1,85,0,1,17,0,0,0,0,0
1700 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6301
     0,0,0,68,0,1,81,0,1,21,0,1,8,0,1,17
1710 data 0,0,136,0,1,84,0,0,5,0,1,69,0,0,             6742
     136,0,1,5,0,1,81,0,1,20,0,1,17,0,0
1720 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6180
     0,0,0,0,0,0,68,0,1,1,0,0,16,0,0,8,0
1730 data 1,16,0,0,128,0,0,4,0,0,0,0,64,               6436
     0,0,128,0,1,0,0,0,1,0,0,4,0,1,1,0,0
1740 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6118
     0,0,0,0,0,0,4,0,1,1,0,0,0,0,0,0,1
1750 data 0,0,0,128,0,0,4,0,0,0,0,0,0,0,0,             6316
     0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0
1760 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6165
     0,0,0,84,0,1,85,0,1,85,0,1,153,0,1
1770 data 153,0,1,221,0,1,85,0,1,101,0,1,              6806
     85,0,1,153,0,1,101,0,1,85,0,1,85,0,0
1780 data 68,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,            5721
     0,0,0,0,0,0,0,0,0,84,0,1,85,0,1,85
1790 data 0,1,153,0,1,221,0,1,153,0,1,85,0,            6792
     1,101,0,1,85,0,1,153,0,1,101,0,1,85
1800 data 0,1,85,0,0,68,0,0,0,0,0,0,0,0,0,             5822
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,84,0
1810 data 1,85,0,1,85,0,1,221,0,1,153,0,1,             7018
     153,0,1,85,0,1,101,0,1,85,0,1,153,0
1820 data 1,101,0,1,85,0,1,85,0,0,68,0,0,0,            5495
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1830 data 0,0,0,0,84,0,1,85,0,1,85,0,2,101,            6489
     0,3,117,0,2,101,0,1,85,0,1,149,0,1
1840 data 85,0,2,101,0,1,149,0,1,85,0,1,85,            6270
     0,1,17,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1850 data 0,0,0,0,0,0,0,0,0,0,84,0,1,85,0,             6292
     1,85,0,1,102,0,1,119,0,1,102,0,1,85
1860 data 0,1,89,0,1,85,0,1,102,0,1,89,0,1,            6406
     85,0,1,85,0,1,17,0,0,0,0,0,0,0,0,0
1870 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,84,            6055
     0,1,85,0,1,85,0,1,149,0,1,213,0,1
1880 data 149,0,1,85,0,2,69,0,1,5,0,0,5,0,             6857
     0,21,0,1,85,0,1,85,0,1,17,0,0,0,0,0
1890 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6007
     0,0,0,84,0,1,85,0,1,85,0,1,149,0,1
1900 data 213,0,1,149,0,1,85,0,2,69,0,1,5,             6976
     0,0,5,0,0,21,0,1,85,0,1,85,0,0,68,0
1910 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,             6005
     0,0,0,0,0,0,0,0,84,0,1,85,0,1,85,0,1
1920 data 89,0,1,93,0,1,89,0,1,85,0,1,70,0,            6679
     1,65,0,1,64,0,1,80,0,1,85,0,1,85,0
1930 data 1,17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,            5485
     0,0,0,0,0,0,0,0,0,0,84,0,1,85,0,1
1940 data 85,0,1,89,0,1,93,0,1,89,0,1,85,0,            6516
```


WWW.HOMECOMPUTERWORLD.COM

```
      1,70,0,1,65,0,1,64,0,1,80,0,1,85,0
1950 data 1,85,0,0,68,0,0,0,0,0,0,0,0,0,0,   6203
      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128
1960 data 0,2,162,0,10,168,0,2,170,32,10,    7319
      154,128,10,86,160,42,221,160,43,93,168
1970 data 41,223,104,21,253,104,11,255,224,  5897
      2,191,128,0,126,0,0,0,0,0,0,0,0
1980 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,   6223
      0,2,0,0,8,32,0,0,168,128,0,168,32,2
1990 data 170,0,2,154,0,10,150,128,10,86,    6502
      128,41,223,160,37,253,96,11,255,224
2000 data 2,191,128,0,126,0,0,0,0,0,0,0,0,   6364
      0,0,0,0,0,0,0,0,0,0,0,0,2,0,2,0,8
2010 data 8,0,128,0,0,0,0,0,0,0,0,0,32,0,    6550
      0,168,0,2,170,0,2,150,0,10,93,128
2020 data 9,191,128,11,221,96,2,191,128,0,   6451
      126,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2030 data 0,0,0,0,32,0,0,2,0,32,0,0,0,0,0,  5867
      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,32
2040 data 0,0,40,0,0,154,0,2,230,128,2,183,  5803
      128,0,190,0,0,0,0,0,0,0,0,0,0,0
2050 data 0,0,0,0,0,0,0,0,0,5,85,80,5,85,80, 7191
      6,170,144,6,170,144,6,85,144,6,85,144
2060 data 6,125,144,6,125,144,6,85,144,6,    7695
      255,144,6,255,144,6,85,144,6,125,144
2070 data 6,125,144,6,85,144,6,85,144,6,     7105
      170,144,6,170,144,5,85,80,5,85,80,0,0
2080 data 0,0,5,85,80,5,85,80,7,255,144,7,   7725
      255,144,7,85,144,7,85,144,7,125,144
2090 data 7,125,144,7,85,144,7,255,144,7,    7505
      255,144,7,85,144,7,125,144,7,125,144
2100 data 7,85,144,7,85,144,7,255,144,7,     7248
      255,144,5,85,80,5,85,80,0,0,0,0,5,85
2110 data 80,5,85,80,6,170,144,6,170,144,7,  6981
      85,144,7,85,144,7,125,144,7,125,144
2120 data 7,85,144,6,170,144,6,170,144,6,    6672
      85,208,6,125,208,6,125,208,6,85,208
2130 data 6,85,208,6,170,144,6,170,144,5,    6868
      85,80,5,85,80,0,0,0,0,5,85,80,5,85,80
2140 data 6,170,144,6,170,144,7,85,144,7,    7338
      85,144,7,125,144,7,125,144,7,85,144
2150 data 6,170,144,6,170,144,7,85,144,7,    7248
      125,144,7,125,144,7,85,144,7,85,144
2160 data 6,170,144,6,170,144,5,85,80,5,85,  7670
      80,0,0,0,0,5,85,80,5,85,80,6,255,144
2170 data 6,255,144,6,85,144,6,85,144,6,     7131
      125,144,6,125,144,6,85,144,6,170,144
2180 data 6,170,144,7,85,144,7,125,144,7,    6797
      125,144,7,85,144,7,85,144,7,255,144
2190 data 7,255,144,5,85,80,5,85,80,0,0,0,   7543
      0,5,85,80,5,85,80,6,170,144,6,170,144
2200 data 6,85,208,6,85,208,6,125,208,6,     6826
      125,208,6,85,208,6,170,144,6,170,144
2210 data 7,85,144,7,125,144,7,125,144,7,    6658
      85,144,7,85,144,6,170,144,6,170,144
2220 data 5,85,80,5,85,80,0,0,0,0,5,85,80,   7186
      5,85,80,6,255,208,6,255,208,6,85,208
2230 data 6,85,208,6,125,208,6,125,208,6,    6542
      85,208,6,170,144,6,170,144,6,85,144
2240 data 6,125,144,6,125,144,6,85,144,6,    7277
      85,144,6,170,144,6,170,144,5,85,80,5
2250 data 85,80,0,0,0,0,5,85,80,5,85,80,6,   6654
      170,144,6,170,144,7,85,144,7,85,144
2260 data 7,125,144,7,125,144,7,85,144,7,    7814
      255,144,7,255,144,7,85,144,7,125,144
2270 data 7,125,144,7,85,144,7,85,144,7,     7206
      255,144,7,255,144,5,85,80,5,85,80,0,0
2280 data 0,0,5,85,80,5,85,80,6,170,144,6,   7636
      170,144,6,85,144,6,85,144,6,125,144
2290 data 6,125,144,6,85,144,6,170,144,6,    7319
      170,144,6,85,144,6,125,144,6,125,144
2300 data 6,85,144,6,85,144,6,170,144,6,     7287
      170,144,5,85,80,5,85,80,0,0,0,0,5,85
2310 data 80,5,85,80,6,170,144,6,170,144,6,  6958
      85,144,6,85,144,6,125,144,6,125,144
2320 data 6,85,144,6,170,144,6,170,144,7,    7136
      85,144,7,125,144,7,125,144,7,85,144
2330 data 7,85,144,7,255,144,7,255,144,5,    6909
      85,80,5,85,80,0,0,0,0,5,85,80,5,85,80
2340 data 7,255,208,7,255,208,7,85,208,7,    7145
      85,208,7,105,208,7,105,208,7,85,208
2350 data 7,255,208,7,255,208,7,85,208,7,    7180
      105,208,7,105,208,7,85,208,7,85,208
2360 data 7,255,208,7,255,208,5,85,80,5,85,  7526
      80,0,0,0,0,8,170,0,42,170,160,10,106
2370 data 160,10,170,104,41,154,104,42,90,   7345
      170,170,153,170,171,159,162,165,221
2380 data 170,169,85,104,41,85,104,37,221,   7912
      96,37,125,88,41,85,96,10,85,32,2,85
2390 data 128,15,87,240,63,215,252,63,247,   6941
      252,63,255,252,0,0,0,0,0,0,0,0,0,0
2400 data 3,255,0,3,255,0,3,255,0,3,255,0,   6540
      7,87,64,2,154,0,0,168,0,0,252,0,0,168
2410 data 0,0,136,0,0,136,0,3,207,0,0,0,0,   6648
      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2420 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,   6605
      3,255,0,3,255,0,19,255,16,11,255,128
2430 data 10,154,128,2,170,0,3,3,0,0,0,0,0,  5745
      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2440 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,   6309
      0,0,0,2,170,0,2,170,0,11,190,128,10
2450 data 170,128,10,170,128,11,255,128,19,  6392
      255,16,3,255,0,3,255,0,0,0,0,0,0,0,0
2460 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,   6381
      0,0,0,0,0,0,0,3,207,0,0,136,0,0,136,0
2470 data 0,136,0,0,168,0,4,252,64,2,170,0,  6313
      0,168,0,3,87,0,3,223,0,3,255,0,0,252
2480 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,   5972
      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2490 data 0,0,0,0,3,255,0,3,87,0,15,223,     6896
      192,6,86,64,2,154,0,2,250,0,0,168,0,0
2500 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,   5292
      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Die Spritedaten in „rcs e"

```
1000 rem ------------------------------
1010 rem   runman construction set v2.0
1020 rem
1030 rem      by christian drapatz '86
1040 rem
1050 rem   programmteil 5 : zeichensatz
1060 rem ------------------------------
1070 :
1080 sp(0)=8192:sp(1)=36096:sp(2)=44720      3912
1090 a(0)=703:a(1)=192:a(2)=206              2458
1100 n=0:q$=chr$ (13)                        1547
1110 print "[clr,wht,2down,10right]***       3245
      please wait ***"
1120 a=0                                      358
1130 read da:poke sp(n)+a,da                 1502
1140 if a<a(n) then a=a+1:goto 1130          2040
1150 n=n+1                                    340
1160 if n<3 then 1120                         963
1170 print "[clr,2right,cyn]nun wird das     4655
      programm unter dem file"
1180 print "[down,2right]'runman c.set v2'   4727
      auf diskette, mit"
1190 print "[down,2right]93 bloecke,         2814
      abgespeichert !"
1191 print "[down,2right][ ist denn noch     3996
      platz auf diskette ? ]"
1200 print "[2down,2right,red]sollen die     4146
      einzelfiles vorher noch"
1210 print "[down,2right]geloescht werden    3257
      [j/n] ?"
1220 print "[home,14down,2right]antwort :    3199
      [2spaces,2left]";
1230 open 1,0:input#1,a$:close 1             1360
1240 a$=left$ (a$,1)                          741
1250 if a$="j" then print q$"[down,2right,    5575
      grn]scratch rcs*":open 1,8,15,"s:rcs*"
      :close 1:goto 1270
1260 if a$<>"n" then 1220                    1442
1270 print q$"[down,2right,red]nach          4555
      erscheinen von 'ready',muessen "
1280 print "[down,2right]sie,um zu spielen,  3419
      'run' ein."
1290 print q$"[down,2right,grn]save runman    3426
      c.set v2":sys sp(1)
1300 data 60,102,110,110,96,98,60,0,14,22,   6667
      22,30,38,38,207,0,124,182,182,60,54
1310 data 54,124,0,62,102,194,192,196,108,   6693
      60,0,124,182,182,54,54,54,124,0,254
```


WWW.HOMECOMPUTERWORLD.COM

```
1320 data 98,96,120,98,102,254,0,254,98,96,      7035
     120,96,96,240,0,62,102,194,192,207
1330 data 102,60,0,238,108,108,124,108,108,      7052
     238,0,60,24,24,24,24,24,60,0,62,76
1340 data 108,12,12,108,56,0,115,178,180,        6647
     60,54,50,115,0,240,96,96,96,98,102,254
1350 data 0,238,126,126,110,102,102,227,0,       6698
     230,246,126,110,102,102,231,0,60,102
1360 data 198,198,198,204,124,0,124,182,         6735
     182,52,48,48,120,0,60,78,198,214,222
1370 data 236,126,1,124,182,182,60,52,50,        6684
     123,0,30,114,96,112,28,204,120,0,126
1380 data 178,48,48,48,48,120,0,239,102,         6388
     102,102,102,102,60,0,116,214,150,54,54
1390 data 50,28,0,221,221,217,218,252,216,       6782
     144,0,102,100,56,24,28,38,102,0,231
1400 data 98,124,24,24,24,60,0,254,204,152,      6852
     48,98,198,254,0,60,48,48,48,48,48
1410 data 60,0,28,36,48,124,48,98,252,0,60,      6588
     12,12,12,12,12,60,0,0,24,60,126,24
1420 data 24,24,24,0,16,48,127,127,48,16,0,      6389
     0,0,0,0,0,0,0,0,24,24,24,24,0,0,24
1430 data 0,102,102,102,0,0,0,0,0,102,102,       6476
     255,102,255,102,102,0,24,62,96,60,6
1440 data 124,24,0,98,102,12,24,48,102,70,       7109
     0,60,102,60,56,103,102,63,0,6,12,24
1450 data 0,0,0,0,0,12,24,48,48,48,24,12,0,      6310
     48,24,12,12,12,24,48,0,0,102,60,255
1460 data 60,102,0,0,0,24,24,126,24,24,0,0,      6019
     0,0,0,0,0,24,24,48,0,0,0,126,0,0,0
1470 data 0,0,0,0,0,0,24,24,0,0,3,6,12,24,       6382
     48,96,0,60,78,214,214,214,228,124,0
1480 data 24,56,24,24,24,24,60,0,60,102,         6520
     102,6,24,112,126,0,60,102,6,28,6,102
1490 data 60,0,12,28,44,76,126,12,12,0,30,       6311
     48,48,28,6,6,60,0,12,24,48,108,102
1500 data 102,60,0,126,6,12,24,48,48,48,0,       6387
     24,102,118,24,110,102,24,0,60,102,102
1510 data 54,6,12,24,0,0,24,24,0,0,24,24,0,      6544
     0,24,24,0,0,24,24,48,14,24,48,96,48
1520 data 24,14,0,0,0,126,0,0,126,0,0,112,       6886
     24,12,6,12,24,112,0,254,6,6,62,48,0
1530 data 48,0,0,0,0,0,0,0,0,0,65,85,65,65,      6324
     65,85,65,65,0,0,0,170,170,170,170
1540 data 255,0,0,0,2,10,42,170,255,0,0,0,       6520
     170,168,160,128,192,0,48,252,48,48
1550 data 48,48,48,0,0,0,170,170,168,160,        6307
     224,65,85,65,105,105,85,65,65,0,0,0
1560 data 170,170,10,42,63,16,16,16,154,         7168
     154,16,16,16,0,0,0,234,186,174,171,255
1570 data 0,5,61,61,61,60,0,0,0,52,52,52,        7126
     60,60,60,0,0,60,60,60,52,52,52,0,65
1580 data 85,65,105,105,170,170,255,16,16,       6795
     16,16,16,16,16,16,16,16,16,154,154
1590 data 170,170,255,85,215,215,85,125,         7283
     125,85,20,0,126,126,126,126,126,126,0
1600 data 36,36,228,4,4,252,0,0,0,0,0,171,       7670
     174,178,202,255,36,36,39,32,32,63,0
1610 data 0,0,126,66,66,66,66,126,0,0,0,         4710
     255,0,0,255,0,0
1620 data 169,54,133,1,169,0,133,166,165,        7211
     166,10,10,10,170,160,0,189,98,141,153
1630 data 176,0,232,200,192,6,208,244,160,       7474
     0,177,176,145,180,165,177,197,179,208
1640 data 6,165,176,197,178,176,14,230,176,      7079
     208,2,230,177,230,180,208,228,230
1650 data 181,208,224,230,166,165,166,201,       7453
     10,208,196,162,8,32,186,255,169,15
1660 data 162,178,160,141,32,189,255,169,        6932
     43,162,5,160,100,32,216,255,169,55,133
1670 data 1,76,116,164,176,174,127,175,1,8,      6619
     0,0,0,144,100,159,208,8,0,0,0,32,192
1680 data 34,0,112,0,0,0,160,161,174,52,24,      6409
     0,0,0,48,0,62,0,176,0,0,0,128,91,140
1690 data 10,39,0,0,0,112,192,114,101,51,0,      6673
     0,0,80,224,111,0,144,0,0,0,144,224
1700 data 175,37,54,0,0,0,176,0,190,5,86,0,      7815
     0,82,85,78,77,65,78,32,67,46,83,69
1710 data 84,32,86,50                            1191
1720 data 27,8,10,0,158,32,32,50,48,55,55,       7799
     58,82,85,78,77,65,78,32,67,46,32,83
1730 data 69,84,0,0,0,169,0,141,32,208,141,      7599
     33,208,133,2,173,17,208,41,239,141
1740 data 17,208,165,2,10,10,10,170,160,0,       6734
     189,123,8,153,167,0,232,200,192,6,208
1750 data 244,160,0,177,167,145,171,165,         7884
     168,197,170,208,6,165,167,197,169,240
1760 data 14,230,167,208,2,230,168,230,171,      6665
     208,228,230,172,208,224,230,2,165
1770 data 2,201,8,208,196,162,255,232,189,       7530
     187,8,157,7,8,224,4,208,245,76,0,144
1780 data 208,8,52,24,0,144,0,0,52,24,10,        6576
     39,0,160,0,0,10,39,101,51,0,128,0,0
1790 data 101,51,37,54,0,112,0,0,37,54,5,        6376
     86,0,16,0,0,5,86,5,100,0,48,0,0,0,16
1800 data 224,47,0,80,0,0,0,112,192,114,0,       6415
     32,0,0,51,54,56,54,52,0,0,0,0,0,0,0
1810 data 0,0,0,0,0,0,0,0,0                      1587
```

„rcs f" beinhaltet den geänderten Zeichensatz

FLEXIBLE DB-ORGANISATION CONTRA „STÜCKLISTEN-EXPLOSIONEN"

Immer wenn die computerunterstützte Produktionsplanung und -steuerung mit wenig stabilen Rahmenbedingungen leben muß, ist die DV-technische Seite besonders gefordert — in kleineren und mittleren Betrieben oft überfordert. Häufige Produktwechsel, Störungen an den Werkzeugen oder Maschinen oder Unterbrechungen bei der Materialversorgung stellen typische PPS-Problemfälle dar. Und geradezu wenn eine hohe Anzahl von Varianten innerhalb des Erzeugungsspektrums gegeben ist, fressen Stücklisten und Bedarfsrechnung Speicherkapazität und Rechnergeschwindigkeit auf.

Mit der vorliegenden Arbeit werden in der Praxis entwickelte und erprobte Datenstrukturen und Algorithmen gezeigt. Unter anderem wird ein Expertensystem für variantenreiche Produktionsprogramme realisiert, das durch seine Verknüpfungstechniken redundante Daten abbaut, „Stücklisten-Explosion" vermeidet und somit die wirtschaftliche Verarbeitung großer Datenmengen im PPS-Bereich ermöglicht.

CW-EDITION

Fachbücher für die Computerwelt

Rheinstraße 28, 8000 München 40, Tel.: 089/3 60 86-0
Nur Festbestellungen können berücksichtigt werden. Ein Rückgaberecht besteht nicht. Bei Sachmangel kann der Käufer lediglich Nachlieferung beanspruchen. Preisänderungen vorbehalten

Ich/Wir bestelle(n):

_____ Expl. Best.Nr.: _____

Name _____

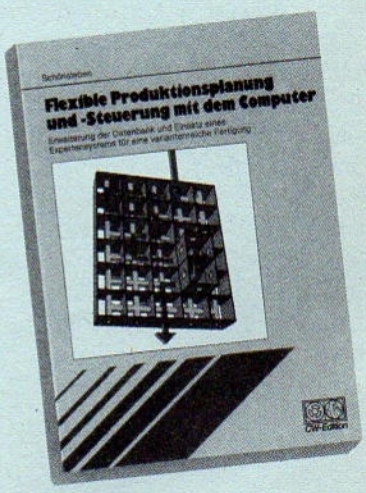
Vorname _____

Straße _____

Postleitzahl/Ort _____

Unterschrift _____

☐ Bitte senden Sie mir Ihr Bücherverzeichnis



Prof. Dr. Paul Schönsleben
Flexible Produktionsplanung und -steuerung mit dem Computer 144 Seiten, München 1985
Best.-Nr.: 1 CW 53-2 DM 58,—

WWW.HOMECOMPUTERWORLD.COM

„Explode“ jagt den Commodore in die Luft!!!

System: C64/16/128
Typ: Spiel
Sprache: Basic

Zwei Spieler nehmen in diesem Denkspiel den Kampf gegeneinander auf. Nur wer die bessere Übersicht hat und trotz andauernder Explosionen alles kontrolliert, hat Chancen zu gewinnen.

„Explode“ vereinigt Action- und Strategiespiel in einem Programm. Es geht darum, ein 8 mal 8 Quadrate großes Feld so zu besetzen, daß es zu Explosionen und Kettenreaktionen kommt, die den Gegner auslöschen. Das Programm ist zwar für Commodore 64-Computer geschrieben, jedoch ohne große Schwierigkeiten mit der ausführlichen Programmbeschreibung an alle anderen Commodore Computer der Serie C16/116/Plus 4 und C128 anzupassen.

Delete Enemy

Nach dem Start meldet sich „Explode“ mit Vorspann und Copyrightserklärung. Danach sind die Namen der beiden Spieler einzugeben. Auf Wunsch printet das Programm eine zweiseitige Spielbeschreibung. Das Spiel beginnt mit der Eingabe des X- und dann des Y-Wertes durch den ersten Spieler. Danach folgt der zweite Spieler, der seine Werte eingibt. Kurzzeitig unterbrochen wird das Spiel durch Explosionen, die zu dauerhaften und, zum Teil, langwierigen Kettenreaktionen führen. Hat ein Spieler gewonnen, gibt der Commodore das bisherige Spielergebnis aus und will wissen, ob man weiterspielt. Je nach Eingabe geht's in die nächste Runde oder das Programm verabschiedet sich.

Spielanleitung:

„Explode“ ist nur auf einem Computer optimal zu spielen. Es wurde für diese Geräte entwickelt. Sinn ist es, alle gegnerischen Felder mit der eigenen Farbe zu besetzen und so das Spiel zu gewinnen. Jeder Spieler setzt pro Durchgang einen Stein auf ein beliebiges, jedoch entweder freies oder bereits mit eigener Farbe besetztes Feld. Je nach Feldlage und Anzahl der bereits dorthingelegten Steine explodiert das Ursprungsfeld und schleudert seine Bruchstücke in alle senkrechten Richtungen. Die getroffenen Felder erhalten die Spielfarbe des aktuellen Spielers, der gerade gesetzt hat. Ihre Werte werden um eins erhöht. Das explodierte Feld erhält den Wert: „Ursprungswert-Wert, der zur Explosion als Minimum erreicht sein muß“.

Explosiv: 2, 3 und 4

Explosionen erfolgen bei folgenden Zahlen:
Eckfelder (1,1/1,8/8,1/8,8) bei zwei oder mehr Steinen.
Randfelder (zum Beispiel 1,2 bis 1,7) bei 3 oder mehr Steinen
Mittelfelder (zum Beispiel 2,6) bei 4 oder mehr Steinen. Die Farbe eines vom Gegner besetzten Steines kann nur dadurch umgedreht werden, daß im Nachbarfeld eine Explosion stattfindet und ein Trümmerstück auf dieses Feld stürzt. Auf bereits besetzte Felder der Gegenfarbe kann nicht gesetzt werden. Felder, die nach einer Explosion den Wert „0“ aufweisen, sind von beiden Parteien wieder besetzbar. Bei einem Spiel ist es wichtig, einen Stein weit ab vom explosiven Geschehen zu halten, damit im Falle einer ungeahnten gegnerischen Kettenreaktion mindestens dieser eine Stein als Basis erhalten bleibt. Sind alle Steine bei einer Kettenreaktion oder einer einfachen Explosion umgedreht worden, hat der aktuelle Spieler gewonnen. Äußerst reizvoll ist das Spiel, wenn man viele Steine auf dem Brett hat, so daß eine Kettenreaktion nicht immer den Verlauf nimmt, den man erwartet hat. Dieses Programm ist hervorragend zum Zeitvertreib zu zweit geeignet und bringt Action in die Kategorie der Denkspiele.

Zeile	Funktion
00001— 00001	Sprung in das Hauptmenü ab Zeile 5000
00002— 00002	Das Feld mit der Koordinate X,Y wird um eins erhöht. Das Feld erhält die Spielernummer des aktuellen Spielers SP.
00010— 00010	Sound bei Explosion ausgeben. Lautstärke (VO,15) voll aussteuern. Cursor nach links oben (Home). Besitzer eines Commodore-Computers der Serie C16 oder C128 sollten die Pokes entweder entfernen (nicht eintippen) oder die Anweisung SOUND benutzen. Nur so gibt es keine Kompatibilitätsprobleme. Der Sound ist allerdings nicht überragend. (Ist ja auch kein Musikprogramm!!!)
00070— 00120	Der Cursor wird für die Ausgabe vorbereitet, indem er an die entsprechende X/Y-Achse gesteuert wird.
00150— 00150	Ist der Wert der Zelle gleich 0, so wird der Spielerwert des Feldes neutralisiert (auf 0 gesetzt) und eine Null weiß auf schwarz ausgegeben.
00160—	Ausgabe des errechne-

WWW.HOMECOMPUTERWORLD.COM

00160 ten Feldwertes, der vorher in einen String umgewandelt wurde. Interessanterweise ist dieser String mindestens zweistellig, so daß mit RIGHT-$(String,1) eine einzige Stelle ausgegeben wird.

00168— Berechnung der Werte
00169 zur Explosion.

00170— Vom Feld mit Koordina-
00170 ten X/Y wird der entsprechende Maximalwert, festgelegt in der M-Matrix, abgezogen. Das Feld erhält den Spielerwert neutral (=0). Falls das Feld einen Wert über 0 hat, erhält es anschließend den Spielerwert des aktuellen Spielers SP.

00172— Die Variablen X und Y
00172 werden in X1 und Y1 als Stammvariablen untergebracht.

00174— Um die vier senkrecht
00180 herumliegenden Felder der Stammvariablen werden die Feldwerte um eins erhöht, falls das Feld innerhalb der Begrenzungen im sichtbaren Bereich liegt. Danach werden die neuen Daten automatisch ausgegeben.

00182— FLAG wird neutralisiert
00182 und dient als Testvariable.

00184— Prüft alle Felder auf Er-
00190 reichen oder Überschreiten des Maximalwertes. Falls Ergebnis positiv, wird FLAG auf 1 gesetzt, was bedeutet, daß eine Explosion fällig ist. Die Koordinaten des Feldes stehen in X1 und Y1 (Stammvariablen). Eine weitere Prüfung findet dann nicht mehr statt. Dies ist der Grund, weshalb eine Explosion in den oberen, linken Feldern schneller entdeckt und ausgeführt wird als eine Explosion im rechten, unteren Teil des Feldes. (Geprüft wird von links oben nach rechts unten).

00194— Ist Flag = 1 so X und Y ist
00194 gleich X1 und Y1. Danach Sprung zum Anfang der Explosionsroutine in Zeile 00170.

00197— Keine weitere Explosio-
00197 nen.

00199— Hier erfolgt die Eingabe.

00200— Die ersten vier Bild-
00204 schirmzeilen löschen und den Cursor nach links oben an den Bildanfang setzen.

00210— Eingabeabfrage für den
00221 jeweiligen Spieler SP: X-Achse.

00222— Ist X im Bereich des
00224 Spielfeldes? Wenn ja, dann weitermachen, wenn nein, dann nächste Eingabe.

00225— Eingabe, die gemacht
00225 wurde, anzeigen.

00230— Wie zwischen 00210 bis
00251 00225 nur mit der Y-Achse.

00255— Die ersten vier Bild-
00260 schirmzeilen löschen und zum Menü zurück.

00299— Hier wird geprüft, ob das
00370 eingegebene Feld schon vom Gegenspieler besetzt ist.

00300— EFLAG=ErrorFlag wird
00300 auf 0 gesetzt.

00310— Wenn der Spielerwert
00310 des Feldes neutral ist, ist die Eingabe o.k., daher Rücksprung ins Menü.

00320— Wenn der Spielerwert
00320 des Feldes dem Wert des Spielers entspricht, der gerade gesetzt hat, so ist die Eingabe o.k., und es folgt ein Rücksprung ins Menü.

00330— Die Eingabe war nicht
00370 möglich. Diese Meldung wird ebenfalls am Bildschirm ausgegeben, das EFlag auf 1 gleichbedeutend mit Eingabefehle gesetzt. Sprung ins Menü erfolgt am Ende.

00500— Testschleife zur Zeitopti-
00540 mierung.

04999— Hauptprogramm von Ex-
05490 plode = Steuerzentrale

05000— Soundwerte und -varia-
05000 blen initialisieren. Benutzer von Rechnern der C16/128-Serien sollten hier einen REMark eingeben und müssen dann ebenfalls nicht die Zeilen 59999 bis 60030 eingeben.

05001— Hintergrund und Rahmen
05001 schwarz machen. Hier sollten C16/116/Plus4 und 128er Benutzer die COLOR-Anweisung nutzen, damit EXPLODE auf diesen Rechnern läuft.

05002— Die Variablen (siehe Va-
05003 riablenerklärungen) werden initialisiert und Großschrift eingestellt.

05005— Die Matrix mit den Maxi-
05005 malwerten des jeweiligen Feldes wird durch einen GOSUB erstellt.

05010— Das erste Bild nach dem
05110 Programmstart, das man am Monitor sieht, wird durch diese Zeilen geprintet.

05120— Hier wartet der C64 auf
05120 einen Tastendruck. Besitzer anderer Commodore-Serien wechseln diesen Befehl mit GETKEY$ oder INPUT aus.

05115— Dieser Poke löscht den
05115 Tastaturzwischenspeicher, so daß bereits gedrückte Tasten nicht berücksichtigt werden. Diese Speicherstelle ist je nach Rechner verschieden. Es ist der Wert einzusetzen, der im Handbuch als Zähler der im Tastaturpuffer stehenden Zeichen bezeichnet wird.

05120— Abfrage nach Spielre-
05210 geln. Falls die Spielregeln erwünscht sind, wird in das Unterprogramm „Spielregeln ausgeben" gesprungen.

05230— Bildschirm löschen
05250 (CLR/HOME) und Abfrage nach den Namen der beiden Spieler.

05390— Sprung zum Unterpro-
05390 gramm, das das Spielfeld aufbaut.

05400— Die Nummer des aktuel-
05400 len Spielers wird um eins erhöht. Ist die Zahl größer zwei (es gibt nur zwei ▶

WWW.HOMECOMPUTERWORLD.COM

Spieler), so ist Spieler 1 an der Reihe und SP daher 1. Zu Anfang des Spieles ist SP=0, so daß immer erst Spieler 1 beginnt. (P=0+1)

05405— Siehe Zeile 05115.

05406— 05410 Sprung in das Unterprogramm „Eingabe".

05420— 05420 Sprung in das Unterprogramm „Eingabe korrekt?"

05430— 05430 Wenn das Errorflag (EFLAG) gesetzt ist, so ist die Eingabe falsch und muß wiederholt werden. Daher Rücksprung zu Zeile 05410 des Hauptmenüs.

05435— 05435 Der eingegebene Feldwert wird um eins erhöht und das Feld erhält den Spielerwert des aktuellen Spielers, egal ob es diesen schon hatte oder nicht.

05440— 05440 Falls der neue Feldwert den Maximalwert erreicht, wird in die Explosionsroutine gesprungen, die die Explosion auslöst und eventuell eine ganze Kettenreaktion selbständig ausführt.

05450— 05450 Liegt keine Explosion vor, wird der um eins erhöhte Feldwert in der Routine „Ausgabe" ausgegeben.

05460— 05480 Ist ZFLAG 0, so war dies der erste Zug. Daher kann noch nicht geprüft werden, ob alle Steine des Gegners vernichtet sind und Spieler 1 somit gewonnen hat, da noch gar keine Steine von Spieler 2 gesetzt wurden. Daher Siegersuchroutine umgehen und ZFLAG auf 1 setzen, da erster Zug ausgeführt.

05490— 05490 Der nächste Spieler kommt, indem an den Anfang des Hauptmenüs in Zeile 05400 gesprungen wird.

06999— 07100 Spielfeld aufbauen. In diesem Unterprogramm baut EXPLODE das Spielfeld auf, in das die Steine gesetzt werden. Sieht man von einer gewöhnlichen FOR . .NEXT-Schleife ab, ist dieses Unterprogramm nur aus einfachen PRINT-Kommandos aufgebaut.

08499— 08570 Ist der Gegenspieler besiegt.

08500— 08500 Das Siegerflag SFLAG wird auf 1 gesetzt, und somit wird davon ausgegangen, daß der aktuelle Spieler gewonnen hat, da der Gegenspieler kein Feld mit eigenen Steinen mehr hat.

08505— 08505 Der aktuelle Gegenspieler wird ermittelt. Man vergleiche hierzu Zeile 05400.

08510— 08560 Alle Felder werden darauf geprüft, ob sie den Spielerwert des Gegenspielers haben. Hat ein Feld diesen Spielerwert, so wird SFDLAG auf 0 gesetzt, da der aktuelle Spieler noch nicht gewonnen hat, und die Suche nach weiteren Feldern mit dem Spielerwert des Gegenspielers abgebrochen.

08570— 08570 Anschließend Rückkehr ins Hauptmenü.

19999— 20220 Ein Spieler hat gewonnen. Der Spielstand wird ausgegeben und gefragt, ob weitergespielt wird.

20005— 20040 Fünf mal eine ansteigende Tonhöhe erklingen lassen. Diese Zeilen müssen entweder mit SOUND-Anweisungen ersetzt oder in REMarks umgewandelt werden, damit Besitzer anderer Commodore-Rechner als dem C64 dieses Programm nutzen können.

20045— 20110 Bildschirm löschen und das aktuelle Spielergebnis printen (Spielernamen mit jeweiliger Punktzahl).

20120— 20150 Frage, ob weitergespielt wird. Falls nein: Bildschirm löschen und „Schade . . ." hinschreiben und Programm beenden. Falls ja: Im Programm weitermachen.

20160— 20210 Alle NULL-Variablen wie nach einem RUN auf 0 setzen. Feldvariablen mit deren Spielerwerten auf 0 setzen und somit neu initialisieren.

20215— 05400 Das Spielfeld neu aufbauen und zum Hauptmenü.

59999— 60030 Sound initialisieren. Diese Routine muß von Besitzern anderer Commodore-Systeme als den C64 nicht abgetippt werden. Im Unterprogramm werden die Variablen für die Soundchipbeeinflussung (siehe Variablen) sowie die Tonart festgelegt.

60999— 61270 Spielregeln: Wie jedes Spiel hat auch EXPLODE eine kleine Routine, in der die Spielregeln stehen.

61999— 62060 M-Matrix anlegen. Bei Explode explodieren die Steine, je nach Lage, bei verschieden hohen Werten. Diese Maximalwerte oder Explosionswerte sind in der zugehörigen M(X/Y)-Variable zu finden. Diese Variablen müssen daher zu Beginn des Spieles mit diesen Werten beschrieben werden. Dies geschieht mit einer kleinen READ DATA Routine, die die jeweiligen Werte aus der nachfolgenden DATA-Matrix liest und den jeweiligen M-Matrixwerten zuordnet.

62100— 62170 In diesem Programmteil sind die DATA-Zeilen untergebracht, aus denen die M-Matrix initialisiert wird. Es ist aus der Form der DATA's leicht zu erkennen, welches Feld bei welcher Zahl explodiert.

Variablen:

Variable	Funktion
a$,a, xe$,i ye$,ei$, ii,	Eingabe der Zählvariablen ohne große Bedeutung (werden nur kurzzeitig eingesetzt).

WWW.HOMECOMPUTERWORLD.COM

	Wert der oftgebrauchten Zahl 1. Der Computer verarbeitet Variablen schneller als festgeschriebene Zahlen. Dieser Geschwindigkeitsvorteil macht es sinnvoll, oft benutzte Zahlen in Variablen umzuwandeln.
z	Zero = Null. Wert für 0.
X	X-Achse, die ausgegeben wird.
Y	Y-Achse, die ausgegeben wird.
X1	Stammvariable der X-Achse (Echtwert).
Y1	Stammvariable der Y-Achse (Echtwert)
sp	Aktueller Spieler
gs	Aktueller Gegenspieler
n	Neun. Wert für die Zahl 9.
sm	54272. Hier beginnen die SID-Adressen (Sound-Chip).
sz	Speicherstelle für Tonartbeeinflussung.
hf	Speicherstelle für Frequenzbeeinflussung des Tones.
vo	Speicherstelle für Lautstärkenbeeinflussung.
nz	Wert für 19. Entspricht bei CHR$(nz) einem HOME.
FLAG	Wenn diese Variable 0 ist, so bedeutet das: Kein Feldwert erreichte oder überschritt den Maximalwert. Daher keine Explosion.
EFLAG	Wenn diese Variable 1 ist, so bedeutet dies: Eingabe nicht möglich, da Feld schon vom Gegenspieler besetzt ist. Wenn diese Variable 0 ist, so bedeutet dies: Eingabe o.k., Programm läuft weiter.
ZFLAG	1 bedeutet: Dies ist mindestens der zweite Zug. 0 bedeutet: Dies ist der allererste Zug. Daher darf noch keine Siegesermittlung gemacht werden, denn sonst gewinnt ungerechtfertigterweise jedesmal Spieler 1.
SFLAG	1 bedeutet: Der Gegenspieler hat kein Feld mehr mit seiner Farbe besetzt und daher verloren. 0 bedeutet: Noch ist kein Sieg in Sicht.
SP$(1)	Name des ersten Spielers.
SP$(2)	Name des zweiten Spielers.
SP(1)	Punktezahl des ersten Spielers.
SP(2)	Punktezahl des zweiten Spielers.
Q(1,1 bis 8,8)	In dieser Matrix steht der Wert des Feldes an der jeweiligen Stelle am Bildschirm (in X- und Y-Achse).
M(1,1 bis 8,8)	In dieser Matrix steht der Wert, den ein Q-Feld maximal benötigt (oder überschreiten muß), um zu explodieren. Ein explodierendes Feld wird mit dem Betrag des korrespondierenden M-Feldes <gleicher X- und Y-Achsenwert) subtrahiert.
W(1,1 bis 8,8)	In dieser Matrix steht, ob das Feld Q(X,Y) neutral ist (0) und daher von beiden Spielern besetzt werden kann oder ob es von einem Spieler (1 oder 2) besetzt ist. Bei der Eingabe prüft Explode mit diesem Wert, ob die Daten korrekt waren oder ob ein Fehler vorliegt.

In Maschinensprache tragen Testspeicherstellen den Namen FLAG. Um den Übergang von Basic zu Maschinensprache zu erleichtern, behält „Explode" diese Schreibweise bei. Oftmals wird ein Byte, bestehend aus 8 Bits, in B Flag's geteilt, da jedes einzelne Byte den Wert 1=wahr oder 0=falsch erreicht. Damit spart man viel Speicherplatz. Allerdings kostet diese Art des „Flaggen-Setzens" viel Zeit, da man zum Prüfen des FLAGs das richtige Bit herausfiltern, extra Rechenoperationen ausführen muß.

Damit sind alle Funktionen und Variablen des Programmes ausführlichst erklärt. Ich hoffe, daß vor allem Basic-Beginner nach diesen Erläuterungen die einzelnen Programmschritte verstehen und in eigenen Programmen anwenden können.

F. Bastian

```
  1 goto 5000:rem hauptprogramm                    638
  2 q(x,y)=q(x,y)+e:w(x,y)=sp                     2784
  9 rem ausgaberoutine
 10 poke vo,15:poke sz,33:poke hf,5:poke          3917
    sz,32:print chr$ (nz)
 70 for i=e to x:print "[2right]";:next :         3077
    print "[5right,4down]";
120 for i=e to y:print "[2down]";:next :if        2434
    sp then print "[rvs]";
150 if q(x,y)=z then print "[off]0";:w(x,        2933
    y)=z:return
155 if sp=e then print "[off]";                   1416
160 print right$ ( str$ (q(x,y)),e);:            1863
    return
168 :
169 rem explosonsroutine
170 q(x,y)=q(x,y)-m(x,y):w(x,y)=z:if q(x,        5930
    y)>z then w(x,y)=sp
171 gosub 10                                      431
172 x1=x:y1=y                                    1042
174 x=x1-e:if x>z then gosub 2                   1755
176 x=x1+e:if x<n then gosub 2                   1695
178 x=x1:y=y1-e:if y>z then gosub 2              2113
180 y=y1+e:if y<n then gosub 2                   1701
182 flag=0                                        334
184 for x=1 to 8:for y=1 to 8:if q(x,y)>=        6867
    m(x,y) then flag=1:y1=y:x1=x:y=8:x=8
190 next :next                                    256
194 if flag then x=x1:y=y1:goto 170              2204
197 return                                        142
198 :
199 rem eingaberoutine
200 print "[home,38spaces]"                       1448
201 print "[38spaces]"                           1434
202 print "[38spaces]"                           1434
203 print "[38spaces]"                           1434
204 print "[home]";                               493
210 print "spieler: ";sp$(sp)                    1935
220 print "ihre eingabe x-achse (1-8):           2413
    [2spaces]";
```


WWW.HOMECOMPUTERWORLD.COM

```
  221 get xe$                                          391
  222 x=val (xe$)                                      735
  224 if x<1 or x>8 then goto 221                     1366
  225 print x                                          329
  230 print "ihre eingabe y-achse (1-8):";           2727
  232 get ye$                                          393
  240 y=val (ye$)                                      752
  250 if y<1 or y>8 then goto 232                     1424
  251 print y                                          331
  255 print "[home,38spaces]"                        1448
  256 print "[38spaces]"                             1434
  257 print "[38spaces]"                             1434
  258 print "[38spaces]"                             1434
  260 return                                           142
  298 :
  299 rem eingabe moeglich ?
  300 eflag=0                                          482
  305 print "[off]";                                  491
  310 if w(x,y)=0 then return                         1252
  320 if w(x,y)=sp then return                        1584
  330 rem feld war nicht moeglich !
  350 print "[home]eingabe nicht moeglich
       !!!"                                           2963
  355 eflag=1                                          546
  360 for i=1 to 300:next i                            926
  365 print "[home,36spaces]"                         1349
  370 return                                           142
  500 e=1:ti$="000000"                                1272
  510 for i=1 to 1000                                  810
  520 if e=1 gosub 197                                1103
  530 next                                             130
  540 print ti:end                                     575
 4998 :
 4999 rem hauptprogramm
 5000 gosub 60000:rem soundinit                        721
 5001 poke 53280,0:poke 53281,0:print "[grn]
      "                                               1964
 5002 e=1:z=0:n=9:nz=19                               2167
 5003 print chr$ (142):rem grosschrift                 833
 5005 gosub 62000:rem m-matrixinit                     729
 5010 rem erstes bild-5200
 5020 print "[clr] e x p l o d e[3spaces]-
      [3spaces]d a s[3spaces]s p i e l"               2226
 5030 print :print :print "willkommen zu
       [rvs]explode[off], dem neuen "                 4083
 5040 print "denkspiel fuer den commodore
       64."                                           3198
 5050 print                                           153
 5060 print "[down,rvs]explode[off] wurde
       spezell fuer computer"                         3915
 5080 print "konzipiert und nutzt die
       vorteile"                                      3673
 5090 print "des commodore 64 gegenueber"             2612
 5100 print "einem spielbrett voll aus !!!"           3141
 5110 print :print :print                              733
 5115 poke 198,0                                      870
 5120 print "wollen sie erklaerungen ? (j/n)
       :";                                            4052
 5130 get ei$:if ei$="" then 5130                     1745
 5140 if ei$<>"n" and ei$<>"N" and ei$<>"j"
      and ei$<>"J" then goto 5130                     4052
 5210 if ei$="j" or ei$="J" then gosub
      61000:rem spielregeln                           1908
 5220 rem spielernameneingbe
 5230 print "[clr]"                                    316
 5235 print "bitte geben sie ihre namen ein
       :[2down]"                                      2995
 5240 input "[2down]1. spieler :";sp$(1)              1992
 5250 input "2. spieler :";sp$(2)                     2084
 5390 gosub 7000:rem spielfeld aufbauen                646
 5400 sp=sp+1:if sp>2 then sp=1                       2112
 5405 poke 198,0                                      870
 5410 gosub 200:rem eingabe                           725
 5420 gosub 300:rem war eingabe moeglich ?            727
 5430 if eflag=1 then goto 5410:rem war nich
      t moeglich                                      1702
 5435 q(x,y)=q(x,y)+1:w(x,y)=sp                       2779
 5440 if q(x,y)>=m(x,y) then gosub 170:goto
      5460:rem explosionsroutine                      3675
 5450 gosub 10                                        431
 5460 if zflag=0 then zflag=1:goto 5480:rem
      erster zug                                      2665
 5470 gosub 8500:rem gegner besiegt ?                 668
 5480 if sflag=1 then goto 20000:rem sieger           1569
 5490 goto 5400:rem weiter gehts !                    654
 6998 :
 6999 rem spielfeld aufbauen
 7000 print "[clr]"                                    316
 7010 print "[5down]";                                620
 7020 print "[5right]";                              1088
 7030 print "  B1B2B3B4B5B6B7B8B "                    2042
 7040 for i=1 to 8                                    695
 7050 print "[5spaces]CCCCCCCCCCCCCCCCCCC"           3095
 7060 print "[3spaces]";i;"B B B B B B B B
       B";i                                          2185
 7070 next i                                          276
 7080 print "[5spaces]CCCCCCCCCCCCCCCCCCC"           3095
 7090 print "[6spaces]B1B2B3B4B5B6B7B8B "            2536
 7092 print "[home,10down,30right,rvs]
      explode[off]"                                   5314
 7093 print "[down,29right,rvs]das spiel
      [off]"                                          5083
 7094 print "[down,27right,rvs]by f. bastian
      [off]"                                          5190
 7100 return                                          142
 8498 :
 8499 rem spieler besiegt ?
 8500 sflag=1                                         560
 8505 gs=sp+1:if gs>2 then gs=1                       2220
 8510 for x=1 to 8                                    725
 8520 for y=1 to 8                                    727
 8530 if w(x,y)=gs then x=8:y=8:sflag=0               3450
 8540 next y                                          308
 8560 next x                                          306
 8570 return                                          142
19998 :
19999 rem weiterspielen ?
20000 rem
20005 for i=1 to 5                                    854
20010 for mz=1 to 200 step 4                         1207
20020 poke vo,7:poke sz,33                           1600
20030 poke hf,mz:poke sz,32                          1676
20040 next mz:next i                                  679
20045 print "[clr]"                                   316
20046 print "[2spaces]e x p l o d e[2spaces]
      -[2spaces]d a s[2spaces]s p i e l"             2342
20047 print
      "-----------------------------------"          6312
20050 print "[2down]spieler ";sp$(sp);" hat
       gewonnen !!!"                                  3738
20060 sp(sp)=sp(sp)+1                                1614
20070 print "[4down]";                                769
20080 print "punkte[7spaces]spieler"                  2019
20083 print "---------------------------"             5037
20085 print                                           153
20090 print "[2spaces]";sp(1);"[10spaces]";           2141
      sp$(1)
20100 print "[2spaces]";sp(2);"[10spaces]";           2175
      sp$(2)
20110 print "[4down]";                                769
20120 print "[rvs]weiterspielen (j/n) [off]"         2287
      ;
20130 get a$:if a$="" then goto 20130                1620
20140 if a$="n" or a$="N" then print chr$             4147
      (147);"schade ...":end
20150 if a$<>"j" and a$<>"J" then goto 20130         2372
20160 sflag=0:zflag=0:sflag=0                        1553
20170 for x=1 to 8                                    725
20180 for y=1 to 8                                    727
20190 q(x,y)=0:w(x,y)=0                              1771
20200 next y                                          308
20210 next x                                          306
20215 gosub 7000                                      575
20220 goto 5400                                       583
59998 :
59999 rem sid initialisieren
60000 sm=54272:hf=sm+1:vo=sm+24:sz=sm+4              3576
60010 for i=sm to sm+24:poke i,0:next                2316
60020 poke sm+5,8:poke sm+5,8                        1399
60030 return                                          142
60998 :
60999 rem spielregeln
61000 print "[clr]spielregeln zu [rvs]              2474
      explode[off]:"
61010 print                                           153
61020 print "zuerst baut der computer am              3860
       bildschirm"
61030 print "ein 8 mal 8 felder grosses               3677
       spielfeld"
61040 print "auf."                                     829
61050 print                                           153
```


WWW.HOMECOMPUTERWORLD.COM

```
61060 print "dieses spielfeld gilt es mit      4133
       steinen zu"
61070 print "besetzen. jeder spieler hat       3251
       seine "
61080 print "eigene farbe und je zug kann er   3713
       nur"
61090 print "einen stein legen.":print          2220
61100 print "steine koennen nur auf leere      3189
       felder"
61110 print "oder auf besetzte felder der       3421
       eigenen"
61120 print "farbe gelegt werden !":print :     3844
      print :print :gosub 61260:print "[clr]
      "
61130 print "sinn und zweck des spieles ist,    3940
       die "
61140 print "steine der gegenseite durch        3976
       geschickte"
61150 print "kettenreaktionsexplosionen zu      3957
       eigenen"
61160 print "steinen zu machen, so dass der     3620
       andere"
61162 print "spieler keine steine mehr hat."    3275
      :print
61190 print "bei eckfeldern tritt die           3717
       explosion nach"
61200 print "zwei steinen, bei den              4001
       restlichen rand-"
61210 print "feldern bei 3 und bei den          3584
       innenfeldern"
61220 print "bei 4 steinen auf.":print          1938
61222 print "explodiert ein feld, werden        3649
       steine in 4"
61223 print "andere richtungen geshleudert."    3125
      :print
61224 print "die felder, auf denen steine       3876
       fallen,"
61225 print "nehmen die farbe des               3963
       ursprungsfeldes an"
61226 print "und koennen ihrerseits weiter-"    3175
61227 print "explodieren.":print                1892
61230 print "hat ein spieler kein               3907
       einzigsfeld mit[2spaces]"
61240 print "eigener farbe, so hat er           3078
       verloren."
61250 print                                      153
61260 input "<return> druecken ";a$              2254
61270 return                                     142
61998 :
61999 rem m-matrix initialisieren
62000 for x=1 to 8                               725
62010 for y=1 to 8                               727
62020 read a                                     265
62030 m(x,y)=a                                   869
62040 next y                                     308
62050 next x                                     306
62060 return                                     142
62100 data 2,3,3,3,3,3,3,2                      1827
62110 data 3,4,4,4,4,4,4,3                      1912
62120 data 3,4,4,4,4,4,4,3                      1912
62130 data 3,4,4,4,4,4,4,3                      1912
62140 data 3,4,4,4,4,4,4,3                      1912
62150 data 3,4,4,4,4,4,4,3                      1912
62160 data 3,4,4,4,4,4,4,3                      1912
62170 data 2,3,3,3,3,3,3,2                      1827
```

„Explode" ist Denkspiel und Basic-Kurs in einem.

Die Erde als Abfalleimer

System : C16/116/Plus 4
Typ : Spiel
Sprache: Basic

In der Zukunft versucht eine fremde Macht, ihren radioaktiven Müll auf der Erde endzulagern. Laufend schütten Raumschiffe ihre verbrauchten Brennstoffe auf Städte, Seen und Gebirge. Mit einem Schutzschirm muß man die anfliegenden Partikel einsammeln.

New York, 2605. Aliens benutzen die Erde als gigantischen Mülleimer. Um größte Schäden zu vermeiden, besitzt jede Großstadt einen Abwehrschirm. Der Spieler steuert diesen Abwehrschirm. Brechen mehrere Teilchen durch, explodiert New York mitsamt Einwohnern und der Abwehranlage. Überschreitet man den High-Score, gibt man seinen Namen ein. „Stardust" speichert diesen in der High-Score Tabelle auf Kassette ab.

So wurde programmiert

Zeile 10 bis 60
— Highscore laden?
— Spiel mit Joysticktastatur?

Zeile 70 bis 143
Bildschirmgraphik zeichnen
Zeile 150 bis 160
Positionen von Spieler und Staubteilchen festlegen.
Zeile 170
Spieler zeichnen

Zeile 180 bis 190
Joystick/Tastatur — Routine
Zeile 200 bis 210
Bewegung des Spielers, alte Position löschen
Zeile 220 bis 240
— Zeichnen der Staubteilchen
— Nächste Positionen auf Crash untersuchen
— Löschen der alten Positionen
Zeile 250 bis 270
— Reaktionsroutine der Staubteilchen:
  — Staubteil abgewehrt
  — Staubteil in der Stadt angekommen
Zeile 280 bis 320
Effekte und Reaktionsverarbeitungen
Zeile 330 bis 390

Endroutine:
— Score anzeigen
— Highscore?
— Highscoreinhaber anzeigen
— Highscore abspeichern?
— Spiel von neuem beginnen

WWW.HOMECOMPUTERWORLD.COM

Wichtige Variablen:
HS = Highscore
N$ = Name des Highscoreinhabers
SC = Score
JT = Joystick/Tastatur-Variable
C = Spielposition
M = Richtung der Spielerbewegung
A(I) = Position der Staubteilchen
B(I) = alte Position
Z = Zählwerkvariable

M. Goes

```
10 gosub 20:goto 30                                          908
20 scnclr :color 0,1:print "[rvs,red,                       4131
   9spaces]radioactive stardust[11spaces]
   "
25 print "[2down] by martin goes":return                    2246
30 print "[blu,2down,3spaces]shall i load                   4164
    the highsore (y/n)[3spaces]":get key
   a$
40 if a$="y" then open 1,1,0,"highscore":                   4241
   input#1,hs:input#1,n$:close 1
50 print "[2down,grn] press k for                           6082
    keyboard or any other key[3spaces]for
    joysick !":get key a$
60 if a$="k" then jt=2:else jt=1                            2450
70 sc=0:color 0,1:color 1,9,7:color 2,4,                    4430
   3:color 3,3,4:graphic 3,1
80 char 1,0,18," B[3spaces]B[19spaces,                      2552
   shift -,2spaces]B[2spaces,3cbm +,
   3spaces,cbm -] "
90 char 1,0,19," [cbm +,3spaces,cbm +,                      3651
   11spaces,3cbm +,cbm -,4spaces,cbm +,
   2spaces,cbm +,2spaces,2cbm -,cbm +,
   2spaces,3cbm -]"
100 char 1,0,20,"[3cbm +] [3cbm +,4spaces,                  4752
   shift -,5spaces,4cbm -,3spaces,6cbm +]
    [3cbm +,2spaces,2cbm +,cbm -]"
110 char 1,0,21,"[2cbm -,cbm +] [2cbm -,                    4464
   cbm +,4spaces,cbm +,5spaces,3cbm +,
   cbm -,3spaces,2cbm -,cbm +,2cbm -,
   cbm +] [2cbm -,cbm +,2spaces,3cbm -]"
120 char 1,0,22,"[7cbm +,3spaces,3cbm +,                   5129
   4spaces,4cbm -,3spaces,6cbm +]
    [3cbm +,2spaces,2cbm +,cbm -]"
130 char 1,0,23,"[2cbm +]O[cbm y]P[2cbm +,                  6482
   3cbm r,cbm +]O[cbm +]RRRR[cbm +]OP
   [cbm -,3cbm r,2cbm +]OP[2cbm +,cbm i,
   cbm +]O[cbm +,2cbm r,2cbm +,cbm -]"
140 char 1,0,24,"[40cbm u]"                                 6128
143 draw 1,50,0 to 110,0 to 111,1 to 49,1                   5425
   to 48,2 to 112,2 to 113,3 to 47,3 to
   46,4 to 114,4
150 vol 8:c=20:char 2,c+1,16,"[cbm i]"                       1973
160 for i=1 to 5:a(i)=int ( rnd (0)*30)+                     4232
   65:b(i)=b(i-1)+20
170 char 2,c,16,"[cbm i]":on jt goto 180,                    2204
   190
180 if joy (1)=3 then m=1:goto 200:else if                   4587
   joy (1)=7 then m=-1:goto 200:else 220
190 get a$:if a$="[right]" then m=1:goto                     4702
   200:else if a$="[left]" then m=-1:else
   220
200 if c+m<3 or c+m>36 then 320                              2210
210 c=c+m:char 2,c-2*m,16," "                                1974
220 draw 3,a(i),b(i):next i                                  1365
230 for i=1 to 5:b(i)=b(i)+4:locate a(i),                    4473
   b(i):if rdot (2)<>0 then 250
240 draw 0,a(i),b(i)-8:goto 170                              1916
250 if rdot (2)=1 then for z=1000 to 200                     4635
   step -50:sound 3,z,2:next :gosub 310:
   goto 280
260 if rdot (2)=2 then draw 0,a(i),b(i) to                   6264
   a(i),b(i)-8:a(i)=int ( rnd (0)*80)+40:
   b(i)=40
270 char 2,c,16,"[cbm i]":char 2,c-m,16,"                    5263
   [cbm i]":sound 1,900,7:sc=sc+1:goto
   170
280 for f=1 to 10:y=int ( rnd (0)*60)-30:x                   4121
   =int ( rnd (0)*60)-30
290 draw 0,a(i),b(i) to a(i)+x,b(i)+y:next                   3851
   :for o=1 to 20:get a$:next
300 for r=1 to 500:next :goto 330                            1444
310 for t=1 to 50:color 0,2:for o=1 to 10:                   3966
   next :color 0,1:next :return
320 m=-m:goto 210                                            1010
330 graphic 0:gosub 20                                        699
340 print "[4down,wht,2spaces]score:";sc:                    5627
   if hs<sc then hs=sc:input "[2down]
    enter your name please";n$
350 if n$="" then n$="---"                                   1851
360 print "[3down,2spaces]highscore: ";n$,                   2452
   hs
370 print "[3down,pur] press s to save the                   3140
    highscore"
380 print " press any other key to                           5461
    continue":get key a$:if a$="s" then
   390:else 50
390 open 1,1,1,"highscore":print#1,hs:                       3813
   print#1,n$:close 1:goto 50
```

Feuerwehr ist out, Müllabfuhr ist in, bei „Radioactive Stardust"

Die Erde ist für alle da...

Darauf hoffen Millionen, die hungrig sind und kein Dach über dem Kopf haben, für die es keine Ausbildungsmöglichkeit, keinen Arbeitsplatz, keine Krankenversicherung und keine Altersrente gibt. Was der einzelne bei uns tun kann? Den Lebensstil vereinfachen. Alle Bemühungen nach sinnvoller Entwicklungshilfe unterstützen. Sich dafür einsetzen, um mehr Gerechtigkeit und Menschenwürde für andere zu schaffen. Kostenlose Information von »Brot für die Welt«, Postfach 476, 7000 Stuttgart 1. Spendenkonto: 500 500-500 beim Postscheckamt Köln.

Brot für die Welt



WWW.HOMECOMPUTERWORLD.COM

Bandwürmer im Computer!

System : C16/116/Plus 4
Typ : Spiel
Sprache: Basic

Nach Start mit RUN lädt man auf Wunsch eine High-Score-Tabelle, gibt Level und Eingabegeräte (Joystick oder Tastatur) an. Danach erscheint das Spielfeld. Blaue, nahrhafte und gelbe, giftige Bytes liegen herum. Wer es schafft, alle blauen Punkte abzugrasen, ohne sich dabei in den eigenen Schwanz zu beißen oder sich zu vergiften, erreicht den nächsthöheren Level.

Kleine Würmer „pflanzt" dieses Programm in die C16-Chips ein. Ab und zu fressen die Tierchen ein Byte und wachsen dadurch. Ziel ist es, einen Wurm zu steuern und viele Bytes zu futtern.

„Wurmi im Detail"

Zeile 10 bis 80: Vorspann
— Highscores einladen?
— Schwierigkeitsgrad eingeben
Zeile 90
— Anfangsrichtung für die Bewegung des Würmchens bestimmen
Zeile 100 bis 120
— Bildschirmgraphik zeichnen
Zeile 130
— Würmchen zeichnen
Zeile 140 bis 180
— Joystick/Tastaturroutine
Zeile 190 bis 200
— Würmchen fortbewegen
— Überprüfen der neuen Position auf Kollision mit irgend etwas untersuchen
Zeile 210 bis 220
— Score erhöhen
— gefressenen Punkt löschen
— neue Punkte zeichnen
Zeile 230
— Würmchen tot-Effekt
Zeile 240 bis 345
— Endroutine:
  — Score anzeigen
  — Highscore gebrochen?
  — Highscores abspeichern?
  — Nochmal spielen?
Zeile 350
— Warteschleife (vom Schwierigkeitsgrad abhängig)
Zeile 360 bis 370
— Positionen für neue Gift- und Bonuspunkte festsetzen

Wichtige Variablen

H (0) = Highscore im jeweiligen Schwierigkeitsgrad
S (O) = Score im jeweiligen Schwierigkeitsgrad
JT = Joystick/Tastaturvariable
M,N = Richtungsvariablen des Würmchens
A,B = Positionsvariablen des Würmchens
C,D = Positionsvariablen für Liftpunkte
E,F = Bonuspunkte

Martin Goes

```
10 print "[clr]highscores einladen ?          4861
   (j/n)":get key a$:if a$="j" then 20:
   else 40
20 open 1,1,0:for hr=0 to 3:input#1,          3697
   n$(hr):input#1,h(hr):next :close 1
40 color 0,2:vol 8:scnclr :print "[rvs,       3182
   red,flashon,11spaces]w u e r m c h e n
   [12spaces]"
50 print "[2down] j fuer joystick             4607
   [25spaces]c fuer cursortastatur"
60 get key a$:if a$="c" then jt=1:else jt      2026
   =2
70 for o=0 to 3:s(o)=0:next                   1850
80 m=0:n=0:input "[blu,2down]                 6428
   verlangsamungsgrad 0-3";o:if o<0 or o>
   3 or o<>int (o) then 80
90 t=int ( rnd (0)*4)+1:if t=1 then m=1:      6252
   else if t=2 then m=-1:else if t=3 then
   n=1:else n=-1
100 graphic 3,1:color 4,1:a=80:b=100          2315
110 color 0,3,5:color 2,2:color 3,7,2         2486
120 color 1,1:for i=1 to 100:gosub 360:       5013
   draw 2,c,d to c,d+1:draw 3,e,f to e,f+
   1:next
130 gosub 350:draw 1,a,b-n to a,b:get a$:     4031
   if a$="" and joy (1)=0 then 190
140 on jt goto 150,170                        1020
150 if a$="[left]" then m=-1:n=0:goto 190:    5429
   else if a$="[right]" then m=1:n=0:goto
   190
160 if a$="[up]" then n=-2:m=0:else if a$=    3373
   "[down]" then n=2:m=0
170 if joy (1)=7 then m=-1:n=0:goto 190:      5618
   else if joy (1)=3 then m=1:n=0:goto
   190
180 if joy (1)=1 then n=-2:m=0:else if joy    3536
   (1)=5 then n=2:m=0
190 a=a+m:b=b+n:locate a,b:if a<0 or a>       5238
   159 or b<0 or b>199 then 230:else if
   rdot (2)=0 then 130
200 if rdot (2)=2 or rdot (2)=1 then 230:     3255
   else if rdot (2)=3 then 210
210 s(o)=s(o)+1:draw 0,a,b+1 to a,b-1:        3446
   sound 1,600,5
220 gosub 360:draw 2,c,d to c,d+1:draw 3,     3891
   e,f to e,f+1:goto 130
230 sound 3,300,40:for i=1 to 50:color 0,     2998
   1:color 0,2:next :scnclr
240 graphic 0:color 0,1:print "[clr,          4592
   flashon,cyn,6spaces]sie erreichten "
   s(o)" punkte"
245 for qw=1 to 10:get t$:next                1282
250 if s(o)>h(o) then h(o)=s(o):input "       4610
   [2down]wie heisst du";n$(o)
260 print "[rvs,wht,10spaces]h i g h s c o    2055
   r e s[11spaces]"
270 print :print :for o=0 to 3:if n$(o)=""    4127
   then n$(o)="---":goto 280
280 print "[red]";n$(o),o,h(o):next           1765
290 print "[2down,wht,3spaces]nochmal         3367
   spielen - taste druecken[4spaces]"
300 print "[3spaces]highscores saven - s      5300
   druecken":get key a$:if a$<>"s" then
   m=0:n=0:goto 40
310 input "geben sie die zaehlwerknr. ein"    2860
   ;k
320 open 1,1,1:for hr=0 to 3:print#1,         3779
   n$(hr):print#1,h(hr):next :close 1
340 print "[down,2left]spulen sie die         3946
   datasette auf"k"zurueck"
345 print "[rvs,7spaces]- druecken sie         3891
   eine taste -[6spaces]":get key a$:
   goto 40
350 for i=0 to o*10:next :return              1723
360 c=int ( rnd (0)*159):d=int ( rnd (0)*     2939
   199)
370 e=int ( rnd (0)*159):f=int ( rnd (0)*     3510
   199):return
```

Je länger desto Wurm: „Würmchen"

WWW.HOMECOMPUTERWORLD.COM

Das Schwert der Samurai

System : C16/Plus4
Typ : Spiel
Sprache: ASM

Krieg droht im Reich der aufgehenden Sonne. Grund: Das Zauberschwert ist verschwunden. Nur ein mutiger Karate-Kämpfer kann Blutvergießen und Leid verhindern.

Das Versteck des Schwertes ist auf einer Karte verzeichnet. Diese Karte wurde in 60 Teile zerrissen und liegt in einem dunklen Labyrinth. Mit Joystick in Port 2 steuert man durch die Räume. Hier und da liegen Reisschalen und andere, punktesteigernde Gegenstände. Gefährlich sind feindliche Karateka. Sofort greifen diese an. Jetzt gilt es zu kämpfen. Alle Handlungen der Figur steuert man über den Joystick/Feuerknopf:

links	— geht nach links
rechts	— geht nach rechts
oben	— geht nach oben
unten	— geht nach unten
Feuer/links-oben	— Sprung links
Feuer/rechts-oben	— Sprung rechts
Feuer/links-unten	— tritt links
Feuer/rechts-unten	— tritt rechts
Feuer/links	— Boxhieb links
Feuer/rechts	— Boxhieb rechts

„RUN/STOP" unterbricht das Spiel so lange, bis man erneut diese Taste drückt. Druck auf (RETURN) beendet den Spieldurchgang (Aufgabe). Aus Platzgründen sind die DATA-Basic-Lader in 7 Teile aufgegliedert. Man tippt und speichert alle 7 Programme. Danach lädt man einen Teil nach dem anderen, startet die Programme mit RUN und geht danach (kurze Wartezeit) in den MONITOR. Dort gibt man für die unterschiedlichen Teile ein:

1.) T 3932,3FFF,1000
S„SL1",(1oder8),1000,16CD
2.) T 3932,3FFF,16CD
S„SL2",(1oder8),16CD,1D9A
3.) T 3932,3FFF,1D9A
S„SL3",(1oder8),1D9A,2467
4.) T 3932,3FFF,2467
S„SL4",(1oder8),2467,2B34
5.) T 3932,3FFF,2B34
S„SL5",(1oder8),2B34,3201
6.) T 3932,3FFF,3201
S„SL6",(1oder8),3201,38CE
7.) T 3932,3FFF,38CE
S„SL7",(1oder8),38CE,3DE0

Zwischenzeitlich muß man ins Basic zurück, um die Basic-Datalader einzuladen und zu starten. Mit MONITOR gelangt man wieder in den Monitor des C16/Plus4. Sind alle Programme in dieser Art bearbeitet, lädt man vom Monitor die 7 „SL"-Files:

L„SL1",8
L„SL2",8
L„SL3",8
und so weiter bis
L„SL7",8

Mit S„SHOGUN", (1oder8),1001, 3DE0 speichert man die Daten als ein einziges Programm auf Diskette/Kassette. Danach schaltet man den Computer aus, dann an und lädt „Shogun". RUN startet das Spiel.

Olaf Panz

```
 0 rem      shogun lader 1
 1 rem
 2 rem      (c)   1987  by
 3 rem
 4 rem      olaf panz
 5 rem
 6 rem      drosselgasse  4
 7 rem
 8 rem      2095 marschacht
 9 rem
10 poke 56,56:clr :for t=14642 to 16382:    6496
   read a$:poke t,dec (a$):pr=int ((pr+t*
   dec (a$))/256)+t:next t
11 if pr<>19062 then print "Fehler in       2808
    datas !"
12 data 00,0c,1c,c1,07,9e,31,35,37,35,32,   6230
   00,00,00,d8,88,23,07,06,0c,0c
13 data 0c,0c,08,88,c0,c0,60,60,60,60,20,   5826
   03,06,06,02,03,07,07,0f,00,80
14 data 40,80,00,c8,f0,a0,07,03,03,03,03,   5913
   06,0c,0e,80,80,c0,c0,60,60,20
15 data 30,00,01,02,01,00,13,0f,05,c0,60,   5471
   60,40,c0,e0,e0,f0,01,01,03,03
16 data 06,06,04,0c,e0,c0,c0,c0,c0,60,30,   6052
   70,01,12,1b,09,07,06,03,01,c0
17 data a4,6c,c8,f8,30,60,40,03,07,07,0e,   6389
   0e,02,02,06,c0,e0,70,38,38,20
18 data 20,30,03,06,06,02,03,07,07,0f,00,   6279
   80,40,80,00,fe,f2,80,07,03,07
19 data 06,0e,18,18,1c,80,80,c0,60,60,60,   6219
   60,70,00,01,02,01,00,7f,4f,01
20 data c0,60,60,40,c0,e0,e0,f0,01,01,03,   6056
   06,06,06,06,0e,e0,c0,e0,60,70
21 data 18,18,38,0f,5e,bf,bf,7f,1e,0f,00,   5913
   00,7f,fe,c0,f8,f0,00,00,00,00
22 data 00,00,00,00,00,00,00,00,00,00,00,   5084
   00,00,00,00,fe,7f,03,1f,0f,00
23 data 00,f0,7a,fd,fd,fe,78,f0,00,00,00,   5362
   00,00,00,00,00,00,00,00,00,00
24 data 00,00,00,00,00,00,00,00,00,00,00,   5235
   00,30,68,64,28,30,7f,fe,fc,00
25 data 00,40,40,c0,80,04,74,7d,3f,3f,38,   5678
   30,30,20,30,fc,c0,00,00,00,00
26 data 00,00,00,00,02,02,03,01,20,2e,0c,   6162
   16,26,14,0c,fe,7f,3f,3f,03,00
27 data 00,00,00,00,00,be,fc,fc,1c,0c,0c,   5500
   04,0c,00,20,01,00,10,01,01,01
28 data 80,00,00,04,c0,a0,90,a0,00,01,03,   5583
   03,03,03,03,01,c0,c0,fc,f0,c0
29 data c0,f9,ff,00,00,00,14,a9,7e,3c,18,   6181
   00,64,64,ff,7f,6c,4c,00,18,3c
30 data ff,24,ff,66,3c,18,18,24,42,42,42,   6078
   42,24,18,0c,14,24,04,04,04,04
31 data 04,0c,12,12,02,04,08,10,1e,1c,02,   5878
   02,0c,0c,02,02,1c,24,24,24,3f
32 data 04,04,04,04,1e,10,10,1c,02,02,12,   5588
   0c,0e,11,20,20,3e,21,21,1e,3e
33 data 02,04,08,10,10,10,10,1c,22,22,1c,   6367
```


WWW.HOMECOMPUTERWORLD.COM

```
   1c,22,22,1c,3c,42,42,3e,02,02
34 data 42,3c,30,78,fc,30,30,30,30,30,0c,   6390
   1e,3f,0c,0c,0c,0c,0c,30,30,30
35 data 30,30,30,30,30,0c,0c,0c,0c,0c,0c,   5345
   0c,0c,00,00,00,00,00,00,00,00
36 data 00,00,00,00,00,00,00,00,00,00,00,   4799
   00,00,00,00,00,00,00,00,00,00
37 data 00,00,00,08,14,08,1c,2a,08,14,14,   5683
   d2,7d,39,46,82,00,00,00,ff,9f
38 data 9f,ff,f1,f1,f1,ff,ff,f9,f9,ff,8f,   6351
   8f,8f,ff,ff,8f,8f,8f,ff,f9,f9
39 data ff,ff,f1,f1,f1,ff,9f,9f,ff,fe,fe,   6090
   c6,fe,fe,c6,c6,00,fc,fc,c0,c0
40 data c0,fc,fc,00,fe,c0,c0,fe,c0,c0,fe,   6819
   00,fc,fc,30,78,78,30,30,00,cc
41 data d8,f0,e0,f0,d8,cc,00,38,7c,c6,c6,   6340
   c6,7c,38,00,fc,cc,cc,fc,d8,cc
42 data c6,00,fe,c0,c0,fe,06,06,fe,00,fc,   5031
   fc,30,30,30,30,30,00,00,18,18
43 data 00,18,18,00,00,01,02,44,28,30,30,   5282
   48,00,00,00,00,00,00,00,00,00
44 data 00,00,00,00,00,00,00,01,01,03,06,   5909
   0c,1a,3c,68,f1,80,73,60,30,58
45 data 3c,16,8f,00,00,00,00,00,00,80,00,   4864
   00,00,00,00,00,00,00,00,00,00
46 data 00,00,00,00,01,07,03,07,0d,3e,7c,   5548
   f1,a6,cf,a7,cf,17,29,48,9c,08
47 data ff,e5,f3,e8,94,12,39,10,ff,c0,e0,   5887
   b0,7c,36,8f,65,f3,00,00,00,00
48 data 00,00,80,e0,8e,7b,80,64,1f,03,01,   5272
   00,b0,c0,00,00,c0,ff,00,c0,3e
49 data 1c,00,03,cf,fc,31,06,7c,38,00,c0,   5916
   f3,3f,8c,60,0d,03,00,00,03,ff
50 data 00,03,71,de,01,26,f8,c0,80,00,00,   5377
   00,00,80,60,d2,0c,12,3c,23,29
51 data 27,21,51,41,4b,38,c0,00,c0,3c,03,   6489
   01,ff,1c,03,00,03,3c,c0,80,ff
52 data 3c,c4,94,e4,84,a3,82,ca,12,0d,10,   6186
   20,da,06,09,00,10,08,48,65,97
53 data 09,10,10,61,5d,d5,5d,c1,45,51,41,   5992
   00,00,00,80,80,80,80,80,00,00
54 data 00,01,01,01,01,01,82,bb,ba,ba,82,   5511
   92,c6,82,00,04,c8,30,0c,02,00
55 data 00,08,1f,3f,df,09,60,fc,ff,e7,f9,   5912
   e0,c0,81,00,00,ff,80,80,1f,7f
56 data ff,ff,3f,0f,01,01,f8,f0,c0,f8,fe,   6156
   ff,e7,98,7c,3f,1f,3f,1f,80,00
57 data c0,38,84,e3,f0,f8,00,87,98,60,47,   6097
   58,60,8f,f0,e1,19,06,e2,1a,06
58 data f1,0f,00,18,24,43,33,33,33,30,87,   6889
   9d,66,47,5f,7c,bf,f2,e1,59,c6
59 data e2,fa,76,f9,8f,a7,9f,66,4f,5c,6c,   6208
   8f,f8,e9,d9,c6,e2,7a,66,f1,2f
60 data ff,7e,3c,18,66,c3,81,00,7f,fe,48,   6516
   26,7c,83,00,00,ee,bd,99,a5,c3
61 data 81,42,00,00,00,02,01,00,00,00,00,   5390
   1c,3f,ff,ff,ff,1f,04,04,00,c0
62 data f8,ff,ff,ff,ff,1f,00,73,3f,ff,ff,   6410
   ff,ff,f8,f8,fc,ff,ff,ff,f8,20
63 data 20,00,00,40,80,00,00,00,00,00,00,   5441
   00,81,7f,1f,07,02,08,08,3e,ff
64 data ff,ff,ff,e0,47,20,18,cf,f8,f3,e2,   6596
   43,e2,04,18,f3,1f,cf,47,c2,10
65 data 10,7c,ff,ff,ff,ff,07,00,00,00,81,   5736
   fe,f8,e0,40,02,02,06,04,04,0c
66 data 08,08,00,70,50,50,70,00,00,01,58,   5874
   5c,53,c1,80,80,80,80,1a,7a,8a
67 data 03,01,01,01,01,00,0e,0a,0a,0e,00,   5474
   00,00,40,40,60,20,20,30,10,10
68 data 08,18,70,40,40,78,4d,c2,01,01,01,   6026
   01,01,8d,df,21,80,70,0c,03,00
69 data 00,00,7f,01,0e,30,c0,00,00,00,fe,   5577
   80,80,80,80,81,b3,fe,80,18,08
70 data 0e,02,02,9e,f2,01,80,80,87,81,8d,   5747
   83,81,81,01,01,c3,02,22,c2,46
71 data 44,8f,1c,70,40,40,40,40,40,f1,38,   6026
   0e,02,02,02,02,02,80,80,c4,4a
72 data 41,40,60,23,01,01,11,21,41,81,91,   5705
   e1,87,8c,80,80,80,80,80,ff,c4
73 data 24,04,04,04,04,04,00,40,40,40,40,   5331
   40,40,40,c0,02,02,02,02,02,02
74 data 02,03,24,21,22,20,20,20,20,ff,81,   5901
   c1,81,41,01,01,01,ff,00,00,00
75 data 00,00,00,00,00,a2,00,8a,9d,00,3e,   5153
   e8,e0,00,d0,f8,ad,00,ff,29,0f
76 data c9,01,90,f7,9d,00,3f,e8,e0,00,d0,   5361
   ef,bd,00,3f,29,08,c9,08,d0,08
77 data bd,00,3f,29,f7,9d,00,3f,e8,e0,10,   6384
   d0,ea,a2,f0,bd,00,3f,29,02,c9
78 data 02,d0,08,bd,00,3f,29,fd,9d,00,3f,   5267
   e8,e0,00,d0,ea,bd,00,3f,29,01
79 data c9,01,d0,08,bd,00,3f,29,fe,9d,00,   5952
   3f,8a,18,69,10,aa,90,e8,a2,0f
80 data bd,00,3f,29,04,c9,04,d0,08,bd,00,   5716
   3f,29,fb,9d,00,3f,8a,18,69,10
81 data aa,90,e8,a2,00,bd,00,3f,c9,00,d0,   5574
   11,ad,00,ff,29,0f,c9,00,f0,f7
82 data 9d,00,3f,a2,00,4c,4c,15,e8,e0,00,   5916
   d0,e3,a2,00,bd,00,3f,29,08,c9
83 data 08,d0,0d,8a,38,e9,10,a8,b9,00,3f,   5919
   09,02,99,00,3f,bd,00,3f,29,02
84 data c9,02,d0,0d,8a,18,69,10,a8,b9,00,   5622
   3f,09,08,99,00,3f,bd,00,3f,29
85 data 04,c9,04,d0,0b,8a,a8,c8,b9,00,3f,   5854
   09,01,99,00,3f,bd,00,3f,29,01
86 data c9,01,d0,0b,8a,a8,88,b9,00,3f,09,   5962
   04,99,00,3f,e8,e0,00,d0,a7,ae
87 data 00,ff,86,d6,bd,00,3e,09,80,9d,00,   6124
   3e,a2,00,86,d2,a2,00,bd,00,3f
88 data 29,01,c9,01,d0,1b,8a,a8,88,20,4b,   6650
   16,4c,5d,16,b9,00,3e,29,80,c9
89 data 80,d0,08,bd,00,3e,09,80,9d,00,3e,   5189
   60,bd,00,3f,29,02,c9,02,d0,08
90 data 8a,18,69,10,a8,20,4b,16,bd,00,3f,   5309
   29,04,c9,04,d0,06,8a,a8,c8,20
91 data 4b,16,bd,00,3f,29,08,c9,08,d0,08,   6124
   8a,38,e9,10,a8,20,4b,16,e8,e0
92 data 00,d0,a6,e6,d2,a6,d2,e0,10,90,9c,   6230
   a2,00,bd,00,3e,29,80,c9,80,f0
93 data 03,4c,30,15,ad,00,ff,9d,00,3e,e8,   5611
   e0,00,d0,e9,a2,00,ac,00,ff,b9
94 data 00,3f,29,10,c9,10,f0,f4,b9,00,3f,   5376
   09,10,99,00,3f,e8,e0,28
```

Listing 1: Harte Kämpfe in „Shogun"

```
 0 rem      shogun lader 2
 1 rem
 2 rem      (c)   1987  by
 3 rem
 4 rem      olaf panz
 5 rem
 6 rem      drosselgasse  4
 7 rem
 8 rem      2095 marschacht
 9 rem
10 poke 56,56:clr :for t=14642 to 16382:   6496
   read a$:poke t,dec (a$):pr=int ((pr+t*
   dec (a$))/256)+t:next t
11 if pr<>23171 then print "fehler in      3093
    datas !"
12 data 90,e7,a2,00,ac,00,ff,b9,00,3f,29,  6613
   20,c9,20,f0,f4,b9,00,3f,09,20
13 data 99,00,3f,e8,e0,6e,90,e7,a2,00,ac,  6602
   00,ff,b9,00,3f,29,40,c9,40,f0
14 data f4,b9,00,3f,29,20,c9,20,f0,eb,b9,  6351
   00,3f,09,40,99,00,3f,e8,e0,56
15 data 90,de,a2,00,ac,00,ff,b9,00,3f,29,  6607
   20,c9,20,f0,f4,b9,00,3f,29,40
16 data c9,40,f0,eb,b9,00,3f,29,80,c9,80,  6177
   f0,e2,b9,00,3f,09,80,99,00,3f
17 data e8,e0,3c,90,d5,a2,00,bd,00,3e,09,  6064
   80,9d,00,3e,e8,e0,00,d0,f3,60
18 data a0,0c,20,8f,0d,28,30,0e,26,f8,0e,  5777
   26,5d,0c,96,00,00,00,a0,0c,1a
19 data 8f,0d,28,33,0e,23,a3,0c,86,b7,0c,  6083
   86,95,0d,84,b1,0d,84,33,0e,8a
20 data 4c,0e,8a,00,00,00,a0,0c,20,18,0d,  5523
   20,8f,0d,28,30,0e,15,4f,0e,07
21 data 0b,0f,0c,39,0c,83,a4,0c,83,b8,0c,  6034
   83,21,0d,83,33,0d,83,93,0d,84
22 data a4,0d,89,b1,0d,84,4e,0e,85,36,0e,  5800
   8a,3c,0e,8a,00,00,00,ae,0c,10
23 data 1a,0d,0d,8f,0d,28,3d,0e,0f,58,0e,  5593
```


WWW.HOMECOMPUTERWORLD.COM

Setz dich an die Spitze! RUN.

Hier war eine Abo-Bestellkarte – aber jemand war schneller als Sie und hat seine „RUN" mit rund 13% Preisvorteil bereits abonniert! Sie können diesen Vorteil auch nutzen: Schreiben Sie einfach an CW-Publikationen, Vertrieb RUN, Postfach 40 04 29, D-8000 München 40, Stichwort: „RUN-Abo".



Cool die Führung übernehmen. Mit RUN, dem unabhängigen Home-Computermagazin.
RUN geht nach vorn: anspruchsvolle Listings, Programmiertips und Tricks, Kaufempfehlungen auch für Hardware. Aktuelle Nachrichten weltweit. Z. B. USA und Europa im Vergleich.
Wertvolle Informationen – mit Schwerpunkt C64, 128 und Ami
RUN im Zeitschriftenhandel. Jed Monat neu und aktuell. Falls vergriffen, einfach anrufen oder den Verlag schreiben. Wir schicken postwendend ein Prob exemplar – kostenlos.
RUN. Denn ein Profi kennt sein Ziel.

CW PUBLIKATIONE

Ihre starke Verbindung zur Computerw

Postfach 40 04 29, D-8000 München 40

WWW.HOMECOMPUTERWORLD.COM

WWW.HOMECOMPUTERWORLD.COM

Das Bestell-Programm für Disketten Jockeys — RUN-o-thek Disketten-Service

Diskette mit allen Programm-Listings des aktuellen RUN:

Nur DM 21,80 (inkl. MwSt.) plus DM 3,— Versandkosten = DM 24,80.

In der RUN-o-thek gibt's jeden Monat die Diskette mit allen VC-20- oder C 64-Programmen, die in der aktuellen RUN als Listings veröffentlicht wurden.
Sie brauchen nur noch ein Floppy-Laufwerk und einen Commodore VC-20 oder C 64, um mit den Programmen aus der RUN arbeiten zu können — ohne Listings abtippen zu müssen und ohne Tippfehler!
Disketten-Jockeys bestellen ihre Diskette mit dieser Bestellkarte — einfach ausfüllen, abtrennen und im ausreichend frankierten Umschlag einsenden an:

CW-Publikationen Verlagsgesellschaft mbH — RUN-o-thek —
Postfach 40 04 29, D-8000 München 40



Zu der in RUN Heft ..., S. ..., erschienenen Anzeige gebe ich folgende Bestellung auf:

Menge	Produkt	ges. DM

Bitte ich um Information über das Produkt:

Bitte schicken Sie mir ☐ Preisliste ☐ Katalog/Prospekt

Datum/Unterschrift

Bitte diesen Coupon direkt an die jeweilige Firma zuschicken.

Ich nutze die Gelegenheit! Meine Anzeige im RUNboard

Ich packe die Gelegenheit beim Schopf: Hier ist meine Gelegenheits-Anzeige zum Taschengeld-Tarif von nur DM 5,—* für sechs Zeilen (jeder Buchstabe, jedes Satzzeichen, jeder Wortzwischenraum = 1 Kästchen). Adresse nicht vergessen!

Das ist meine Anzeige fürs RUNboard:

RUBRIK UND BEREICH (bitte je 1 Kreuzchen) ○ Verkaufe ○ Suche ○ Tausche ○ Hardware ○ Peripherie ○ Sonstiges ○ Software ○ Disketten

*Endpreis inkl. ges. Mehrwertsteuer
Achtung: Sie erhalten keine Rechnung. Der Betrag von DM 5,— muß Ihrer Bestellung beiliegen, sonst erfolgt keine Veröffentlichung der Anzeige.
Im RUNboard werden nur private Anzeigen veröffentlicht. Chiffre-Anzeigen sind nicht möglich!

Die DM 5,— liegen bei als: ○ V-Scheck ○ Geldschein

RUN Abo-Karte

Ich bestelle RUN ab Ausgabe . . . . . mit rund **13% Preisvorteil,** für mindestens ein Jahr (12 Ausgaben) **frei Haus.**

Dieses Abonnement verlängert sich automatisch um ein weiteres Jahr, wenn es nicht acht Wochen vor Ablauf der Bezugszeit gekündigt wird.

Zustellung, Vertriebskosten und MwSt. sind im günstigen Preis von DM 68,— bereits enthalten. Dieses Angebot gilt nur für die Bundesrepublik Deutschland und West-Berlin. Auslandsendpreis: DM 78,—, für die Schweiz sfr. 74,—. Luftpostversand auf Anfrage.

☐ **Geschäft** ☐ **Privat**

Vorname, Name

Geb.-Jahr

Firma (nur wenn Lieferanschrift)

Straße, Nr./Postfach

PLZ/Ort

Datum/Unterschrift
(bei Minderjährigen: Unterschrift des ges. Vertreters)

Gewünschte Zahlungsweise

☐ **Bequem und bargeldlos** durch Bankabbuchung

Bankleitzahl (bitte vom Scheck abschreiben)

Konto-Nr./Inhaber

Geldinstitut/Ort

☐ **Gegen Rechnung,** zahlbar sofort nach Erhalt.
(Bitte keine Vorauszahlung leisten — Rechnung abwarten).

GARANTIE
Ich weiß, daß ich diese Vereinbarung innerhalb einer Woche bei der Bestelladresse schriftlich widerrufen kann, wobei bereits die rechtzeitige Absendung meines Widerrufschreibens zur Fristwahrung ausreicht.
Ich bestätige das durch meine zweite Unterschrift.

Datum Unterschrift

Meine Adresse:

Vorname, Name

Straße, Nr.

PLZ/Ort

Telefon (für evtl. Rückfragen)

Unterschrift

(bei Minderjährigen: Unterschrift des gesetzlichen Vertreters)

Achtung:

Unbedingt hier unterschreiben! Nicht unterschriebene Anzeigen-Bestellungen können nicht ausgeführt werden.

RUN-o-thek-Bestellkarte

Ich bestelle die folgenden Disketten aus der RUN-o-thek:

ANZAHL DISKETTEN	RUN/AUSGABE	JAHRGANG

Preis pro Diskette: DM 21,80 (inkl. MwSt.) plus DM 3,— Versandkosten = DM 24,80
Anzahl der bestellten Disketten: ____ Stück. Gesamtpreis der bestellten Disketten: DM _____

- ○ Ein Verrechnungs-Scheck über den Gesamt-Betrag von DM _____ liegt bei.
- ○ Ich bezahle die bestellten Disketten per Nachnahme zum Preis von DM 24,80 pro Diskette (inkl. MwSt. und Versandkosten) plus NN-Gebühr.

Meine Anschrift:

Vorname, Name

Straße, Nr.

PLZ/Ort

Datum Unterschrift (bei Minderjährigen Unterschrift des Erziehungsberechtigten)

RUN UNABHÄNGIGES COMMODORE-COMPUTERMAGAZIN

Lesen Sie RUN jetzt regelmäßig mit rund 13% Preisvorteil

Bitte ausreichend frankieren

Antwortkarte

CW-Publikationen
Verlagsgesellschaft mbH
— Vertrieb RUN —
Postfach 40 04 29

D-8000 München 40

Absender (Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufname

Bitte diesen Coupon direkt an die jeweilige Firma zuschicken.

Bitte ausreichend frankieren

Postkarte

Firma

Straße/Postfach

PLZ Ort

WWW.HOMECOMPUTERWORLD.COM

Impressum

Chefredakteur: Manfred S. Schmidt (mss) (verantwortlich, Anschrift siehe Redaktion)
Redaktion: Christian Rogge (rg), Werner Breuer (wb), Michael Nickles (mn), Manfred Wenzel (mw)
Redaktionelle Mitarbeiter:
Christoph Grunwald (cgr), Siegfried Schwarze (sis), Andreas Vichr (avi), Peter Vogel (pv), Claus Peter Lippert (cpl)
Redaktionsassistentin: Siggi Pesch (sp)
Gestaltung: Karin Wirth
Run-o-thek: Wolfgang Kurtz (wok)
Anschrift der Redaktion: RUN, Postfach 40 04 29, Rheinstraße 26/28, 8000 München 40, Telefon: 0 89/3 60 86-0, Telex: 5 215 350 comw d, Telekopierer: 0 89/3 60 86-1 09
Auslandsredaktionen:
Österreich: Erich K. Surböck, c/o ADV, Trattnerhof 2, A-1010 Wien, Tel.: 00 43/222/52 32 71
Schweiz: Günter Schilling, Karl-Jaspers-Allee 4, CH-4052 Basel, Tel.: 00 41/61/42 47 16
USA: CW-COMMUNICATIONS INC., 375 Cochituate Road, Box 880, USA-Framingham, Mass. 01701, Tel.: 001/617/879 07 00, Tx.: 00230/951 153 computwrld fmh
Japan: COMPUTERWORLD/Japan, 1- 19-7, Shintomi Chuoku, J-Tokyo 104, Tel.: 00 81/3/5 51 38 82
China: China COMPUTERWORLD, 74 LuGuGun Road, P.O. Box 750, RC-Beijing 100039, Tel.: 00 88/6/814-61 74
Verlagsrepräsentanten:
Großbritannien: Beere Hobson Ass., Euan C. Rose, 34 Warwick Road, Kenilworth, GB-Warwickshire CV8 1HE, Tel.: 00 44/926/51 24 24.
Comecon: Klaus J. Ruppert, Goethestr. 10, 6000 Frankfurt/Main 1, Tel.: 06 11/28 26 90
USA: CW International Marketing Services, Diana La Muraglia, 375 Cochituate Road, Box 880, USA-Framingham, Mass. 01701, Tel.: 001/617/879 07 00, Tx.: 00230/951 153 computwrld fmh
Taiwan: Alice Chen ACE Media, Agency Co., Ltd., P.O.Box 26-578, Taipei R.O.C., Tel. 021721-4302, Telex 14142 acegroup
Japan: Steven Yamada, Tokyo Representative Corp., Sanshin Kogyo Jimbocho 3F, Chiyoda-ku, Tokyo 101 Japan, Tel.: 230-4117, Tlx.: (781) J26860
© Copyright CW-Publikationen Verlags GmbH
Anzeigenpreise: Für Produktanzeigen fordern Sie bitte unsere Mediaunterlagen an.
Vertrieb Handelsauflage: MZV Moderner Zeitschriften Vertrieb GmbH, Breslauer Straße 5, 8057 Eching, Tel.: 089/31 90 06-0, Telex: 522 656.
Telefon-Durchwahl: 089/3 60 86 ( )
Vertriebsleitung: Brigitte Schleibinger (-153/-155)
Leserservice: Gustav Drechsler (-218)
Anzeigenleitung: Sylvia Stier (-161); (verantwortlich für Anzeigen, Anschrift siehe unter Anzeigen)
Anzeigenverkaufsleitung:
Barbara Schönberger (-130)
Anzeigenverkauf: Marianne Gad (-201)
Anzeigendisposition: Ursel Sauter (-126)
Anschrift für Anzeigen und Vertrieb: RUN, Postfach 400 429, Rheinstraße 26/28, 8000 München 40, Telefon: 0 89/3 60 86-0, Telex: 5 215 350 comw d, Telekopierer 089/3 60 86-1 09
Objektdisposition:
Rainer Oberländer (Leitung), Stefan Liba
Druck und Beilagen:
Carl Gerber Grafische Betriebe, Muthmannstraße 4, 8000 München 45, Tel.: 089/3 23 93-233 (Anschrift für Beilagen)
Zahlungsmöglichkeiten:
Bayerische Vereinsbank, BLZ 700 202 70, Konto-Nr. 116 000, Pschk. München 97 40-800
Erfüllungsort, Gerichtsstand: München
Verlag: CW-Publikationen Verlagsgesellschaft mbH, Rheinstraße 26/28, 8000 München 40, Telefon: 089/3 60 86- 0, Telex: 5 215 350 comw d, Telekopierer: 089/3 60 86-109
Produktionsleitung: Heinz Zimmermann
Redaktionsdirektor: Dieter Eckbauer
Geschäftsführer: Eckhard Utpadel, Walter Boyd, Patrick McGovern

CW PUBLIKATIONEN

RUN ist ein Mitglied der CW-Communications/Inc.-Gruppe, der Welt größter Verleger für computerbezogene Informationen. Die Gruppe veröffentlicht über 50 Computer-Publikationen in mehr als 20 Ländern. Neun Millionen Menschen lesen eine oder mehrere Publikationen dieser Gruppe pro Monat. Die Mitglieder der CWCI-Gruppe sind am CW Communications International News Service angeschlossen, einem täglichen Nachrichtendienst für die aktuellsten Meldungen aus dem internationalen DV-Geschehen.
Mitglieder dieser Verlagsgruppe sind:
Argentinien: Computerworld/Argentina; Asien: Asian Computerworld; Australien: Computerworld Australia, Australian PC World, Macworld; Brasilien: DataNews, PC Mundo; China: China Computerworld, China Computerworld Monthly; Dänemark: Computerworld/Danmark, PC World, Run (Commodore); Finnland: Mikro; Frankreich: Le Monde Informatique, Golden (Apple), OPC (IBM), Theoreme, Distributique; Deutschland: Computerwoche, Infowelt, PC Welt, Computer Business, RUN; Italien: Computerworld Italia, PC Magazine; Japan: Computerworld Japan; Mexiko: Computerworld/Mexico; Niederlande: Computerworld Netherlands, PC World; Norwegen: Computerworld Norge, PC Mikrodata; Spanien: Computerworld Espana, PC World, Commodore World; Schweden: Computer-Sweden, Mikrodatorn, Svenska PC World; Schweiz: Computerworld Schweiz; Großbritannien: Computer News, PC Business World, Computer Business; Ungarn: Computerworld Informatica; Venezuela: Computerworld Venezuela; USA: Amiga World, Computerworld, inCider, Infoworld, MacWorld, Micro Marketworld, PC World, Run, 73 Magazine, 80 Micro, Focus Publications, Network World.

EINE PUBLIKATION DER CW COMMUNICATIONS

EIDP
Deutschland Report

Newsletter der IDC-Deutschland

- aktuelle Daten der Computerindustrie
- Markttrends
- Analysen
- Neuankündigungen

EIN MUSS
für das Management der Computerindustrie

Erscheinungsweise: 14tägig
Preise:
Einzelausgabe DM 50,—
Jahresabonnement DM 680,—

Anfragen und Bestellungen richten Sie bitte an:

IDC
IDC DEUTSCHLAND GMBH
Stuttgarter Str. 10,
6236 Eschborn,
Telefon: 06196/4 86 88

RUNboard-

Gebrauchs-Anweisung

1. Anzeigentext in die vorgedruckten Zeilen der Karte schreiben — maximal 6 Zeilen mit Anschrift oder Telefon-Nummer (pro Buchstabe, Satzzeichen, Wortzwischenraum = 1 Kästchen).
2. Rubrik, Bereich und Zahlungsweise ankreuzen, Adresse auf der Rückseite der Karte angeben.
3. Datum und Unterschrift nicht vergessen
4. Karte zusammen mit 5-DM-Schein oder V-Scheck über DM 5,— in Umschlag stecken.
5. Umschlag zukleben, ausreichend frankieren und abschicken an:
   CW-Publikationen
   — RUNboard —
   Postfach 40 04 29
   8000 München 40
6. Die Anzeige erscheint im nächstmöglichen RUNboard.

Ich habe die Gelegenheit genutzt!

MEMO

Ich habe am _____ eine Gelegenheits-Anzeige im RUNboard bestellt. Sechs Zeilen zum Taschengeld-Tarif von nur DM 5,— inkl. ges. Mehrwertsteuer. Die DM 5,— habe ich der Bestellung beigelegt als

○ V-Scheck ○ Geldschein

RUN-o-thek Disketten-Service

MEMO

Ich habe am _____ bei der RUN-o-thek _____ Diskette(n) mit den Programmen aus RUN-Ausgabe _____ bestellt.
Preis pro Diskette: DM 21,80 plus DM 3,— Versandkosten = DM 24,80

○ Bezahlt mit V-Scheck-Nr. _____
Konto-Nummer _____
BLZ/Geldinstitut _____

○ Ich bezahle per Nachnahme.

WWW.HOMECOMPUTERWORLD.COM

```
   07,7b,0e,03,05,0f,0f,63,0c,82
24 data bd,0c,94,19,0d,83,28,0d,83,96,0d,  6205
   85,9c,0d,8e,5b,0e,89,7a,0e,89
25 data 00,00,00,c8,0c,12,8f,0d,28,58,0e,  5707
   26,f8,0e,0b,62,0c,88,93,0d,85
26 data a5,0d,85,64,0e,89,70,0e,89,7a,0e,  5675
   89,00,00,00,40,0d,13,a3,0d,14
27 data 30,0e,11,52,0e,04,f9,0e,21,62,0c,  6164
   91,44,0d,86,51,0e,8a,fb,0e,85
28 data 00,00,00,a0,0c,20,68,0d,1a,aa,0d,  5226
   02,ad,0d,83,24,0e,04,28,0e,02
29 data 52,0e,04,ae,0d,08,8e,0e,0f,30,0e,  5379
   02,5a,0e,02,5c,0e,02,86,0e,02
30 data b0,0e,02,da,0e,02,04,0f,02,2e,0f,  5885
   02,58,0f,02,ae,0c,85,79,0d,87
31 data 8d,0e,88,00,00,00,ab,0c,0d,9b,0d,  5416
   1c,30,0e,13,49,0e,0e,33,0f,13
32 data aa,0c,94,b9,0c,86,ad,0d,84,44,0e,  6110
   86,53,0e,86,3b,0f,84,00,00,00
33 data ff,a9,08,78,20,d2,ff,a9,1e,20,d2,  5763
   ff,ae,00,ff,8e,15,ff,8e,19,ff
34 data a2,ff,86,89,a2,57,8e,15,ff,8e,19,  6277
   ff,a9,93,20,d2,ff,a2,4b,86,d2
35 data a2,17,86,d3,a6,d6,bd,00,3e,6a,29,  5507
   07,aa,e0,00,f0,17,a0,01,b1,d2
36 data c9,00,d0,01,ca,a5,d2,18,69,03,90,  5812
   02,e6,d3,85,d2,4c,67,21,a0,00
37 data b1,d2,85,d4,c8,b1,d2,c9,00,f0,70,  5470
   85,d5,c8,b1,d2,aa,29,80,c9,80
38 data d0,22,8a,29,7f,aa,a0,00,a9,77,91,  6636
   d4,a9,78,c8,91,d4,a5,d4,18,69
39 data 28,90,02,e6,d5,85,d4,ca,e0,00,d0,  6246
   e5,4c,1e,19,8a,a8,a9,43,91,d4
40 data 88,c0,00,d0,f7,a5,d2,18,69,03,90,  6061
   02,e6,d3,85,d2,4c,d8,18,4f,49
41 data 4d,4e,4a,51,20,20,20,20,20,20,20,  5404
   2e,20,51,20,20,20,20,4c,4e,48
42 data 4b,50,51,20,42,51,20,20,20,20,42,  5773
   51,20,20,20,20,20,20,a2,00,bd
43 data 2c,19,9d,00,0c,e8,e0,28,90,f5,a9,  6196
   0c,8d,1b,08,a9,77,8d,21,08,ad
44 data 11,ff,29,80,09,0f,8d,11,ff,ad,12,  5899
   ff,29,fb,8d,12,ff,a2,00,a9,2f
45 data 9d,28,0c,e8,e0,28,d0,f8,a2,0c,86,  5389
   d3,a2,4f,86,d2,a0,00,a9,2f,91
46 data d2,c8,91,d2,a5,d2,18,69,28,90,02,  6386
   e6,d3,85,d2,a6,d3,e0,10,90,e6
47 data a2,00,a9,2f,9d,c0,0f,e8,e0,28,d0,  5345
   f8,a6,d6,bd,00,3f,aa,29,01,c9
48 data 01,d0,0c,a0,00,a9,3e,99,38,0c,c8,  5734
   c0,08,90,f8,8a,29,02,c9,02,d0
49 data 10,a0,00,a9,3f,99,8f,0d,98,18,69,  5729
   28,a8,c9,b0,90,f2,8a,29,04,c9
50 data 04,d0,0c,a9,40,a0,00,99,d0,0f,c8,  5917
   c0,08,d0,f8,8a,29,08,c9,08,d0
51 data 10,a0,00,a9,41,99,68,0d,98,18,69,  5657
   28,a8,c9,b0,90,f2,a2,00,a9,53
52 data a0,00,9d,71,0c,e8,c8,18,69,01,c0,  5809
   06,90,f4,85,d2,8a,18,69,22,aa
53 data a5,d2,e0,f0,90,e5,4c,91,1a,a0,00,  5696
   91,d2,c8,91,d2,a0,28,91,d2,c8
54 data 91,d2,60,a0,00,91,d2,c8,18,69,01,  5792
   91,d2,18,69,01,a0,28,91,d2,18
55 data 69,01,c8,91,d2,60,a2,13,86,d2,a2,  6372
   0d,86,d3,a9,20,20,33,1a,a6,d6
56 data bd,00,3f,aa,29,80,c9,80,d0,06,a9,  5140
   2e,8d,3b,0d,60,8a,29,40,c9,40
57 data d0,05,a9,2d,4c,75,1a,8a,29,20,c9,  6692
   20,d0,ec,a9,79,4c,75,1a,a6,d6
58 data bd,00,3e,29,80,c9,80,f0,03,20,5a,  5307
   1a,a6,d6,bd,00,3f,29,10,c9,10
59 data d0,05,a9,44,8d,99,0f,78,a2,be,8e,  6686
   14,03,a2,1a,8e,15,03,58,4c,f2
60 data 1a,a5,d8,c9,08,b0,08,18,69,01,85,  5529
   d8,4c,0e,ce,a2,00,bd,20,12,a8
61 data bd,28,12,9d,20,12,bd,30,12,9d,28,  5763
   12,bd,38,12,9d,30,12,98,9d,38
62 data 12,e8,e0,08,90,e1,a9,00,4c,c7,1a,  6293
   a6,d6,bd,00,3e,29,30,c9,00,d0
63 data 04,a2,2c,a0,1b,c9,10,d0,04,a2,00,  5097
   a0,14,c9,20,d0,04,a2,f8,a0,13
64 data c9,30,d0,04,a2,f0,a0,13,86,d2,84,  5376
   d3,a0,00,b1,d2,99,18,12,c8,c0
65 data 08,90,f6,4c,65,1b,d2,7d,39,46,82,  6083
   00,00,00,a2,98,86,d2,a2,12,86
66 data d3,a2,08,86,d4,a2,14,86,d5,b1,d2,  5334
   aa,b1,d4,91,d2,8a,91,d4,c8,c0
67 data 00,d0,f1,a6,d2,e0,98,d0,0b,a2,b8,  5960
   86,d2,a2,28,86,d4,4c,89,1b,60
68 data a6,d6,bd,00,3e,29,01,c9,01,d0,0f,  6166
   ad,a0,12,c9,1c,f0,05,a0,00,20
69 data 34,1b,4c,86,1b,ad,a0,12,c9,1c,f0,  5791
   f1,4c,8e,1b,a0,df,4c,44,1b,a6
70 data d6,bd,00,3e,29,40,c9,40,d0,13,a2,  6014
   2b,86,d2,a2,66,86,d3,a2,04,86
71 data d4,a2,0f,86,d5,4c,bc,1b,a2,12,86,  5919
   d2,a2,30,86,d3,a2,0d,86,d4,a2
72 data 28,86,d5,a2,00,86,db,a2,0c,86,dc,  5655
   a0,00,b1,db,c9,77,d0,16,a6,d2
73 data a5,dc,38,e9,04,85,dc,8a,91,db,18,  5946
   a5,dc,69,04,85,dc,4c,05,1c,c9
74 data 78,f0,e6,c9,53,90,09,c9,77,b0,05,  6681
   a6,d3,4c,ce,1b,c9,2f,d0,05,a6
75 data d4,4c,ce,1b,c9,43,d0,05,a6,d5,4c,  5543
   ce,1b,e6,db,a6,db,e0,00,d0,b7
76 data e6,dc,a6,dc,e0,10,90,af,a2,0c,86,  5504
   dc,a2,52,86,db,a2,63,86,de,a5
77 data d9,85,d2,a5,da,85,d3,ad,06,ff,09,  6087
   10,8d,06,ff,20,04,1f,20,70,24
78 data 20,71,25,a5,d9,85,d2,a5,da,85,d3,  6010
   20,04,1f,4c,71,25,a9,3a,a2,0c
79 data e2,e8,e0,03,90,f9,a5,d9,85,d2,a5,  5421
   da,85,d3,ad,06,ff,09,10,8d,06
80 data ff,4c,04,1f,a5,dc,85,d3,a9,00,a2,  5834
   09,20,69,21,60,ea,ea,20,42,1a
81 data a5,d3,38,e9,04,85,d3,8a,20,33,1a,  5855
   18,a5,d3,69,04,85,d3,60,a0,00
82 data 84,d4,84,d5,20,a1,1c,a0,01,20,a1,  5304
   1c,a0,28,20,a1,1c,a0,29,4c,a1
83 data 1c,b1,d2,c9,20,b0,07,a5,d4,09,40,  5920
   85,d4,60,c9,7a,90,04,c9,7e,90
84 data f1,c9,3f,d0,07,a5,d4,09,02,85,d4,  6135
   60,c9,3e,d0,07,a5,d4,09,01,85
85 data d4,60,c9,40,d0,07,a5,d4,09,04,85,  5830
   d4,60,c9,41,d0,07,a5,d4,09,08
86 data 85,d4,60,c9,2f,d0,07,a5,d4,09,30,  6468
   85,d4,60,c9,3a,90,0b,c9,3e,b0
87 data 07,a5,d4,09,80,85,d4,60,c9,79,d0,  6183
   07,a5,d5,09,01,85,d5,60,c9,2e
88 data d0,07,a5,d5,09,02,85,d5,60,c9,2d,  6202
   d0,07,a5,d5,09,04,85,d5,60,c9
89 data 43,d0,0d,a5,d5,09,08,85,d5,a5,d4,  6132
   09,10,85,d4,60,c9,44,d0,07,a5
90 data d5,09,10,85,d5,60,c9,77,d0,07,a5,  6600
   d5,09,20,85,d5,60,c9,78,f0,f5
91 data 4c,5f,1d,ad,00,ff,85,d7,a9,0e,85,  5874
   da,a9,04,85,d9,a9,ff,85,dd,4c
92 data 6d,1d,c9,21,90,07,c9,2d,b0,03,4c,  6103
   a7,1c,4c,3e,21,a9,00,85,e6,4c
93 data e2,23,20,89,1c,a5,d4,aa,29,01,c9,  6364
   01,d0,0f,c6,d6,a2,83,a0,0f,86
94 data d9,84,da,20,14,2b,38,60,8a,29,02,  5206
   c9,02,d0,0e,a5,d6,18,69
```

Listing 2: Boxen, Treten, Springen: Viel Action ist notwendig, um das Schwert zu finden

```
 0 rem      shogun maker 3
 1 rem
 2 rem     (c)   1987  by
 3 rem
 4 rem     olaf panz
 5 rem
 6 rem     drosselgasse  4
 7 rem
 8 rem     2095 marschacht
 9 rem
10 poke 56,56:clr :for t=14642 to 16382:   6496
   read a$:poke t,dec (a$):pr=int ((pr+t*
   dec (a$))/256)+t:next t
11 if pr<>30053 then print "Fehler in      2807
    datas !"
12 data 10,85,d6,a2,e2,a0,0d,4c,86,1d,8a,  6312
   29,04,c9,04,d0,09,e6,d6,a2,62
13 data a0,0c,4c,86,1d,8a,29,08,c9,08,d0,  5788
```


WWW.HOMECOMPUTERWORLD.COM

```
    0e,38,a5,d6,e9,10,85,d6,a0,0e
14 data a2,04,4c,86,1d,8a,29,10,c9,10,f0,  5936
    0b,8a,29,40,c9,40,d0,06,a9,50
15 data 85,d4,38,60,8a,29,80,c9,80,d0,2b,  6157
    a5,e1,09,01,85,e1,ad,11,ff,09
16 data 15,8d,11,ff,ad,12,ff,29,fd,09,01,  5305
    8d,12,ff,ad,00,ff,8d,0e,ff,a5
17 data dd,38,e9,50,b0,02,a9,00,85,dd,38,  5758
    60,a5,d5,aa,29,01,c9,01,d0,22
18 data ad,11,ff,09,17,8d,11,ff,ad,12,ff,  6184
    09,03,8d,12,ff,a9,0d,8d,0e,ff
19 data a5,e6,18,69,42,90,02,e6,e7,85,e6,  6891
    18,60,8a,29,02,c9,02,d0,30,ad
20 data 11,ff,09,37,8d,11,ff,ad,12,ff,09,  5941
    03,8d,12,ff,a2,00,8e,0e,ff,ca
21 data 8e,0f,ff,ad,10,ff,29,fc,09,02,8d,  6237
    10,ff,18,a5,e6,69,42,90,02,e6
22 data e7,4c,19,26,ea,8a,29,04,c9,04,d0,  6145
    1a,ad,11,ff,09,14,8d,11,ff,ad
23 data 12,ff,29,fc,09,01,a9,00,8d,0e,ff,  5535
    a2,ff,86,dd,18,60,8a,29,10,c9
24 data 10,d0,f7,ae,00,ff,bd,00,3f,29,10,  5042
    c9,10,d0,f4,86,d6,a2,0f,86,da
25 data a2,49,86,d9,a9,57,8d,11,ff,a9,03,  5226
    8d,10,ff,a9,d0,8d,0f,ff,ad,12
26 data ff,09,03,8d,12,ff,a9,00,a0,00,8c,  5351
    0e,ff,a2,00,e8,e0,40,90,fb,c8
27 data ee,19,ff,ee,15,ff,c0,80,90,eb,8c,  5041
    0e,ff,a2,00,e8,e0,40,90,fb,88
28 data c0,00,d0,f1,18,69,01,c9,08,d0,d3,  5780
    a9,00,8d,11,ff,a9,93,20,d2,ff
29 data 20,14,2b,38,60,a0,00,b1,d2,8d,10,  6598
    06,c8,b1,d2,8d,11,06,a0,28,b1
30 data d2,8d,12,06,c8,b1,d2,8d,13,06,60,  5763
    bd,10,06,91,d2,e8,60,a2,00,a0
31 data 00,20,1f,1f,c8,20,1f,1f,a0,28,20,  5801
    1f,1f,c8,20,1f,1f,a5,d3,38,e9
32 data 04,85,d3,a6,d6,bd,00,3e,a2,2b,29,  6319
    40,c9,40,f0,02,a2,12,8a,20,33
33 data 1a,a5,d3,18,69,04,85,d3,60,c9,77,  6405
    d0,01,60,c9,78,d0,fb,a9,77,60
34 data 85,2b,ad,10,06,20,5c,1f,c9,77,d0,  6840
    03,4c,14,20,20,04,1f,ad,12,06
35 data 20,5c,1f,c9,77,f0,21,ad,13,06,20,  6092
    5c,1f,f0,19,a5,2b,20,2a,2a,38
36 data a5,d3,e9,04,85,d3,a9,0c,20,33,1a,  6204
    a5,d3,18,69,04,85,d3,60,ad,12
37 data 06,20,5c,1f,c9,77,f0,0a,e6,d2,a6,  5957
    d2,e0,00,d0,02,e6,d3,ad,13,06
38 data 20,5c,1f,c9,77,f0,0a,c6,d2,a6,d2,  6641
    e0,ff,d0,02,c6,d3,20,38,23,90
39 data 0e,a5,d9,85,d2,a5,da,85,d3,20,04,  6342
    1f,4c,68,1f,20,04,1f,ad,10,06
40 data 20,5c,1f,c9,77,f0,0e,a0,00,a9,00,  6436
    91,d2,c8,a9,01,91,d2,4c,06,20
41 data a0,00,a9,7a,91,d2,c8,a9,7b,91,d2,  5738
    a0,28,a9,7c,91,d2,c8,a9,7d,91
42 data d2,4c,91,1f,20,04,1f,ad,12,06,20,  6130
    5c,1f,c9,77,f0,0a,c6,d2,a6,d2
43 data e0,ff,d0,02,c6,d3,ad,13,06,20,5c,  6262
    1f,c9,77,f0,0a,e6,d2,a6,d2,e0
44 data 00,d0,02,e6,d3,20,38,23,b0,8e,20,  6440
    04,1f,a2,00,bd,10,06,20,5c,1f
45 data c9,77,f0,08,e8,e0,03,d0,f1,4c,8c,  6412
    1f,4c,e3,1f,a2,00,8e,11,ff,a0
46 data 50,b1,d9,c9,20,d0,4b,c8,b1,d9,c9,  5822
    20,d0,44,a2,17,20,3a,29,a5,d9
47 data 8d,0e,ff,a5,da,38,e9,0c,0d,12,ff,  5625
    8d,12,ff,e6,e8,a6,e8,e0,04,90
48 data 08,a6,dd,e0,00,f0,02,c6,dd,a5,d9,  5739
    85,d2,a5,da,85,d3,20,26,1f,a5
49 data d2,18,69,28,90,04,e6,d3,e6,da,85,  5538
    d2,85,d9,a9,0c,4c,68,1f,a5,d9
50 data 85,d2,a5,da,85,d3,4c,f1,23,a9,00,  5624
    8d,08,ff,20,af,25,aa,a0,00,84
51 data 2b,29,80,c9,80,f0,03,4c,72,21,8a,  5538
    29,01,c9,01,f0,0b,a5,d2,38,e9
52 data 28,b0,02,c6,d3,85,d2,8a,29,02,c9,  5953
    02,f0,0b,a5,d2,18,69,28,90,02
53 data e6,d3,85,d2,8a,29,04,c9,04,f0,0e,  5911
    c6,d2,a4,d2,c0,ff,d0,02,c6,d3
54 data a0,08,84,2b,8a,29,08,c9,08,f0,0e,  5993
    e6,d2,a4,d2,c0,00,d0,02,e6,d3
55 data a0,04,84,2b,20,38,23,90,08,a5,d9,  5846
    85,d2,a5,da,85,d3,20,6a,1f,a5
56 data d2,85,d9,a5,d3,85,da,60,c9,53,90,  6064
    0a,c9,77,b0,06,a5,d4,09,10,85
57 data d4,60,20,04,1f,4c,17,20,10,06,20,  6460
    5c,1f,c9,77,f0,08,e8,e0,03,d0
58 data f1,4c,8c,1f,4c,17,20,ad,06,ff,29,  5751
    ef,8d,06,ff,4c,bd,18,8a,29,0f
59 data c9,0e,d0,62,a5,d2,38,e9,50,b0,02,  6165
    c6,d3,85,d2,4c,25,21,c9,20,90
60 data 08,c9,2c,b0,06,c9,21,90,02,38,60,  5871
    c9,7f,b0,04,c9,7a,b0,f6,18,60
61 data a0,03,b1,d2,20,87,21,b0,09,a0,2a,  5903
    b1,d2,20,87,21,90,ec,a9,4f,8d
62 data 11,ff,a9,02,8d,10,ff,ad,00,ff,8d,  6520
    09,ff,38,60,a0,27,b1,d2,20,87
63 data 21,b0,e5,c6,d3,a0,ff,b1,d2,e6,d3,  6021
    20,87,21,b0,d8,18,60,86,2c,8a
64 data 29,0f,c9,07,d0,21,e6,d2,a6,d2,e0,  6498
    00,d0,02,e6,d3,20,9f,21,90,0b
65 data a5,de,38,e9,03,b0,02,a9,00,85,de,  6191
    a0,10,84,2b,4c,25,21,a5,2c,29
66 data 0f,c9,0b,d0,2b,c6,d2,a6,d2,e0,ff,  5253
    d0,02,c6,d3,20,c3,21,90,0b,a5
67 data de,38,e9,03,b0,02,a9,00,85,de,a0,  6159
    14,4c,00,22,a5,de,38,e9,05,b0
68 data 02,a9,00,85,de,60,a5,2c,29,0f,c9,  6399
    05,d0,38,e6,d2,a6,d2,e0,00,d0
69 data 02,e6,d3,20,2b,23,b0,08,20,9f,21,  6014
    90,06,20,2c,22,4c,71,22,e6,d2
70 data a6,d2,e0,00,d0,02,e6,d3,20,2b,23,  5965
    b0,08,20,9f,21,90,03,20,2c,22
71 data a0,21,84,2b,4c,25,21,a5,2c,29,0f,  6003
    c9,09,d0,55,c6,d2,a6,d2,e0,ff
72 data d0,02,c6,d3,20,2b,23,b0,08,20,c3,  6174
    21,90,06,20,2c,22,4c,b1,22,c6
73 data d2,a6,d2,e0,ff,d0,02,c6,d3,20,2b,  6135
    23,b0,08,20,c3,21,90,03,4c,94
74 data 22,a9,25,85,2b,4c,25,21,e6,d2,a6,  6156
    d2,e0,00,d0,02,e6,d3,20,2b,23
75 data 90,01,60,20,9f,21,90,06,20,2c,22,  6060
    20,2c,22,18,60,a5,2c,29,0f,c9
76 data 06,d0,33,a9,18,85,2b,20,b8,22,b0,  6058
    08,20,b8,22,b0,03,20,b8,22,4c
77 data 25,21,c6,d2,a6,d2,e0,ff,d0,02,c6,  5517
    d3,20,2b,23,90,01,60,20,c3,21
78 data 90,06,20,2c,22,20,2c,22,18,60,a5,  6030
    2c,29,0f,c9,0a,d0,11,a9,1c,85
79 data 2b,20,f1,22,b0,cb,20,f1,22,b0,c6,  6166
    20,f1,22,4c,25,21,60,20,25,21
80 data 20,26,1f,a5,e1,09,02,85,e1,60,6c,  5633
    3f,00,a2,00,86,e8,20,26,1f,a0
81 data 50,a9,00,85,d4,20,a1,1c,a5,d4,29,  5697
    40,c9,40,f0,01,60,a0,51,a9,00
82 data 85,d4,20,a1,1c,a5,d4,29,40,c9,40,  6024
    d0,ee,a5,de,38,e9,0d,b0,02,a9
83 data 00,85,de,a5,e6,18,69,0a,90,02,e6,  5789
    e7,85,e6,a9,4f,8d,11,ff,a9,02
84 data 8d,10,ff,a9,20,8d,09,ff,60,18,a0,  5849
    06,a2,00,20,f0,ff,a5,e7,a6,e6
85 data 20,5f,a4,18,a2,00,a0,0f,20,f0,ff,  5916
    a9,00,a6,2d,20,b7,38,a0,1d,a2
86 data 00,18,20,f0,ff,a9,00,a6,de,20,5f,  5017
    a4,18,a2,00,a0,23,20,f0,ff,a9
87 data 00,a6,dd,20,5f,a4,a2,20,a5,de,c9,  6246
    0a,b0,03,8e,1e,0c,a5,dd,c9,64
88 data b0,03,8e,25,0c,c9,0a,b0,11,4c,ed,  6396
    23,85,e7,85,e8,85,2d,85,2b,ea
89 data ea,60,8e,24,0c,60,a5,e1,29,02,c9,  6287
    02,f0,06,20,3b,23,4c,c1,20,a5
90 data e1,29,fd,85,e1,a9,00,85,2b,20,26,  5537
    1f,20,3b,23,4c,25,21,a5,df,85
91 data d2,85,d4,a5,e0,85,d3,38,e9,04,85,  5888
    d5,a5,e2,a0,00,91,d2,ad,14,06
92 data 91,d4,c8,a5,e3,91,d2,ad,15,06,91,  6168
    d4,a0,28,a5,e4,91,d2,ad,16,06
93 data 91,d4,c8,a5,e5,91,d2,ad,17,06,91,  6048
    d4,a9,00,85,d4,c6,d3,a0,d9,20
94 data a1,1c,a0,d8,20,a1,1c,e6,d3,a5,e1,  5273
    29,01,c9,01,f0,0b,a5,d4
```

Listing 3: Das Versteck des Schwertes ist auf einer Karte verzeichnet

WWW.HOMECOMPUTERWORLD.COM

```
 0 rem      shogun lader 4
 1 rem
 2 rem     (c)   1987  by
 3 rem
 4 rem     olaf panz
 5 rem
 6 rem     drosselgasse  4
 7 rem
 8 rem     2085 marschacht
 9 rem
10 poke 56,56:clr :for t=14642 to 16382:   6496
   read a$:poke t,dec (a$):pr=int ((pr+t*
   dec (a$))/256)+t:next t
11 if pr<>25725 then print "fehler in      2972
    datas !"
12 data 29,40,c9,40,d0,2a,20,eb,25,a5,e1,  6007
   29,fe,85,e1,a9,0f,85,d3,ad,00
13 data ff,29,3f,c9,25,b0,f7,18,69,47,85,  5766
   d2,a9,00,85,d4,20,89,1c,a5,d4
14 data 29,40,c9,40,f0,e3,a5,d2,38,e9,28,  5568
   b0,02,c6,d3,85,d2,20,89,1c,a5
15 data d4,29,2f,c9,00,d0,c9,a5,d5,29,17,  5553
   c9,00,d0,c1,a5,d4,29,10,c9,10
16 data d0,08,a5,d5,c9,08,29,08,f0,b1,a5,  5893
   d2,85,d4,a5,d3,38,e9,04,85,d5
17 data a0,00,b1,d2,85,e2,b1,d4,8d,14,06,  5489
   c8,b1,d2,85,e3,b1,d4,8d,15,06
18 data a0,28,b1,d2,85,e4,b1,d4,8d,16,06,  5607
   c8,b1,d2,85,e5,b1,d4,8d,17,06
19 data a9,3a,20,42,1a,a5,d3,85,e0,a5,d5,  6217
   85,d3,a9,0c,20,33,1a,a6,d2,86
20 data df,ee,3b,05,ee,3b,05,e0,50,90,05,  5675
   a2,00,8e,3b,05,60,20,89,1c,a5
21 data d4,29,df,c9,00,f0,02,38,60,a5,d4,  5999
   29,1f,c9,00,d0,f6,4c,42,29,a9
22 data 77,8d,99,1f,a9,74,85,3f,a9,1d,85,  6700
   40,4c,5e,20,a9,0c,8d,99,1f,a9
23 data 1f,85,3f,a9,34,85,40,a5,e6,85,41,  6401
   a5,e7,85,42,20,71,25,20,63,20
24 data 20,71,25,a5,41,85,e6,a5,42,85,e7,  6123
   60,ea,ea,a6,dd,a5,de,85,dd,86
25 data de,a2,00,bd,10,06,a8,bd,18,06,9d,  5813
   10,06,98,9d,18,06,e8,e0,04,d0
26 data ed,a6,d9,a5,db,86,db,85,d9,a6,da,  6077
   a5,dc,86,dc,85,da,a6,e8,a5,43
27 data 85,e8,86,43,a5,e1,a6,44,86,e1,85,  5773
   44,60,a5,3f,c9,74,d0,68,ad,08
28 data ff,49,ff,85,d4,2a,29,80,05,d4,49,  5669
   ff,60,a5,d5,38,e9,0c,85,d5,a0
29 data 00,a5,d4,38,e9,28,b0,0b,c6,d5,a6,  6229
   d5,e0,ff,d0,03,84,d5,60,85,d4
30 data c8,4c,cd,25,85,d4,c8,4c,cd,25,a5,  6057
   d3,38,e9,05,85,d3,a0,d8,b1,d2
31 data aa,c8,b1,d2,a8,a5,d3,18,69,05,85,  5535
   d3,e0,77,f0,04,c0,77,d0,03,4c
32 data e4,1d,a5,de,38,e9,50,b0,02,a9,00,  5894
   85,de,60,e6,2d,18,60,20,4a,26
33 data 4c,f2,29,86,48,a5,d2,18,65,48,90,  6206
   02,e6,d3,85,d2,60,86,48,a5,d2
34 data 38,e5,48,b0,02,c6,d3,85,d2,60,a6,  5694
   db,86,d4,a6,dc,86,d5,4c,c4,25
35 data a5,d2,85,d4,a5,d3,85,d5,20,c4,25,  6223
   a5,d4,85,45,a5,d5,85,46,a5,df
36 data 85,d4,a5,e0,85,d5,20,c4,25,a4,d5,  6148
   c4,46,90,47,a6,d4,e4,45,d0,36
37 data a5,89,29,04,c9,04,f0,19,a2,02,20,  6408
   31,26,20,89,1c,a2,02,20,23,26
38 data a5,d4,c9,00,d0,03,a9,7a,60,a9,76,  6480
   60,a2,02,20,23,26,20,89,1c,a2
39 data 02,20,31,26,a5,d4,c9,00,f0,ea,d0,  6018
   e5,e8,e4,45,f0,c5,ca,ca,e4,45
40 data f0,bf,20,3f,26,a4,d5,c8,c8,c4,46,  5868
   d0,1c,a6,d4,e4,45,d0,0b,a0,50
41 data b1,d2,c9,77,d0,a6,a9,fd,60,e8,e4,  6459
   45,f0,bf,ca,ca,e4,45,f0,a0,a0
42 data 27,b1,d2,c9,13,d0,1e,a2,50,20,31,  5848
   26,20,89,1c,a2,50,20,23,26,a5
43 data d4,c9,00,d0,13,a5,d5,29,df,c9,00,  5584
   d0,0b,a9,7e,60,a0,02,b1,d2,c9
44 data 14,f0,da,20,3f,26,a4,d5,88,88,c4,  6227
   46,d0,7b,a6,d4,e4,45,d0,03,a9
45 data ff,60,ca,e4,45,d0,12,a0,51,b1,d2,  5835
   c9,04,d0,03,a9,75,60,c9,10,f0
46 data f9,a9,f7,60,e8,e8,e4,45,d0,12,a0,  6456
   50,b1,d2,c9,09,d0,03,a9,79,60
47 data c9,15,f0,f9,a9,fb,60,e8,e4,45,d0,  6067
   1b,a2,02,20,31,26,20,89,1c,a2
48 data 02,20,23,26,a5,d4,c9,00,d0,2b,a5,  5707
   d5,29,df,c9,00,f0,dd,ca,ca,ca
49 data ca,e4,45,d0,1b,a2,02,20,23,26,20,  6186
   89,1c,a2,02,20,31,26,a5,d4,c9
50 data 00,d0,08,a5,d5,29,df,c9,00,f0,a2,  5996
   a0,28,b1,d2,c9,77,d0,1b,a0,4f
51 data b1,d2,c9,20,d0,13,a0,52,b1,d2,c9,  6330
   20,d0,0b,a5,89,c9,fe,f0,04,c9
52 data fd,d0,01,60,a4,d5,c4,46,d0,5c,a6,  5994
   45,e8,e8,e4,d4,d0,1b,ad,00,ff
53 data 29,01,c9,01,d0,03,a9,77,60,ad,00,  6140
   ff,29,02,c9,02,d0,03,a9,75,60
54 data a9,76,60,e8,e4,d4,f0,ec,e8,e4,d4,  6439
   f0,f3,a6,45,ca,ca,e4,d4,d0,1b
55 data ad,00,ff,29,01,c9,01,d0,03,a5,7b,  5681
   60,ad,00,ff,29,02,c9,02,d0,03
56 data a9,79,60,a9,7a,60,ca,e4,d4,f0,ec,  5932
   ca,e4,d4,f0,f3,d0,0b,c8,c4,46
57 data f0,9f,88,88,c4,46,f0,99,a4,d5,c4,  6484
   46,d0,22,a6,d4,e4,45,90,0e,a0
58 data 2a,b1,d2,c9,43,d0,03,a9,f6,60,a9,  6121
   f7,60,a0,27,b1,d2,c9,43,d0,03
59 data a9,fa,60,a9,fb,60,c8,c4,46,f0,d9,  5916
   c8,c4,46,f0,d4,a4,d5,88,c4,46
60 data f0,cd,88,c4,46,f0,c8,4c,96,29,ea,  6107
   90,03,4c,d5,28,a0,50,b1,d2,c9
61 data 77,d0,03,a9,fd,60,c9,78,d0,03,a9,  5922
   fb,60,c8,b1,d2,c9,77,d0,03,a9
62 data f7,60,a5,89,29,04,c9,04,d0,eb,a5,  6429
   89,29,08,c9,08,d0,ed,a6,d4,e4
63 data 45,90,1e,a0,50,b1,d2,a2,00,86,d5,  6243
   20,a3,1c,a5,d5,29,20,c9,20,f0
64 data d4,c8,a5,d5,29,08,c9,08,f0,e6,d0,  5098
   bf,a0,50,b1,d2,a2,00,86,d5,20
65 data a3,1c,a5,d5,29,20,c9,20,f0,ac,88,  6225
   a5,d5,29,08,c9,08,f0,e6,d0,ab
66 data a0,28,b1,d2,c9,77,d0,03,a9,fe,60,  5732
   a5,89,29,04,c9,04,d0,32,a5,89
67 data 29,08,c9,08,d0,06,a6,d4,e4,45,90,  5622
   24,a0,00,b1,d2,a2,00,86,d5,86
68 data d4,20,a3,1c,c8,a5,d5,29,20,c9,20,  5950
   d0,03,a9,f7,60,a5,d4,29,2f,c9
69 data 00,f0,e1,a9,fb,60,a0,28,b1,d2,a2,  6358
   00,86,d5,86,d4,20,a3,1c,88,a5
70 data d5,29,20,c9,20,f0,e7,a5,d4,29,2f,  5664
   c9,00,f0,e4,d0,d2,4c,9b,33,a9
71 data fd,85,89,60,a2,28,20,31,26,20,d4,  6543
   29,a2,28,20,23,26,a5,d4,29,40
72 data c9,40,f0,2f,18,60,18,60,a5,89,ea,  5886
   ea,c9,fb,d0,11,a0,27,b1,d2,c9
73 data 43,d0,06,a9,fb,85,89,60,ea,a5,89,  6097
   60,a5,89,ea,ea,c9,f7,d0,f5,a0
74 data 2a,b1,d2,c9,43,d0,ed,f0,e5,a6,d2,  5837
   e4,d9,d0,08,a6,d3,e4,da,d0,02
75 data 18,60,38,60,a5,89,c9,fd,f0,04,c9,  6079
   fe,d0,06,a0,00,b1,d2,c9,77,f0
76 data 26,a4,46,c4,d5,4c,62,28,a0,4f,b1,  5889
   d2,c9,43,d0,0a,a2,fb,a5,e1,09
77 data 40,4c,07,2b,60,a0,52,b1,d2,c9,43,  6152
   d0,df,a2,f7,4c,b9,29,4c,a8,33
78 data 00,00,00,a0,00,b1,d2,84,d4,84,d5,  5601
   20,a3,1c,a5,d4,29,40,c9,40,f0
79 data 01,60,84,d4,c8,b1,d2,20,a3,1c,60,  5388
   ea,85,89,a0,00,b1,d2,c9,77,f0
80 data 03,4c,5b,29,a5,89,ea,ea,c9,fb,d0,  6514
   11,88,c6,d3,b1,d2,e6,d3,c9,43
81 data d0,ea,20,a8,29,4c,f2,29,a5,89,ea,  5657
   ea,c9,f7,d0,dc,a0,02,b1,d2,c9
82 data 43,d0,d4,f0,e8,c9,20,d0,04,a9,00,  5892
   85,2b,aa,a5,e1,29,40,c9,40,d0
83 data 0a,a5,e1,29,bf,85,e1,a9,ff,85,89,  6274
   8a,4c,42,1a,08,f4,45,7c,00,18
84 data 16,25,04,00,28,15,15,04,00,28,15,  5749
   0d,04,00,c8,f4,15,7c,00,08,14
85 data 25,04,00,08,14,45,04,00,08,14,45,  5234
   7c,00,00,00,00,00,00,00,00,00
86 data 00,00,00,00,00,00,00,00,60,fc,03,  5138
   00,00,60,fc,03,00,00,00,60,00
87 data 00,00,60,60,00,00,00,60,60,00,00,  5264
   00,60,60,00,00,00,60,60,00,00
88 data 00,a9,9f,20,06,34,a2,02,8e,19,ff,  5315
   8e,15,ff,a9,93,20,d2,ff,a2,00
```


WWW.HOMECOMPUTERWORLD.COM

```
92 data 12,ff,09,04,8d,12,ff,20,4b,1d,20,   5674
   30,15,4c,fb,2b,85,e1,8a,a6,d4
93 data e4,45,d0,03,4c,a8,29,60,20,87,18,   6602
   a2,00,86,d2,86,d4,a2,0c,86,d3
94 data a2,08,86,d5,a0,00,b1,d2,c9,53,90,   4766
   08,c9,77,b0,04,a9,00,91
89 data 86,d2,86,d4,a2,0c,86,d3,a2,01,86,   5771
   d5,a4,d4,b9,49,2a,a2,51,24,d5
90 data d0,02,a2,20,8a,a0,00,91,d2,e6,d2,   6149
   a6,d2,e0,00,d0,02,e6,d3,a5,d5
91 data 18,65,d5,85,d5,90,db,e6,d4,a6,d4,   6109
   e0,5a,90,cf,a9,00,8d,11,ff,ad
```

Listing 4: Die Karte besteht aus 60 Teilen.

```
 0 rem      shogun lader 5
 1 rem
 2 rem     (c)   1987  by
 3 rem
 4 rem     olaf panz
 5 rem
 6 rem     drosselgasse  4
 7 rem
 8 rem     2095 marschacht
 9 rem
10 poke 56,56:clr :for t=14642 to 16382:   6496
   read a$:poke t,dec (a$):pr=int ((pr+t*
   dec (a$))/256)+t:next t
11 if pr<>18546 then print "fehler in      2880
    datas !"
12 data d4,c8,c0,00,d0,ed,e6,d5,e6,d3,a6,  6193
   d3,e0,0e,d0,e1,a0,00,bd,00,3f
13 data 29,1f,c9,00,d0,10,c8,c0,00,d0,f2,  5992
   a5,d6,c5,d7,d0,05,a9,52,8d,81
14 data 0f,4c,ff,36,a5,4a,c9,03,90,0a,c9,  5815
   03,f0,03,20,48,25,20,48,25,20
15 data 48,25,a5,de,c9,00,f0,03,4c,fe,2b,  6300
   a9,2f,20,b7,33,a9,f7,8d,0f,ff
16 data a9,03,8d,10,ff,ce,0f,ff,ae,0f,ff,  6693
   e0,00,d0,0a,ce,10,ff,ae,10,ff
17 data e0,7c,f0,0a,a0,00,c8,c0,00,d0,fb,  5403
   4c,8d,2b,a6,db,86,d2,a6,dc,86
18 data d3,a9,29,20,42,1a,a9,4f,8d,11,ff,  4918
   ee,10,ff,a0,00,a2,00,e8,ee,19
19 data ff,e0,00,d0,f8,c8,c0,20,d0,f1,ad,  5955
   11,ff,29,0f,38,e9,01,85,d4,c9
20 data 00,f0,0d,ad,11,ff,29,f0,05,d4,8d,  6174
   11,ff,4c,c0,2b,a9,00,8d,11,ff
21 data a6,d6,bd,00,3e,29,7f,9d,00,3e,20,  6357
   14,2b,20,e1,ff,d0,29,c6,00,a0
22 data 00,c8,c0,00,d0,fb,e8,e0,00,d0,f4,  5652
   ee,19,ff,20,e1,ff,d0,f8,a9,d7
23 data 8d,19,ff,a2,00,a0,00,c8,c0,00,d0,  4908
   fb,e8,e0,00,d0,f4,20,8b,23,a0
24 data 00,a2,00,e8,e0,00,d0,fb,c8,c4,49,  5852
   d0,f4,a6,d6,bd,00,3f,29,10,c9
25 data 10,d0,06,ad,00,ff,8d,99,0b,4c,75,  5473
   2c,a8,a9,00,85,d4,98,20,a3,1c
26 data a5,d4,29,40,c9,40,d0,02,a0,20,98,  6079
   60,a2,00,b5,e2,20,51,2c,95,e2
27 data e8,e0,04,d0,f4,60,a5,4a,c9,04,90,  6242
   06,20,66,2c,20,12,24,20,66,2c
28 data 20,12,24,a5,4a,c9,01,d0,06,20,38,  5976
   25,20,cd,2c,20,38,25,20,9c,2c
29 data 4c,df,2c,a2,00,bd,10,06,c9,2d,d0,  5234
   0f,a4,d6,b9,00,3f,29,1f,99,00
30 data 3f,a9,20,4c,bc,2c,c9,2e,f0,ed,c9,  6564
   79,f0,e9,c9,3a,90,04,c9,3e,90
31 data eb,9d,10,06,e8,e0,04,d0,d2,60,20,  6136
   9c,2c,a0,00,a2,00,e8,e0,00,d0
32 data fb,c8,c0,20,d0,f4,60,20,8b,23,a5,  5505
   dd,c9,00,f0,03,4c,95,2d,a9,3f
33 data 8d,11,ff,ad,12,ff,29,fc,09,01,8d,  5912
   12,ff,a2,00,8e,0e,ff,a2,f7,8e
34 data 0f,ff,a2,03,8e,10,ff,ee,0e,ff,ee,  6311
   19,ff,ee,15,ff,ce,0f,ff,ae,0f
35 data ff,e0,ff,d0,0a,ce,10,ff,ae,10,ff,  5515
   e0,7c,f0,0a,a2,00,e8,e0,00,d0
36 data fb,4c,09,2d,a6,d9,86,d2,a6,da,86,  5251
   d3,a9,29,20,42,1a,a9,7e,8d,10
37 data ff,a9,40,8d,0f,ff,a9,5f,8d,11,ff,  5700
   ad,12,ff,29,fc,09,01,8d,12,ff
38 data a2,80,8e,0e,ff,a2,00,a0,00,ee,0e,  6095
   ff,ee,19,ff,ee,15,ff,c8,c0,00
39 data d0,f2,e8,e0,20,d0,eb,ad,11,ff,29,  5572
   0f,38,e9,01,85,d4,c9,00,f0,0d
40 data ad,11,ff,29,f0,05,d4,8d,11,ff,4c,  5957
   5b,2d,a9,00,8d,11,ff,4c,18,32
41 data a5,c6,c9,01,f0,f7,4c,29,34,ea,a2,  6424
   00,86,d2,a2,10,86,d3,a2,dd,86
42 data d4,a2,2d,86,d5,20,f6,33,29,ef,8d,  6468
   06,ff,78,a0,00,b1,d2,aa,b1,d4
43 data 91,d2,8a,91,d4,c8,c0,00,d0,f1,e6,  6021
   d3,e6,d5,a6,d3,e0,14,90,e5,58
44 data ad,06,ff,09,10,8d,06,ff,60,00,00,  5681
   00,4d,4f,00,00,00,00,00,06,1f
45 data 62,22,1e,01,10,10,18,36,13,16,38,  6240
   10,00,00,02,0c,30,60,30,0e,02
46 data 02,02,07,1a,32,1e,03,00,00,06,1a,  5939
   32,3c,11,1e,1f,0c,08,1e,08,08
47 data 08,10,00,1e,33,1d,01,09,11,0e,10,  6065
   10,70,1c,13,12,12,31,02,06,00
48 data 04,04,08,08,18,02,06,00,04,08,88,  5706
   48,38,20,20,20,7c,22,1c,32,19
49 data 10,10,10,10,10,10,09,0e,00,00,00,  6032
   63,5d,49,49,49,00,00,00,17,3a
50 data 12,12,09,00,00,18,26,21,42,24,18,  5736
   00,00,3c,13,12,3c,10,18,00,00
51 data 0f,3a,12,0f,02,06,00,00,13,3c,10,  5292
   10,08,18,00,00,07,18,20,1c,44
52 data 38,10,10,10,12,7c,10,10,30,00,00,  5690
   21,22,22,22,22,1c,00,00,21,22
53 data 22,14,14,08,00,00,81,92,aa,4a,44,  5723
   44,00,00,00,23,14,08,16,21,00
54 data 00,00,11,0d,02,0c,10,00,00,40,3e,  5371
   05,08,30,1f,00,00,00,00,00,00
55 data 00,00,00,fe,7f,03,1f,0f,00,00,f0,  5448
   7a,fd,fd,fe,78,f0,00,00,00,00
56 data 00,00,00,00,00,00,00,00,00,00,00,  4799
   00,00,00,00,00,00,00,00,00,00
57 data 0c,0c,0c,0c,0c,04,00,08,09,12,00,  4858
   00,00,00,00,00,00,00,00,00,00
58 data 00,12,24,fc,c0,00,00,00,00,00,00,  4993
   00,00,02,02,03,01,20,2e,0c,16
59 data 26,14,0c,fe,7f,3f,3f,03,00,00,00,  5659
   00,00,00,08,10,20,20,20,20,10
60 data 08,10,08,04,04,04,04,08,10,80,00,  5312
   00,04,c0,a0,90,a0,00,01,03,03
61 data 03,03,03,01,00,00,00,00,00,04,0c,  5122
   10,00,00,00,00,66,18,00,00,00
62 data 00,00,00,00,08,10,00,ff,99,99,ff,  5547
   ff,99,99,ff,18,24,4a,52,64,24
63 data 28,10,08,38,48,08,08,08,08,18,08,  6044
   34,44,08,10,20,22,3c,1c,02,02
64 data 0c,02,02,1c,20,18,28,4a,7c,08,08,  5381
   08,18,3c,42,50,68,04,08,30,40
65 data 06,09,10,38,44,42,24,18,02,7c,08,  5823
   10,10,10,20,20,0c,32,24,18,24
66 data 22,14,08,18,24,24,18,08,08,48,30,  5544
   00,00,08,10,00,08,10,00,00,00
67 data 00,04,00,04,0c,10,30,30,30,30,30,  5883
   30,30,30,0c,0c,0c,0c,0c,0c,0c
68 data 0c,00,00,00,00,00,00,00,00,18,24,  4977
   02,1c,20,1c,00,08,00,00,00,00
69 data 00,01,03,0e,00,0f,3f,7f,fc,e0,80,  5500
   00,00,fe,fc,fc,08,00,00,00,38
70 data 50,88,0e,07,03,01,00,00,00,00,00,  6171
   80,c0,e4,78,38,3c,3c,3e,38,70
71 data 60,40,00,00,00,00,00,03,07,0f,3c,  5914
   3f,77,66,c0,80,80,00,3f,7e,3e
72 data 1e,0e,06,02,00,00,00,00,00,00,00,  4910
   80,00,00,80,00,00,00,00,00,00
73 data 00,01,07,0f,0e,1c,18,38,60,d0,88,  5806
   04,06,07,07,0f,70,f0,78,1c,0e
74 data 03,01,00,1e,3c,38,70,e0,c0,80,00,  5964
   00,02,04,07,0c,0c,8c,cc,0e,1e
75 data 1f,3f,1f,0f,06,04,0c,1c,38,70,70,  6081
   30,10,00,00,3c,7e,fc,f8,10,10
76 data 20,60,e1,fe,f8,c0,00,00,00,00,07,  5592
   1f,3e,3e,3c,1c,1c,1c,1c,0c,0c
77 data 0c,04,06,07,1f,01,00,00,00,00,00,  5313
   00,00,00,01,07,1f,1f,1f,3e,0e
78 data 0e,04,06,0f,1f,7c,f0,00,00,00,81,  4979
   7f,1f,07,02,00,00,00,00,00,00
79 data 00,02,02,07,1f,26,0e,0c,0c,0c,1c,  6307
   1c,1c,1c,18,38,38,3e,3f,60,00
```


WWW.HOMECOMPUTERWORLD.COM

```
80 data 00,00,00,00,00,c0,00,80,00,00,00,   5057
   00,00,07,ff,fe,1c,08,08,18,18
81 data 18,18,18,1c,1c,1e,0e,1e,1c,10,00,   5296
   00,00,00,00,00,80,00,00,00,00
82 data 00,00,00,00,00,00,00,00,00,04,08,   5318
   08,08,08,08,08,08,08,18,00,00
83 data 00,00,00,00,00,3f,01,01,01,01,01,   5573
   8d,df,21,1d,7f,9c,1c,0c,0c,0c
84 data 1c,fe,fc,18,10,10,18,18,38,1e,1e,   6062
   1f,1e,3c,38,30,20,18,08,0e,02
85 data 02,9e,f2,01,80,80,87,81,8d,83,81,   5886
   81,01,01,c3,02,22,c2,46,44,8f
86 data 1c,70,40,40,40,40,40,f1,38,0e,02,   5925
   02,02,02,02,80,80,c4,4a,41,40
87 data 60,23,01,01,11,21,41,81,91,e1,87,   5865
   8c,80,80,80,80,80,ff,c4,24,04
88 data 04,04,04,04,ff,40,40,40,40,40,40,   5850
   40,c0,02,02,02,02,02,02,02,03
89 data 24,21,22,20,20,20,20,ff,81,c1,81,   5902
   41,01,01,01,ff,87,98,60,47,58
90 data 60,8f,f0,e1,19,06,e2,1a,06,f1,0f,   6075
   10,18,18,7f,fe,18,18,08,87,9d
91 data 66,47,5f,7c,bf,f2,e1,59,c6,e2,fa,   6742
   76,f9,8f,a7,9f,66,4f,5c,6c,8f
92 data f8,e9,d9,c6,e2,7a,66,f1,2f,ff,7e,   6540
   3c,18,66,c3,81,00,7f,fe,48,26
93 data 7c,83,00,00,00,a9,9f,20,d2,ff,a9,   5855
   8e,20,d2,ff,a9,08,20,d2,ff,a9
94 data a6,20,d2,ff,a9,14,20,d2,ff,a0,00,   4259
   c8,c0,00,d0,fb,b1,d2,20
```

Listing 5: Alle 60 Teile zu finden, ist Aufgabe in „Shogun"

```
 0 rem      shogun lader 6
 1 rem
 2 rem     (c)   1987  by
 3 rem
 4 rem     olaf panz
 5 rem
 6 rem     drosselgasse  4
 7 rem
 8 rem     2095 marschacht
 9 rem
10 poke 56,56:clr :for t=14642 to 16382:   6496
   read a$:poke t,dec (a$):pr=int ((pr+t*
   dec (a$))/256)+t:next t
11 if pr<>19806 then print "fehler in      2814
    datas !"
12 data d2,ff,e6,d2,a6,d2,e0,00,d0,02,e6,  5817
   d3,e4,d4,d0,dc,a6,d3,e4,d5,90
13 data d6,60,ad,07,10,c9,29,f0,03,20,9f,  6423
   2d,a9,04,8d,19,ff,8d,15,ff,4c
14 data df,32,47,41,4d,45,20,4f,56,45,52,  5795
   53,49,45,20,48,41,42,45,4e,20
15 data 45,53,20,4c,45,49,44,45,52,20,4e,  6655
   49,43,48,54,20,47,45,53,43,48
16 data 41,46,46,54,2c,44,41,53,20,5a,49,  6780
   45,4c,20,44,45,53,20,53,50,49
17 data 45,4c,45,53,20,5a,55,20,45,52,52,  6691
   45,49,43,48,45,4e,2e,53,49,45
18 data 20,46,41,4e,44,45,4e,20,20,20,20,  5804
   20,20,20,50,55,5a,5a,4c,45,2d
19 data 54,45,49,4c,45,55,4e,44,20,45,52,  6338
   52,45,49,43,48,54,45,4e,20,20
20 data 20,20,20,20,20,20,20,50,55,4e,4b,  6195
   54,45,1c,82,50,52,45,53,53,20
21 data 46,49,52,45,84,53,49,45,20,42,45,  6551
   53,47,49,45,47,54,45,4e,20,20
22 data 20,20,20,20,20,47,45,47,4e,45,52,  5493
   07,a9,93,20,d2,ff,a0,0f,a2,03
23 data 18,20,f0,ff,a2,32,86,d3,86,d5,a2,  5803
   2d,86,d2,a2,36,86,d4,20,de,31
24 data a2,07,a0,03,18,20,f0,ff,a2,5a,86,  5772
   d4,20,de,31,a2,09,a0,03,18,20
25 data f0,ff,a2,7c,86,d4,20,de,31,18,a2,  6083
   0d,a0,03,20,f0,ff,a2,99,86,d4
26 data 20,de,31,a2,10,a0,03,18,20,f0,ff,  5666
   a2,b7,86,d4,20,de,31,a2,16,a0
27 data 0e,18,20,f0,ff,a2,c2,86,d4,20,de,  5070
   31,a2,12,a0,03,18,20,f0,ff,a2
28 data de,86,d4,20,de,31,a2,0d,a0,0e,18,  5954
   20,f0,ff,a9,00,a6,2d,20,b7,38
29 data a2,10,a0,12,18,20,f0,ff,a5,e7,a6,  5908
   e6,20,5f,a4,a2,12,a0,12,18,20
30 data f0,ff,a5,4c,a6,4b,20,5f,a4,a9,00,  5623
   8d,08,ff,ad,08,ff,a8,29,40,c9
31 data 40,d0,07,98,29,80,c9,80,f0,ea,60,  6101
   8e,11,ff,a6,3f,e0,74,d0,01,60
32 data 4c,3d,29,20,3f,26,a6,d4,e4,45,b0,  6519
   03,4c,af,29,4c,c1,29,8d,11,ff
33 data e6,4b,a6,4b,e0,00,d0,02,e6,4c,a5,  6666
   49,49,ff,18,65,e6,90,02,e6,e7
34 data 85,e6,e6,49,a6,49,e0,ff,d0,0d,e6,  6341
   4a,a6,4a,e0,05,d0,05,c6,4a,20
35 data e7,33,60,a6,4d,a9,1f,38,e9,20,ca,  6311
   e0,00,d0,f8,85,49,60,ad,06,ff
36 data 78,a2,ce,8e,15,03,a2,0e,8e,14,03,  5651
   58,60,20,d2,ff,20,f9,33,ad,0f
37 data 10,c9,01,d0,03,20,9f,2d,a2,00,86,  5752
   4b,86,4c,4c,e7,33,20,20,25,90
38 data 04,a2,ff,86,89,60,a2,00,bd,20,10,  5427
   a8,bd,41,34,9d,20,10,98,9d,41
39 data 34,e8,e0,40,d0,ed,4c,1e,37,03,06,  5946
   06,02,03,07,07,0f,00,80,40,80
40 data 00,c8,f0,a0,07,03,07,06,0c,18,10,  5190
   18,80,80,c0,60,60,60,40,60,00
41 data 01,02,01,00,13,0f,05,c0,60,60,40,  5858
   c0,e0,e0,f0,01,01,03,06,06,06
42 data 02,06,e0,c0,e0,60,30,18,08,18,00,  5171
   00,a9,13,20,d2,ff,20,d2,ff,a9
43 data 93,20,d2,ff,a9,06,8d,19,ff,8d,15,  5669
   ff,a2,0c,86,d3,a2,0a,86,d2,a2
44 data 35,86,d5,a2,00,86,d4,a0,00,b1,d4,  5890
   91,d2,c8,c0,0e,d0,f7,a5,d4,18
45 data 69,0e,85,d4,a5,d2,18,69,28,85,d2,  6134
   c9,c8,90,e3,ad,10,10,c9,10,f0
46 data 03,20,9f,2d,a9,00,8d,11,ff,8d,12,  6065
   ff,78,a2,ec,8e,14,03,a2,34,8e
47 data 15,03,58,a9,00,85,db,a9,81,85,d8,  5804
   60,ad,00,08,78,18,69,02,a2,00
48 data 9d,00,08,e8,e0,00,d0,f8,4c,06,36,  5452
   40,41,42,63,20,20,20,20,20,20
49 data 20,20,20,20,43,44,20,64,65,20,20,  5991
   20,20,20,20,5a,20,20,46,47,4a
50 data 67,68,4b,4c,4f,52,54,57,5b,5f,62,  6472
   48,20,20,69,45,4d,4e,50,53,55
51 data 58,5c,60,20,20,20,20,20,20,20,20,  5631
   51,20,56,5e,5d,61,20,20,00,00
52 data 1d,1e,1e,09,29,1c,19,1d,21,25,1d,  6379
   11,2d,2d,2d,2d,1c,25,25,26,22
53 data 1c,25,25,26,22,1e,22,1e,1a,1d,21,  6062
   1f,29,29,1c,29,29,1e,16,0c,15
54 data 15,16,1e,12,16,12,0e,0a,06,02,01,  6084
   01,02,02,00,09,09,0a,06,00,09
55 data 09,0a,06,02,06,02,02,01,05,03,09,  5947
   09,04,15,15,1e,16,0d,15,1d,09
56 data 0a,02,0d,0e,02,11,19,26,22,1e,22,  5925
   1e,1a,1d,29,21,24,2d,29,2d,29
57 data 2d,29,15,1a,1e,1a,16,19,25,1d,20,  6131
   29,04,28,24,29,24,15,02,06,02
58 data 02,01,0d,05,08,09,00,00,a9,00,07,  5643
   01,5a,01,80,01,c6,01,06,02,3d
59 data 02,56,02,85,02,af,02,c2,02,e5,02,  6767
   05,03,21,03,2d,03,45,03,59,03
60 data 63,03,74,03,84,03,92,03,98,03,a4,  5814
   03,ae,03,b3,03,bc,0c,07,02,10
61 data 24,d8,50,19,ad,11,ff,a8,29,0f,c9,  6112
   04,90,09,a5,d8,29,0f,85,d8,4c
62 data 0e,ce,ee,11,ff,4c,0e,ce,30,03,4c,  6298
   c8,36,ad,11,ff,a8,29,0f,c9,00
63 data f0,03,4c,d9,36,e6,d9,a6,d9,e0,84,  6161
   90,06,a2,00,86,d9,e6,d8,20,f4
64 data 36,ea,09,40,85,d8,a6,d9,bd,49,35,  5336
   a8,29,03,aa,bd,02,36,85,db,98
65 data 6a,6a,29,0f,aa,a5,d8,29,07,c9,01,  5403
   d0,38,8a,18,69,0e,a0,00,8a,20
66 data df,36,8d,0e,ff,e8,b9,cf,35,8d,0f,  5733
   ff,c8,b9,cf,35,8d,10,ff,ad,12
67 data ff,29,fc,1d,cf,35,8d,12,ff,a2,10,  5736
   a5,d8,29,07,c9,04,90,02,a2,30
68 data 8e,11,ff,4c,0e,ce,c9,02,d0,09,a0,  5703
   00,8a,18,69,07,4c,6d,36,c9,03
69 data f0,bd,c9,05,d0,04,a8,4c,a5,36,c9,  6652
   04,d0,01,a8,4c,6d,36,8a,18,69
70 data 07,a8,4c,6d,36,c6,db,a6,db,e0,00,  5448
   d0,06,a5,d8,09,80,85,d8,4c,0e
71 data ce,ce,11,ff,4c,0e,ce,86,da,84,dc,  5904
   98,18,65,dc,85,dc,8a,18,65,da
72 data aa,a4,dc,bd,cf,35,60,a5,d8,29,0f,  5847
   c9,07,d0,02,a9,01,60,a5,2d,c9
73 data 3d,d0,0b,a4,d6,c4,d7,d0,05,a9,52,  5904
```


WWW.HOMECOMPUTERWORLD.COM

```
    8d,72,0f,a9,45,8d,72,0b,60,52
74 data 8d,72,0f,bd,00,3f,60,a2,00,bd,10,  5340
    06,c9,52,f0,08,e8,e0,04,d0,f4
75 data 4c,58,38,20,bf,38,a2,5a,86,d2,a2,  6035
    37,86,d3,86,d5,a2,c8,86,d4,20
76 data e3,31,20,84,33,a9,93,20,d2,ff,a9,  6843
    43,a2,00,9d,c0,0f,e8,e0,28,d0
77 data f8,4c,ff,39,11,11,11,11,11,11,11,  6699
    05,53,49,45,20,48,41,42,45,4e
78 data 20,44,41,53,20,53,43,48,57,45,52,  6478
    54,20,44,45,53,20,53,48,4f,47
79 data 55,4e,8d,8d,47,45,46,55,4e,44,45,  6672
    4e,2e,20,47,45,42,45,4e,20,53
80 data 49,45,20,45,53,20,4e,55,4e,20,53,  6718
    45,49,4e,45,4d,8d,8d,42,45,53
81 data 49,54,5a,45,52,2e,8d,8d,20,20,20,  6231
    20,20,20,20,20,20,20,05,82,50
82 data 52,45,53,53,20,46,49,52,45,84,11,  6860
    11,11,11,11,11,11,05,53,49,45
83 data 20,48,41,42,45,4e,20,41,4c,4c,45,  5999
    20,36,30,20,50,55,5a,5a,4c,45
84 data 2d,54,45,49,4c,45,20,47,45,46,55,  6899
    4e,44,45,4e,2e,8d,8d,53,55,43
85 data 48,45,4e,20,53,49,45,20,4e,55,4e,  6628
    20,44,41,53,20,53,43,48,57,45
86 data 52,54,20,44,45,53,20,53,48,4f,47,  6572
    55,4e,53,2e,8d,8d,45,53,20,49
87 data 53,54,20,49,4e,20,45,49,4e,45,4d,  6314
    20,44,45,52,20,4e,45,42,45,4e
88 data 2d,52,41,45,55,4d,45,2e,8d,8d,20,  6228
    20,20,20,20,82,50,52,45,53,53
89 data 20,46,49,52,45,84,a5,2d,c9,3c,f0,  6434
    03,4c,62,2b,20,bf,38,a2,37,86
90 data d3,e8,86,d5,a2,c9,86,d2,a2,58,86,  5431
    d4,20,e3,31,20,84,33,20,ce,38
91 data 20,14,2b,e6,2d,a6,d6,ad,00,ff,29,  5946
    01,a8,ad,00,ff,29,40,c9,00,d0
92 data 0a,8a,38,f9,a8,38,90,e8,4c,a3,38,  5749
    8a,18,79,a8,38,b0,de,85,d7,4c
93 data 62,2b,01,10,ea,ea,8a,38,e9,10,4c,  5941
    a0,38,ca,4c,8e,38,e0,3d,d0,01
94 data ca,4c,5f,a4,a2,00,b5,d7,9d,20,06,  4634
    e8,e0,08,d0,f6,4c,83,34
```

Listing 6: Als Gegner stellen sich erfahrene Karatekämpfer in den Weg

```
 0 rem      shogun lader 7
 1 rem
 2 rem      (c)   1987  by
 3 rem
 4 rem      olaf panz
 5 rem
 6 rem      drosselgasse  4
 7 rem
 8 rem      2095 marschacht
 9 rem
10 poke 56,56:clr :for t=14642 to 15936:  6680
    read a$:poke t,dec (a$):pr=int ((pr+t*
    dec (a$))/256)+t:next t
11 if pr<>18794 then print "fehler in      2746
     datas !
12 data 20,9f,2d,a2,00,bd,20,06,95,d7,e8,  5490
    e0,08,d0,f3,60,00,00,00,00,00
13 data 00,00,00,00,00,00,03,0c,30,4f,f0,  5599
    00,0f,7f,80,03,3c,c0,00,1f,e0
14 data 00,0f,f0,00,01,02,f0,0f,00,e0,1f,  5772
    e0,18,e4,00,fe,01,02,02,82,42
15 data 42,00,00,00,00,00,00,00,00,00,00,  4815
    00,00,00,00,00,00,00,00,00,00
16 data 00,00,00,00,03,02,01,02,05,06,01,  5795
    01,24,64,92,12,12,0a,99,27,24
17 data 24,24,14,14,14,0c,0c,00,00,00,00,  5267
    00,00,00,00,00,00,00,00,00,00
18 data 00,00,00,00,00,20,50,8a,77,0b,00,  5755
    00,01,03,03,06,04,8d,c8,de,fc
19 data 08,66,f6,f7,f7,0c,0c,0c,0c,0c,0c,  5667
    0c,0c,00,00,00,00,00,00,00,00
20 data 00,00,07,3f,e0,cf,bf,78,07,07,07,  6022
    c3,74,04,f8,f8,d9,e3,f7,ff,ff
21 data 7f,7d,1f,a7,cf,cf,df,bf,00,00,00,  5874
    18,98,98,98,d8,38,18,18,00,00
22 data 00,00,00,00,00,00,f0,e2,c7,47,45,  6074
    44,6c,28,50,f0,10,10,a0,c0,47
23 data 84,0f,0e,04,00,00,07,f8,00,00,00,  5574
    00,00,00,e0,1f,01,18,18,18,18
24 data 18,18,18,08,00,00,23,33,1f,0c,04,  5790
    01,68,e8,f8,b8,18,39,f9,f3,84
25 data 84,84,84,84,32,ce,03,00,00,00,00,  5688
    00,00,00,00,01,01,00,00,00,01
26 data 01,03,08,08,84,84,84,32,ce,03,00,  5530
    20,9f,2d,a2,00,8e,11,ff,20,d1
27 data 38,4c,4e,3a,a2,de,86,d2,a2,38,86,  5805
    d3,a2,12,86,d5,a2,98,86,d4,a0
28 data 00,b1,d2,aa,b1,d4,91,d2,8a,91,d4,  5369
    c8,c0,20,d0,06,a6,d3,e0,39,f0
29 data 0b,c0,00,d0,e7,e6,d3,e6,d5,4c,1d,  5571
    3a,60,a2,00,78,fe,c0,0b,e8,e0
30 data 28,d0,f8,4c,0e,ce,20,0d,3a,78,a2,  5698
    40,8e,14,03,a2,3a,8e,15,03,58
31 data a2,0e,86,d3,a2,f2,86,d2,a9,53,a2,  5875
    00,a0,00,91,d2,18,69,01,c8,c0
32 data 06,90,f6,a8,a5,d2,18,69,28,90,02,  6274
    e6,d3,85,d2,98,e8,e0,06,90,e2
33 data a2,00,8e,19,ff,8e,15,ff,ca,86,dd,  5870
    a2,58,86,d9,c6,2d,a2,00,a9,20
34 data 9d,10,06,e8,e0,0f,d0,f8,a2,0f,86,  5366
    dc,86,da,a2,52,86,d9,a2,60,86
35 data db,a2,63,86,de,20,48,25,20,48,25,  5990
    20,38,25,a6,de,e0,00,d0,12,a6
36 data db,86,d2,a6,dc,86,d3,a9,20,20,33,  5491
    1a,a2,20,4c,b0,3a,a2,00,a0,00
37 data c8,c0,00,d0,fb,e8,e0,40,d0,f4,a5,  5763
    d9,85,d4,a5,da,85,d5,20,c4,25
38 data a5,d4,c9,20,f0,09,a5,dd,c9,00,d0,  6167
    ba,4c,cd,3d,4c,b0,3b,11,11,11
39 data 11,11,11,11,48,45,52,5a,4c,49,43,  6185
    48,45,4e,20,47,4c,55,45,43,4b
40 data 57,55,4e,53,43,48,20,21,21,21,8d,  6796
    8d,53,49,45,20,48,41,42,45,4e
41 data 20,45,53,20,47,45,53,43,48,41,46,  6658
    46,54,20,44,45,4d,20,53,48,4f
42 data 47,55,4e,20,44,41,53,8d,8d,53,43,  6463
    48,57,45,52,54,20,5a,55,52,55
43 data 45,43,4b,20,5a,55,47,45,42,45,4e,  6454
    2e,20,4f,4c,41,46,20,50,41,4e
44 data 5a,20,55,4e,44,8d,8d,4d,41,52,43,  5901
    20,50,45,54,45,52,20,46,52,49
45 data 54,5a,45,20,47,52,41,54,55,4c,49,  6586
    45,52,45,4e,20,49,48,4e,45,4e
46 data 20,21,21,8d,8d,8d,20,20,20,20,20,  5786
    20,82,1f,50,52,53,53,20,46,49
47 data 52,45,84,20,c7,3d,a2,b8,8e,19,ff,  5827
    8e,15,ff,a2,3b,86,d3,86,d5,a2
48 data 02,86,d2,a2,b0,86,d4,20,de,31,20,  5966
    84,33,60,11,11,11,11,11,11,11
49 data 11,11,43,4f,50,59,52,49,47,48,54,  6176
    20,20,31,39,38,35,20,20,42,59
50 data 20,20,20,4f,4c,41,46,20,20,50,41,  6525
    4e,5a,8d,8d,20,20,44,52,4f,53
51 data 53,45,4c,47,41,53,53,45,20,34,20,  6427
    20,20,20,32,30,39,35,20,4d,41
52 data 52,53,43,48,41,43,48,54,8d,8d,20,  6272
    20,47,52,41,50,48,49,43,20,20
53 data 42,59,20,20,4d,41,52,43,20,50,45,  5986
    54,45,52,20,46,52,49,54,5a,45
54 data 8d,8d,20,20,20,20,53,50,49,45,4c,  5938
    8d,8d,20,20,20,20,4c,45,56,45
55 data 4c,8d,8d,20,20,20,20,53,50,45,45,  5828
    44,20,83,34,a2,cc,8a,19,ff,8e
56 data 15,ff,a2,3b,86,d3,e8,86,d5,a2,d0,  6233
    86,d2,a2,61,86,d4,20,de,31,a2
57 data 01,86,4d,86,4a,86,d7,4c,d1,3c,a9,  6700
    0c,a4,d7,18,69,50,88,c0,00,d0
58 data f8,85,d2,a0,0a,84,d3,a0,00,b1,d2,  5496
    e0,00,d0,05,29,7f,4c,aa,3c,09
59 data 80,91,d2,c8,c0,06,d0,ec,18,a0,0c,  5254
    a2,11,18,20,f0,ff,a9,00,a6,4a
60 data 20,5f,a4,a2,13,a0,0c,18,20,f0,ff,  5986
    a9,00,a6,4d,4c,5f,a4,ea,a2,ff
61 data 20,89,3c,a2,00,a0,00,c8,c0,00,d0,  5223
    fb,e8,e0,70,d0,f4,a9,00,8d,08
62 data ff,ad,08,ff,c9,ff,f0,f4,a8,a2,02,  5065
    0a,b0,03,4c,6b,3d,0a,90,fa,a2
63 data 00,98,4a,b0,10,a6,d7,e0,02,90,ce,  6102
    a2,00,20,89,3c,c6,d7,4c,d1,3c
64 data 4a,b0,10,a6,d7,e0,03,b0,bb,a2,00,  5398
    20,89,3c,e6,d7,4c,d1,3c,4a,b0
65 data 20,a6,d7,e0,02,90,a8,e0,02,d0,0b,  5562
    a5,4a,c9,02,90,9e,c6,4a,4c,d1
66 data 3c,a5,4d,c9,02,90,93,c6,4d,4c,d1,  5724
    3c,4a,b0,8b,a6,d7,e0,02,90,85
67 data e0,02,d0,0b,a5,4a,c9,04,b0,ea,e6,  5796
    4a,4c,d1,3c,a5,4d,c9,08,b0,df
68 data e6,4d,4c,d1,3c,a6,d7,e0,01,d0,f7,  5773
    20,a4,2a,4c,61,3c,01,02,03,01
69 data 0f,1c,36,22,80,40,c0,80,e0,70,d8,  5767
    88,23,ae,14,10,e0,0c,f0,03,20
70 data 9f,2d,a2,00,bd,77,3d,9d,00,10,e8,  5765
    e0,10,d0,f5,a2,00,a9,11,20,d2
71 data ff,a0,00,c8,c0,00,d0,fb,e8,e0,17,  6044
    d0,ef,a9,94,20,d2,ff,a9,08,20
72 data d2,ff,a9,13,20,d2,ff,20,d2,ff,4c,  6823
    61,3c,20,0d,3a,4c,83,34,20,0d
73 data 3a,a9,93,20,d2,ff,a9,84,20,d2,ff,  3958
    4c,eb,2c
```

Listing 7: Um das Spiel zu gewinnen, muß man hart trainieren

WWW.HOMECOMPUTERWORLD.COM

Angriff auf Schlummerland

System: C64
Typ: Actionspiel
Sprache: Basic

Steff'sche Flugzeuge greifen das friedliche Schlummerland an. Nur eine Flak steht zwischen den Aggressoren und den Schlummerländern.

Bis zu 9 Spieler in Reihenfolge steuern die Zielangaben der Flak. Von ihnen hängt ab, wie viele Bomber ihr Ziel erreichen. Nach Starten des Programmes mit RUN sind die Spielerzahl und deren Namen einzugeben. Danach beginnt das Spiel. Ein Kampfflugzeug erscheint. Mit den Tasten:

C = Schuß im Winkel von 45 Grad nach links oben
V = Schuß im Winkel von 22 Grad nach links oben
B = Schuß im Winkel von 22 Grad nach rechts oben
N = Schuß im Winkel von 45 Grad nach rechts oben
Leer-Taste = Schuß senkrecht nach oben

gibt man die Feuerbefehle. Je Spiel stehen bis zu 60 Granaten zur Verfügung. Ab dem 30. Schuß bricht das Programm ab, wenn man die ‹RETURN›Taste drückt. FLAK gibt nach jedem Schuß die Trefferquote aus. Gewonnen hat, wer den höchsten Durchschnittswert erreicht.

softteam

```
 1 goto 63030                                    635
 2 poke vo,3:poke sz,33:poke hf,uf:poke         3340
   sz,32:return
 3 for mz=1 to 255 step 4:poke vo,7:poke        5363
   sz,33:poke hf,mz:poke sz,32:next mz:
   return
 4 for mz=255 to 1 step -4:poke vo,7:poke       5326
   sz,33:poke hf,mz:poke sz,32:next mz:
   return
 5 print "[clr,5spaces]*** flak -               3687
   schiessen ***":print :print :z%=1
 6 print "sie muessen mit dem ortsfesten        3339
   flak -
 7 print "geschuetz moeglichst viele            3910
   jagdflugzeuge
 8 print "[up]abschiessen. die zulaessige       3936
   schusszahl
 9 print "liegt zwischen 30 und 60              3329
   schuessen.
10 print "danach wird die treffer-rate          3577
   bewertet.
11 print "nach dem 30.schuss kann mit           3784
   [rvs]return[off] ab-
12 print "gebrochen werden.                     1918
13 print "folgende 5 schuesse sind              4452
   moeglich:":print
14 print "1) taste [rvs,2spaces]c               3760
   [2spaces,off] : 45 grad nach links
15 print "2) taste [rvs,2spaces]v               3704
   [2spaces,off] : 22 grad nach links
16 print "3) taste [rvs]space[off] :            3662
   senkrecht nach oben"
17 print "4) taste [rvs,2spaces]b               3405
   [2spaces,off] : 22 grad nach rechts
18 print "5) taste [rvs,2spaces]n               3413
   [2spaces,off] : 45 grad nach rechts
19 print :print "(winkel jeweils von der        4540
   senkrechten aus)
20 print :print :print "bitte druecken          5159
   sie [rvs]return[off] zum start !":
   gosub 3:gosub 4
21 get a$:if a$<>chr$ (13) goto 21              1971
22 print "[clr,down]anzahl der spieler          2975
   (1-9) ? ";
23 for i=1 to 100:get a$:if a$<>"" goto         2157
   27
24 next :print "[rvs] [left]";                   581
25 for i=1 to 100:get a$:if a$<>"" goto         2157
   27
26 next :print "[off] [left]";:goto 23           939
27 if a$<"1" or a$>"9" goto 23                  1607
28 print "[off]"a$:x%=val (a$)                  1102
29 print "[down]druecken sie [rvs]return        3694
   [off] nach jedem namen!"
30 print "[down]":for k=1 to x%:print           6728
   "name von spieler"k"? [10cbm @,10left]
   ";:b$=""
31 for j=1 to 50:get a$:if a$<>"" goto 34       2633
32 next :print "[rvs,cbm @,left]";:for j=       2703
   1 to 50:get a$:if a$<>"" goto 34
33 next :print "[off,cbm @,left]";:goto         1059
   31
34 a=asc (a$):if a=13 then print "[off,         2679
   cbm @]";:goto 39
35 if a=20 and len (b$)>1 then b$=left$         5072
   (b$,len (b$)-1):print "[off,cbm @,
   2left,cbm @,left]";:goto 31
36 if a=20 and len (b$)=1 then b$="":           4280
   print "[off,cbm @,2left,cbm @,left]";:
   goto 31
37 if a=32 or (64<a and a<98) then print        3341
   "[off]"a$;:b$=b$+a$:if len (b$)>9 goto
   39
38 goto 31                                       435
39 n$(k)=b$:print :print :next k                1792
40 print :print "[4spaces]druecken sie          3059
   [rvs]return[off] zum start !
41 print :print "[7spaces,rvs]home[off]         3262
   zum spiel-abbruch !
42 get a$:if a$<>chr$ (13) and a$<>"            2839
   [home]" goto 42
43 if a$="[home]" then p%=p%-1:if p%<1          3366
   then p%=x%
44 if a$="[home]" then print "[down]ich         4203
   hoffe es hat spass gemacht,"n$(p%)
45 if a$="[home]" then print "[clr]":end        1536
46 pl%=pl%+1:if pl%>x% then pl%=1               2205
48 t%=0:s%=-1                                   1189
50 print "[clr,23down]";                        2114
60 print "[rvs,9spaces]flak[5spaces]-           3049
   [4spaces]schiessen[8spaces,off,up]";
80 print "[home,4down,7right]";:for j=1         5862
   to len (n$(pl%)):print mid$ (n$(pl%),
   j,1);" [down]";:next j
90 rem print"[illegible]
   [illegible]";
100 p=33665:f=32808+40*int (20*rnd (5)):c=      4685
    40:g=f-39
105 x=int (4*rnd (5)+6)/10:ri%=int ( rnd        3472
    (7)+.5)
110 poke p,233:poke p+1,214:poke p+2,223:       3330
    poke p-39,65
120 s=p-79                                       496
125 if s%=-1 then for j=1 to 4000:next j        1872
```


WWW.HOMECOMPUTERWORLD.COM

```
126 print "[home,4down,7right]";:for j=1 to len (n$(pl%)):print "[2spaces,down] ";:next j   4173
128 print "[home,22down]"                                                                    1793
130 get a$:if a$<>"" goto 130                                                                1684
131 s%=s%+1:r%=0:if s%>0 then r%=t%/s%*100                                                   3443
132 print "[rvs]"s%"[left] schuss[2spaces] "t%"[left] treffer[3spaces]rate: "r%"[left] % [up]"   4006
133 if s%=60 goto 1500                                                                       1298
135 get a$:if a$="" then gosub 990:goto 135                                                  2169
140 if a$=" " goto 500                                                                       1273
150 if a$="b" goto 800                                                                       1344
170 if a$="v" goto 900                                                                       1475
180 if a$="n" goto 600                                                                       1471
190 if a$="c" goto 700                                                                       1375
200 if a$=chr$ (13) and s%>29 goto 1500                                                      2021
210 gosub 1000:goto 132                                                                      1035
500 for i=0 to 20                                                                            762
501 uf=i+50:gosub 2                                                                          1119
502 i%=i*40                                                                                  771
505 j%=peek (s-i%):if j%=60 or j%=62 or j%=67 goto 1100                                      3735
510 poke s-i%,46                                                                             820
520 poke s-(i+1)*40,46:poke s-i%,32                                                          2872
530 gosub 1000:next i                                                                        958
540 poke s-i%,32                                                                             1009
550 goto 130                                                                                 568
600 for i=0 to 20                                                                            762
601 uf=i+60:gosub 2                                                                          1151
602 i%=39*i                                                                                  880
605 j%=peek (s-i%):if j%=60 or j%=62 or j%=67 goto 1100                                      3735
610 poke s-i%,46                                                                             820
620 poke s-(i+1)*39,46:poke s-i%,32                                                          2906
630 gosub 1000:next i                                                                        958
640 poke s-i%,32                                                                             1009
650 goto 130                                                                                 568
700 for i=0 to 20                                                                            762
701 uf=i+60:gosub 2                                                                          1151
702 i%=41*i                                                                                  760
705 j%=peek (s-i%):if j%=60 or j%=62 or j%=67 goto 1100                                      3735
710 poke s-i*41,46                                                                           1159
720 poke s-(i+1)*41,46:poke s-i%,32                                                          2876
730 gosub 1000:next i                                                                        958
740 poke s-i%,32                                                                             1009
750 goto 130                                                                                 568
800 for i=0 to 20                                                                            762
801 uf=i+55:gosub 2                                                                          1184
802 i%=int (i*39.5)                                                                          1323
805 j%=peek (s-i%):if j%=60 or j%=62 or j%=67 goto 1100                                      3735
810 poke s-i%,46                                                                             820
820 poke s-int ((i+1)*39.5),46:poke s-i%,32                                                  3669
830 gosub 1000:next i                                                                        958
840 poke s-i%,32                                                                             1009
850 goto 130                                                                                 568
900 for i=0 to 20                                                                            762
901 uf=i+55:gosub 2                                                                          1184
902 i%=int (i*40.5)                                                                          1187
905 j%=peek (s-i%):if j%=60 or j%=62 or j%=67 goto 1100                                      3735
910 poke s-i%,46                                                                             820
920 poke s-int ((i+1)*40.5),46:poke s-i%,32                                                  3533
930 gosub 1000:next i                                                                        958
940 poke s-int (i*40.5),32                                                                   1765
950 goto 130                                                                                 568
990 for j=1 to 20:next j                                                                     1079
1000 if ri%=1 goto 1050                                                                      1058
1005 poke f,60:poke f-1,67:poke f+1,32:f=f-x:c=c-x                                           4101
1010 if c<2 then poke f+1,32:poke f,32:poke f-1,32:ri%=0:if rnd (5)>.5 then ri%=1            5357
1020 if c<2 then c=40:f=32807+40*int (20*rnd (5)):x=int (7*rnd (5)+3)/10:g=f-39              5797
1030 return                                                                                  142
1050 poke g,62:poke g+1,67:poke g-1,32:g=g+x:c=c-x                                           4454
1060 if c<2 then poke g-1,32:poke g,32:poke g+1,32:ri%=0:if rnd (5)>.5 then ri%=1            5597
1070 if c<2 then c=40:f=32807+40*int (20*rnd (5)):x=int (7*rnd (5)+3)/10:g=f-39              5797
1080 return                                                                                  142
1100 poke s-i%,42:t%=t%+1:for j=1 to 70:next j                                               2647
1110 for j=70 to 200 step 10:uf=j:gosub 2:next                                               2634
1120 poke s-i%-1,42:poke s-i%+1,42:poke s-i%-40,42:poke s-i%+40,42                           4928
1125 for j=70 to 200 step 10:uf=j:gosub 2:next                                               2634
1126 for j=200 to 70 step -10:uf=j:gosub 2:next j                                            2434
1127 uf=200:gosub 2                                                                          971
1130 poke s-i%,32                                                                            1009
1140 poke s-i%-1,32:poke s-i%+1,32:poke s-i%-40,32:poke s-i%+40,32                           5038
1160 for j=i% to -120 step -40:poke s-j-40,58:poke s-j,22:uf=200+j/20:gosub 2                4542
1170 if peek (s-j+40)<>32 goto 1380                                                          2244
1175 poke s-j+40,93                                                                          1070
1180 for q=-1 to 2:uf=200+abs (25*q):gosub 2:next                                            2791
1200 next                                                                                    130
1380 uf=15:gosub 2                                                                           842
1390 for q=22 to 1 step -1:poke s-j-40*q,32:uf=255-4*q:gosub 2:for xx=1 to 2:next xx,q        6234
1400 goto 100                                                                                556
1500 for j=1 to 2000:next j                                                                  1052
1502 if m(pl%)=0 then m(pl%)=r%                                                              2204
1503 r%(pl%)=r%                                                                              1125
1507 m(pl%)=.1*int ((m(pl%)*(z%-1)+r%(pl%))/z%*10)                                           3944
1510 print "[clr,down,3right,cbm a,13shift *,cbm r,9shift *,cbm r,9shift *,cbm s]"          3482
1520 print "[3right,shift -,2spaces]spieler [4spaces,shift -,3spaces]rate[2spaces,shift -,2spaces]mittel [shift -]"   3757
1530 print "[3right,cbm q,13shift *,shift +,9shift *,shift +,9shift *,cbm w]"             3515
1540 for i=1 to x%                                                                           772
1550 print "[3right,shift -,13spaces,shift -,9spaces,shift -,9spaces,shift -]"             2651
1560 next i                                                                                  276
1570 print "[3right,cbm z,13shift *,cbm e,9shift *,cbm e,9shift *,cbm x]"                   3434
1580 for i=1 to x%:print "[up]";:next :print "[up]";   2035
1590 for i=1 to x%                                                                           772
1600 print tab(5);n$(i);:if i=pl% then print tab(15);"*";   3279
1610 print tab(19);r%(i)"%";tab(28);m(i)"%"                                                  3330
1620 next i                                                                                  276
1630 if pl%=x% then z%=z%+1:print :print "[7spaces,rvs]es folgt spiel nummer"z%              4566
1650 p%=pl%+1:if p%>x% then p%=1                                                             2170
1660 print "[2down,7spaces]jetzt ist "n$(p%)" dran !!!                                     2756
1670 print :goto 40                                                                          471
7400 sm=54272:hf=sm+1:vo=sm+24:sz=sm+4:for i=sm to sm+24:poke i,0:next                       5561
7410 poke sm+5,8:poke sm+5,8:return                                                          1570
63030 for i=50000 to 50035:read a:poke i,a:next                                              2626
63040 gosub 7400                                                                             591
63050 sys 50000:print chr$ (147);chr$ (5)                                                    2471
63055 print chr$ (142);"[yel]"                                                               1192
63060 goto 5                                                                                 243
63340 data 169,0,141,32,208,141,33,208,141,0,4,141,1,4                                       5109
63350 data 141,2,4,169,149,141,0,221,169,5,141,24                                            4086
63360 data 208,169,128,141,136,2,133,56,96,96                                                4236
63500 rem c64-version by softteam
```

Mit der „Flak" im Kampf gegen Feindflugzeuge

WWW.HOMECOMPUTERWORLD.COM

Gimmy's entführen in NIM

System: C64
Typ: Spiel
Sprache: Basic

NIM ist ein Denkspiel. In 7 Reihen sind 28 Gimmy's aufgeteilt. Wer den letzten Gimmy vom Feld nimmt, hat gewonnen.

Gleich zu Beginn printet der Computer 28 Gimmy's. In der ersten Reihe sind 7, in der zweiten Reihe 6, in der dritten Reihe 5, etcetera. . . . und in der siebten Reihe ein Gimmy. Je Durchgang muß man mindestens einen Gimmy nehmen. Maximal sind so viele Gimmy's möglich, wie in der gewählten Reihe vorhanden sind. Beispiel: In Reihe 1 sind 7 Gimmy's. Davon darf man zwischen 1 und 7 nehmen. Gewinner ist, wer den letzten Gimmy aus dem Spielfeld nimmt.

softteam

```
  1 sm=54272:hf=sm+1:vo=sm+24:sz=sm+4:for i=sm to sm+24:poke i,0:next                5561
  2 poke sm+5,8:poke sm+5,8:gosub 63000:goto 10                                     2502
  3 rem idee: andreas dripke
  4 rem (c) 1980 pet-version
  5 rem c64-version: softteam
  6 rem (c) 1986 c64-version
  8 poke vo,3:poke sz,33:poke hf,q:poke sz,32:return                                3136
 10 rem
 20 rem
 30 rem
 40 tp=198                                                                           737
 50 dim f%(8,8),fl%(7),fs%(3)                                                       2180
 60 q$="[home,14down]":x$="[38spaces]"                                              2855
 80 rem
 85 rem
 99 rem --------------------------------
100 gosub 12000:rem erklaerung                                                      719
110 gosub 1100:rem alle gimmys da!                                                  638
120 gosub 1000:rem gimmys zeichnen                                                  634
130 gosub 20000:s=1:rem c64 gewonnen?                                              1390
135 if f=1 then gosub 2000:rem c64 zug                                             1457
140 gosub 6000:rem eingabe (z,w)                                                    644
150 gosub 7000:rem tabelle korrigieren                                              646
155 gosub 1000:rem gimmys zeichnen                                                  634
160 gosub 20000:s=0:rem sie gewonnen?                                              1388
165 gosub 3000:rem ihr zug                                                          638
170 gosub 8000:f=1:rem c64 denkt                                                   1047
180 gosub 7000:rem tabelle korrigieren                                              646
190 goto 120                                                                        564
999 rem --------------------------------
1000 print "[clr,down]"                                                            469
1010 for j=1 to 7                                                                   920
1015 print j;"[left].[up]";                                                        1144
1020 t=4                                                                            393
1030 for i=1 to 7                                                                   918
1035 if f%(j,i)=0 then i=7:goto 1060                                               2626
1040 gosub 10000                                                                    569
1050 t=t+5                                                                          434
1060 next i                                                                         276
1070 print "[3down]"                                                              510
1080 next j                                                                         278
1085 rem
1090 gg=0:for j=1 to 7:for i=1 to 8-j:if f%(j,i)=1 then gg=gg+1                     4736
1095 next i,j:if gg=0 then print "+++ es sind keine gimmys mehr da! +++":return    5772
1097 if gg=1 then print "↑↑↑ es ist nur noch 1 gimmy da! ↑↑↑[3spaces]":return       4195
1099 print "*** es sind noch";gg;"gimmys da! ***":return                           4307
1100 for j=1 to 7:for i=1 to 8-j:f%(j,i)=1:next i,j                                3384
1110 for j=1 to 7:fl%(j)=8-j:next j                                                2230
1120 q=51:gosub 8:return                                                           1056
2000 print q$:print tab(31);"[5up]mein zug: "                                      2920
2010 print tab(31);"zeile";z:print tab(31);"anzahl";w                              3697
2050 return                                                                         142
3000 print q$:print tab(31);"[5up]ihr zug: "                                       3067
3010 print tab(31);"zeile";z:print tab(31);"anzahl";w                              3697
3050 return                                                                         142
4000 for i=1 to 3:fs%(i)=0:next                                                    1761
4010 for j=1 to 3:for i=1 to 7                                                     1648
4015 print q$:print "[2down]":t=28:gosub 10000:print                               2497
4020 a=fl%(i):gosub 5000                                                           1041
4030 fs%(j)=fs%(j)+val ( mid$ (a$,j,1))                                            2493
4040 next i,j                                                                       534
4050 return                                                                         142
5000 b=8:if a>(b-1) then a$="***fehler***":return                                  3250
5010 a$="":for n=1 to 3:b=int (b/2+.2):a=a-b:if a<0 then a=a+b:a$=a$+"0":goto 5030 6042
5020 a$=a$+"1"                                                                      822
5030 next n:return                                                                  634
6000 cu=31:print q$:print tab(25);"[down] sie sind dran![down]":poke tp,0          3713
6005 print tab(25);"zeile [up,3spaces,down] "                                      1607
6010 print tab(cu);"[up]";mid$ ("?[cbm +]",d%+1,1)                                 2492
6020 i=0:d%=1-d%                                                                   1586
6030 get y$:z=val (y$):if (z>0) and (z<8) then 6060                                3642
6040 i=i+1:if i<10 then 6030                                                       1639
6050 goto 6010                                                                      577
6060 if fl%(z)>0 then print tab(cu)"[2up,cbm @]":print tab(cu);chr$ (18)+y$:goto 6100 4379
6070 goto 6030                                                                      593
6100 print tab(25);"[down]anzahl [up,3spaces,down]":cu=cu+1                        3271
6110 print tab(cu);"[up]";mid$ ("?[cbm +]",d%+1,1)                                 2492
6120 i=0:d%=1-d%                                                                   1586
6130 get y$:w=val (y$):if (w>0) and (w<8) and (w<=fl%(z)) then 6160                 4803
6140 i=i+1:if i<10 then 6130                                                       1671
6150 goto 6110                                                                      581
6160 print tab(cu)"[2up,cbm @]":print tab(cu);chr$ (18)+y$                         3029
6170 for i=1 to 1100:next i:return                                                 1446
7000 if w>fl%(z) then w=fl%(z)                                                     1840
7005 for i=fl%(z)+1 to fl%(z)-w+1 step -1                                          2739
7010 f%(z,i)=0                                                                      960
7020 next i                                                                         276
7030 fl%(z)=fl%(z)-w                                                               1840
7040 return                                                                         142
8000 print q$:print tab(25);"[11cbm @]":print tab(25);"[rvs]ich denke ![off]""     5006
8050 print q$:print "[2down]":t=28:gosub 10000:print                               2497
```


WWW.HOMECOMPUTERWORLD.COM

```
8100 z=1:w=0:if f1%(1)=0 then 8200                                    2045
8110 f1%(1)=f1%(1)-1:gosub 8900:w=w+1:if                              5919
     (p1=1) and (p2=1) and (p3=1) then 8999
8120 if f1%(1)>0 then 8110                                            1340
8130 f1%(1)=f1%(1)+w                                                  1589
8150 print q$:print "[2down]":t=28:gosub                              2497
     10000:print
8200 z=2:w=0:if f1%(2)=0 then 8300                                    2089
8210 f1%(2)=f1%(2)-1:gosub 8900:w=w+1:if                              5943
     (p1=1) and (p2=1) and (p3=1) then 8999
8220 if f1%(2)>0 then 8210                                            1380
8230 f1%(2)=f1%(2)+w                                                  1613
8250 print q$:print "[2down]":t=28:gosub                              2497
     10000:print
8300 z=3:w=0:if f1%(3)=0 then 8400                                    2133
8310 f1%(3)=f1%(3)-1:gosub 8900:w=w+1:if                              5967
     (p1=1) and (p2=1) and (p3=1) then 8999
8320 if f1%(3)>0 then 8310                                            1420
8330 f1%(3)=f1%(3)+w                                                  1637
8350 print q$:print "[2down]":t=28:gosub                              2497
     10000:print
8400 z=4:w=0:if f1%(4)=0 then 8500                                    2177
8410 f1%(4)=f1%(4)-1:gosub 8900:w=w+1:if                              5991
     (p1=1) and (p2=1) and (p3=1) then 8999
8420 if f1%(4)>0 then 8410                                            1460
8430 f1%(4)=f1%(4)+w                                                  1661
8450 print q$:print "[2down]":t=28:gosub                              2497
     10000:print
8500 z=5:w=0:if f1%(5)=0 then 8600                                    2221
8510 f1%(5)=f1%(5)-1:gosub 8900:w=w+1:if                              6015
     (p1=1) and (p2=1) and (p3=1) then 8999
8520 if f1%(5)>0 then 8510                                            1500
8530 f1%(5)=f1%(5)+w                                                  1685
8550 print q$:print "[2down]":t=28:gosub                              2497
     10000:print
8600 z=6:w=0:if f1%(6)=0 then 8700                                    2265
8610 f1%(6)=f1%(6)-1:gosub 8900:w=w+1:if                              6039
     (p1=1) and (p2=1) and (p3=1) then 8999
8620 if f1%(6)>0 then 8610                                            1540
8630 f1%(6)=f1%(6)+w                                                  1709
8650 print q$:print "[2down]":t=28:gosub                              2497
     10000:print
8700 z=7:w=0:if f1%(7)=0 then 8800                                    2309
8710 f1%(7)=f1%(7)-1:gosub 8900:w=w+1:if                              6063
     (p1=1) and (p2=1) and (p3=1) then 8999
8720 if f1%(7)>0 then 8710                                            1580
8730 f1%(7)=f1%(7)+w                                                  1733
8750 print q$:print "[2down]":t=28:gosub                              2497
     10000:print
8800 print q$:print "[8down,3spaces]sie                               4116
      spielen wirklich gut![11spaces]"
8805 for i=1 to 2500:next i                                           1107
8810 z=1:if f1%(z)>0 then f1%(z)=f1%(z)-1:                            3895
     w=1:return
8820 z=2:if f1%(z)>0 then f1%(z)=f1%(z)-1:                            3899
     w=1:return
8830 z=3:if f1%(z)>0 then f1%(z)=f1%(z)-1:                            3903
     w=1:return
8840 z=4:if f1%(z)>0 then f1%(z)=f1%(z)-1:                            3907
     w=1:return
8850 z=5:if f1%(z)>0 then f1%(z)=f1%(z)-1:                            3911
     w=1:return
8860 z=6:if f1%(z)>0 then f1%(z)=f1%(z)-1:                            3915
     w=1:return
8870 z=7:if f1%(z)>0 then f1%(z)=f1%(z)-1:                            3919
     w=1:return
8880 print "***fehler***":stop                                        1857
8900 gosub 4000:p1=0:p2=0:p3=0                                        2123
8905 print q$:print "[2down]":t=28:gosub                              2497
     10000:print
8910 if (fs%(1)/2)-int (fs%(1)/2)=0 then                              2896
     p1=1
8920 if (fs%(2)/2)-int (fs%(2)/2)=0 then                              2972
     p2=1
8930 if (fs%(3)/2)-int (fs%(3)/2)=0 then                              3048
     p3=1
8950 return                                                           142
8999 f1%(z)=f1%(z)+w:t=23:gosub 9000:return                           3309

9000 print q$:print "[down]"                                           974
9010 print tab(t);"[5spaces]*[2spaces]*"                              1115
9012 print tab(t);"[4spaces]* ** *"                                   1208
9014 print tab(t);"[3spaces]* M[2shift -]N                            1716
      *"
9016 print tab(t);"[4spaces]*[shift *]oo                              1499
     [shift *]*"
9018 print tab(t);"[6spaces]JK"                                       1303
9030 for i=240 to 117 step -5:q=i:gosub 8:                            4086
     for n=1 to 50:next n,i:q=0:gosub 8:
     return
10000 tt=16*rnd (1)+1                                                 1535
10010 on tt gosub 10100,10150,10200,10250,                            5440
      10300,10350,10400,10450,10500,10550
10020 if tt>10 then on tt-10 gosub 10600,                             4657
      10650,10700,10750,10800,10850
10050 q=239*rnd (ti)+15:gosub 8:print "[3up]                          2809
      ";:return
10100 print tab(t);"M##N"                                             1213
10110 print tab(t);" oo "                                             1079
10120 print tab(t);" JK "                                             1137
10130 return                                                          142
10150 print tab(t);"M[2cbm f] "                                       1192
10160 print tab(t);" WWM"                                             1374
10170 print tab(t);" JK "                                             1137
10180 return                                                          142
10200 print tab(t);"[shift *]IU[shift *]"                             1379
10210 print tab(t);" QQ "                                             1274
10220 print tab(t);" JK "                                             1137
10230 return                                                          142
10250 print tab(t);"UIUI"                                             1466
10260 print tab(t);" SS "                                             1277
10270 print tab(t);" [cbm z,cbm x] "                                  1236
10280 return                                                          142
10300 print tab(t);"M[2cbm f]N"                                       1285
10310 print tab(t);" WW[shift-space]"                                 1284
10320 print tab(t);" JK "                                             1137
10330 return                                                          142
10350 print tab(t);" [2cbm p] "                                       1223
10360 print tab(t);"NWWM"                                             1545
10370 print tab(t);" JK "                                             1137
10380 return                                                          142
10400 print tab(t);"M##N"                                             1213
10410 print tab(t);" $$ "                                             887
10420 print tab(t);" JK "                                             1137
10430 return                                                          142
10450 print tab(t);" [2cbm f]N"                                       1178
10460 print tab(t);"NWW[shift-space]"                                 1455
10470 print tab(t);" JK "                                             1137
10480 return                                                          142
10500 print tab(t);" [2cbm f] "                                       1085
10510 print tab(t);"NooM"                                             1341
10520 print tab(t);" JK "                                             1137
10530 return                                                          142
10550 print tab(t);"J[2cbm o]K"                                       1495
10560 print tab(t);"[cbm c]oo[cbm v]"                                 1179
10570 print tab(t);" JK "                                             1137
10580 return                                                          142
10600 print tab(t);" ww "                                             1091
10610 print tab(t);"NooM"                                             1341
10620 print tab(t);" JK "                                             1137
10630 return                                                          142
10650 print tab(t);"MwwN"                                             1291
10660 print tab(t);" oo "                                             1079
10670 print tab(t);" JK "                                             1137
10680 return                                                          142
10700 print tab(t);"M[2cbm r] "                                       1207
10710 print tab(t);" ooM"                                             1170
10720 print tab(t);" JK "                                             1137
10730 return                                                          142
10750 print tab(t);" [2cbm r]N"                                       1193
10760 print tab(t);"Noo "                                             1250
10770 print tab(t);" JK "                                             1137
10780 return                                                          142
10800 print tab(t);" [2cbm r] "                                       1100
10810 print tab(t);"[shift *]oo[shift *]"                             1184
10820 print tab(t);" JK "                                             1137
10830 return                                                          142
10850 print tab(t);" [2cbm f] "                                       1085
10860 print tab(t);"[shift *]WW[shift *]"                             1388
10870 print tab(t);" JK "                                             1137
10880 return                                                          142
12000 rem
12005 print "[clr]":ra=102:gosub 50000                                1838
12010 print "[5right,4spaces,9cbm @]":print                          4627
      "[5right]*** [rvs]gimmy-nim[off] ***"
12012 rem
12015 for t=5 to 30 step 5:gosub 10000:next                           1617
      t
12020 print "[4down,2right]abwechselnd                               4420
       nehmen sie und der com-"
12030 print "[down,2right]puter aus einer                            6346
       reihe eine be-":print tab(8);" [15cbm t]"
```


WWW.HOMECOMPUTERWORLD.COM

```
12040 print "[2right]liebige anzahl gimmys      3716
       weg. jeder"
12050 print "[down,2right]muss mindestens      4311
       einen gimmy weg-"
12060 print "[down,2right]nehmen, wenn er am   2965
       zuge ist."
12070 print "[down,2right]wer den letzten      4425
       gimmy wegnimmt, der"
12080 print "[down,2right]hat gewonnen![up]"   2406
      ;
12100 rem
12150 gosub 55000:print "[2up]";:return         1512
20000 if gg>0 then return                        861
20010 if s=1 then 20200                         1017
20100 print "[home,3down,5right,26cbm @]"       3374
20105 print "[5right,rvs]leider haben sie       3582
       verloren!"
20110 print "[down,5right]bekommen sie          4203
       deswegen aber keine"
20120 print "[down,5right]depressionen,         4783
       sondern versuchen"
20130 print "[down,5right]sie es doch noch      3464
       einmal![2down]"
20140 for i=58 to 240 step 5:q=i:gosub 8:for    4267
      n=1 to 100:next n,i:goto 20300
20200 print "[home,3down,5right,26cbm @]"       3374
20210 print "[5right,rvs]gratuliere, sie        4123
       haben gewonnen!"
20220 print "[down,5right]aber geben sie mir    3945
       doch noch"
20230 print "[down,5right]eine chance, auch     4360
       mal zu gewinnen!"
20240 for i=240 to 58 step -5:q=i:gosub 8:      4488
      for n=1 to 100:next n,i:goto 20300
20300 q=0:gosub 8:gosub 55000                    1629
20500 run                                         138
50000 for n=32768 to 32807:poke n,ra:poke n+    3935
      960,ra:next
50010 for n=32808 to 33688 step 40:poke n,      4017
      ra:poke n+39,ra:next :return
55000 poke tp,0:print tab(31);"[6cbm @]":       5229
      print tab(30);"<[rvs]return[off]>":
      wait tp,1:return
63000 for i=50000 to 50035:read a:poke i,a:     2626
      next
63010 data 169,0,141,32,208,141,33,208,141,     5109
      0,4,141,1,4
63020 data 141,2,4,169,149,141,0,221,169,5,     5265
      141,24,208,169,128
63030 data 141,136,2,133,56,96,96               2733
63040 sys 50000:print "[clr]";chr$ (5);chr$     3163
      (142);chr$ (8)
63060 return                                     142
```

NIM: Wenn der C64 Grimassen macht

5 in einer Reihe

System: C64
Typ: Denkspiel
Sprache: Basic

5 Steine in einer Reihe bedeuten den Sieg bei diesem Denkspiel. 30 Züge hat man Zeit, um den elektronischen Gegner zu besiegen.

Das Spielfeld ist 9 mal 9 Quadrate groß. Zuerst ist der Y-Achsenwert, dann der X-Achsenwert einzugeben. „Gomoku" setzt seine Steine nach kurzer Bedenkzeit. 5 Steine sollen in einer Reihe stehen. Dabei ist es unwichtig, ob die Reihe senkrecht oder diagonal liegt. Die ersten 20 Züge bestehen darin, Steine auf beliebige, freie Felder zu setzen. Gelingt es in dieser Zeit keinem Kontrahenten, 5 Steine in eine Reihe zu setzen, folgt Teil 2 des Spiels: 10 Züge lang springt man mit bereits gesetzten Steinen von bisherigen Standorten auf beliebige, freie Felder. Bleiben auch in Teil 2 des Spieles beide Parteien erfolglos, endet der Spieldurchgang unentschieden. Nach jedem Spiel zeigt „Gomoku" die Punktestände an. Danach ist einzugeben, ob man weiterspielen oder aufhören will. „J" eingeben und es geht weiter in die nächste Runde.

softteam

```
1 goto 63000                                     587
2 poke vo,3:poke sz,33:poke hf,5:poke           3080
  sz,32:return
3 for mz=1 to 255 step 4:poke vo,7:poke         5363
  sz,33:poke hf,mz:poke sz,32:next mz:
  return
4 for mz=255 to 1 step -4:poke vo,7:poke        5326
  sz,33:poke hf,mz:poke sz,32:next mz:
  return
5 data 169,0,141,32,208,141,33,208,141,         5109
  0,4,141,1,4
6 data 141,2,4,169,149,141,0,221,169,5,         5265
  141,24,208,169,128
7 data 141,136,2,133,56,96,96                   2733
9 dim v%(81),n1%(140),n2%(140),pt%(3,           6543
  25),mt%(50),zt%(99),i$(26):m1=20:m2=10
10 a=13:e=2000:bl$="[up,13spaces]":w=250:       3887
   pn=32904
40 for i=1 to 73 step 9:for j=0 to 4:k=k+       4980
   1:n1%(k)=i+j:n2%(k)=1:next :next
45 for i=0 to 8:for j=1 to 37 step 9:k=k+       5257
   1:n1%(k)=i+j:n2%(k)=9:next :next
50 for i=1 to 37 step 9:for j=0 to 4:k=k+       5030
   1:n1%(k)=i+j:n2%(k)=10:next :next
55 for i=5 to 41 step 9:for j=0 to 4:k=k+       4802
   1:n1%(k)=i+j:n2%(k)=8:next :next
75 data 1,3,5,10,0,2,0,0,0,0,0,4,0,0,0,0,       4516
   0,6,0,0,0,0,0,0,0,0
76 data 2,4,5,10,0,1,0,0,0,0,0,3,0,0,0,0,       4509
   0,6,0,0,0,0,0,0,0,0
77 data 0,0,0,0,0,0,0,0,5,5,0,0,0,0,0,0,        4537
   0,5,0,0,0,0,0,5,5
```


WWW.HOMECOMPUTERWORLD.COM

```
 80 for i=1 to 3:for j=1 to 25:read pt%(i,    4834
    j):next :next :if r=0 then r=1:goto
    6000
 90 for i=1 to 81:v%(i)=0:next :f=0:mk=1:     6698
    p=int ( rnd (1)*2+1):mc=0:s1=0:wi=0:
    ff=0:ph=0
100 print "[clr]";tab(a)"[7cbm +]***go-       4255
    moku***[6cbm +]
110 print tab(a);:for i=1 to 13:print "       2797
    [2cbm +]";:next :print
120 print tab(a)"[3cbm +] [cbm a,shift *,     2896
    cbm r,shift *,cbm r,shift *,cbm r,
    shift *,cbm r,shift *,cbm r,shift *,
    cbm r,shift *,cbm r,shift *,cbm r,
    shift *,cbm s,3cbm +]
130 for i=1 to 9:print tab(a)"[3cbm +]";      5628
    right$ ( str$ (i),1);"[shift -,
    shift-space,shift -,shift-space,
    shift -,shift-space,shift -,
    shift-space,shift -,shift-space,
    shift -,shift-space,shift -,
    shift-space,shift -,shift-space,
    shift -,shift-space,shift -,3cbm +]"
140 print tab(a)"[3cbm +] [cbm q,shift *,     3606
    shift +,shift *,shift +,shift *,
    shift +,shift *,shift +,shift *,
    shift +,shift *,shift +,shift *,
    shift +,shift *,shift +,shift *,cbm w,
    3cbm +]
150 next :print tab(a)"[up,3cbm +] [cbm z,    3377
    shift *,cbm e,shift *,cbm e,shift *,
    cbm e,shift *,cbm e,shift *,cbm e,
    shift *,cbm e,shift *,cbm e,shift *,
    cbm e,shift *,cbm x,3cbm +]
155 print tab(a)"[3cbm +,2spaces]1 2 3 4 5    2841
     6 7 8 9 [3cbm +]
160 print tab(a);:for i=1 to 13:print "       3978
    [2cbm +]";:next :print :if f=0 then f=
    1:goto 160
180 if a1 then a1=0:return                     867
190 print "[2up]zugnr. 1":gosub 2             1682
195 poke ph,oi:ph=0:print "[home,down]":      3428
    for i=1 to 7:print :print bl$:next
200 dx=99:z2=6:rh=0:ch=0:xh=0:f1=0:f5=0:      6655
    print "[home,down,rvs]ihr zug[2spaces]
    :":if mk>m1 goto 300
205 gosub 1000:if f5 goto 195                 1447
210 gosub 1200:on f5+1 goto 400,195           1669
300 print "[rvs]von :":f1=1:gosub 1000:if     3359
    f5 goto 195
310 if v%(x)<>1 goto 390                      1622
360 poke pk,oi+128:ph=pk:rh=r:ch=c:xh=x       3721
385 print "[rvs]zu:":gosub 1000:on f5+1       2631
    goto 396,195
390 if v%(x)=0 then w1$="[rvs]das feld        4954
    [3spaces,off]":w2$="[rvs]ist frei.
    [off]":goto 393
392 w1$="[rvs]ich setze[off]":w2$="[rvs]      3318
    auf's feld.[off]"
393 print w1$:print w2$:for i=1 to e:next     2030
    :goto 195
395 f2=0:gosub 1200:if f2 goto 394            1992
396 gosub 1200:if f5 goto 195                 1455
397 v%(xh)=0:poke pn+ch+ch+(rh-1)*80,32       3337
400 v%(x)=1:poke pk,oi:mc=mc+1:if mc=2        3730
    then gosub 8000
410 print "[home,down]ihr zug[2spaces]:":     4058
    if mk=<m1 then print "[10down]";:goto
    425
415 print "von:":print "[2down]nach:":        2982
    print "[6down]";
425 print "[rvs]c64'er zug:":if mk>m1 goto    2815
    900
430 print :print bl$:print :print bl$:        2068
    print "[2up]";
500 if mk>1 goto 503                           699
501 x=42:if v%(41)=0 then x=41                2330
502 goto 710                                   572
503 hv=0:k=0:for i=1 to 140:ct=0              1943
505 for j=n1%(i) to n2%(i)*4+n1%(i) step      5766
    n2%(i):ct=ct+v%(j):next :if ct=0 goto
    550
515 if ct=5 goto 3000                         1007
525 if ct=24 goto 2000                        1131
530 v1=pt%(p,ct):if v1>hv then hv=v1:k=1:     4277
    mt%(k)=i:goto 550
545 if v1=hv then k=k+1:mt%(k)=i              3026
550 next                                       130
600 l=0:for i=1 to k:ii=mt%(i):for j=         5457
    n1%(ii) to n2%(ii)*4+n1%(ii) step
    n2%(ii)
610 if v%(j)=0 then l=l+1:zt%(l)=j            2729
620 next :next                                 256
630 hv=0:ii=0:for i=1 to l:ct=0:for j=1 to    2503
    l
640 if zt%(i)=zt%(j) then ct=ct+1             2409
650 next                                       130
655 if ct>hv then hv=ct:ii=1:mt%(ii)=         3755
    zt%(i):goto 670
660 if ct=hv then ii=ii+1:mt%(ii)=zt%(i)      3367
670 next                                       130
680 x=mt%( int ( rnd (1)*ii+1))               1861
690 if f goto 970                              639
710 v%(x)=6:xc=0                              1469
720 if x>9 then xc=xc+1:x=x-9:goto 720        2853
730 poke pn+x+x+xc*80,pi:if mk<=m1 then       4229
    print "[up]c64'er zug:
750 print "y-wert";xc+1:print "x-wert";x      2738
752 gosub 3                                    243
755 mc=mc+1:if mc=2 then gosub 8000           1958
760 on ff+1 goto 195,990                      1268
800 k=0:for i=1 to 140:ct=0                   1990
810 for j=n1%(i) to n2%(i)*4+n1%(i) step      5864
    n2%(i):ct=ct+v%(j):next :if ct=0 goto
    835
820 if ct=5 goto 3000                         1007
830 if pt%(3,ct)=5 then k=k+1:mt%(k)=i        3119
835 next :if k=0 goto 870                      708
840 l=0:for i=1 to k:n=mt%(i):for j=          5543
    n1%(n) to n2%(n)*4+n1%(n) step n2%(n)
850 if v%(j)=6 then l=l+1:zt%(l)=j            2732
860 next :next                                 256
870 x=int ( rnd (ti)*81+1)                    1618
875 for i=x to x+81:x=x+1:if x>81 then x=1     2817
880 if v%(x)<>6 goto 899                      1791
890 for j=1 to l:if x=zt%(j) goto 899         2681
895 next j:goto 910                            783
899 next i:x=99:goto 910                      1467
900 print "[rvs]von :":for i=1 to 7:print      3743
    :print bl$:next :print "[7up]";:goto
    800
910 if x<>99 goto 940                         1250
920 for i=1 to 81:if v%(i)=6 then x=i:goto     2691
    940
930 next                                       130
940 v%(x)=0:xc=0:xh=x                         1709
950 if x>9 then xc=xc+1:x=x-9:goto 950        2981
960 poke pn+x+x+xc*80,pi+128:print "x- :";     6171
    xc+1:print "y-wert:";x:print "[rvs]zu:
    "
965 x1=x:r2=xc:ff=1:goto 500                  2407
970 if x<>xh goto 710                         1202
975 for i=xh to xh+81:x=x+1:if x>81 then       3133
    x=1
980 if v%(x)=0 goto 710                       1462
985 next                                       130
990 print "[home]":for i=1 to 11:print :      3183
    next :print "c64'er zug:"
995 print "von :[2down]":print "zu:":poke     4769
    pn+x1+x1+r2*80,32:ph=0:goto 195
1000 d=0:print "y- :"                         1311
1005 get a$:if a$="" goto 1005                1329
1007 if peek (515)=27 goto 1150               1481
1010 print "[up]";tab(5)a$:r=val (a$):if r<   3271
    1 or r>9 goto 1130
1030 print "x-wert:"                          1024
1040 get a$:if a$="" goto 1040                1392
1042 gosub 2                                   241
1043 if peek (515)=27 goto 1150               1481
1045 print "[up]";tab(8)a$                     902
1080 c=val (a$):if c<1 or c>9 goto 1130       2093
1100 x=((r-1)*9)+c:pk=32904+c+c+(r-1)*80:if   5928
    f1 then f1=0:return
1105 if r=rh and c=ch goto 1150               1908
1110 if v%(x)=0 then return                   1086
1120 print "[rvs]besetzt ![off]":gosub 4:      1976
    goto 1140
1130 print "[rvs]nur 1 bis 9![off]":gosub 4    1632
1140 for i=1 to e:next                        1063
1150 f5=1:return                               887
1200 poke pn+c+c+((r-1)*80),191               1924
1210 for i=1 to w:get a$:if a$="" goto 1240    2541
```


WWW.HOMECOMPUTERWORLD.COM

```
1220 poke pn+c+c+((r-1)*80),32                    1790
1230 f5=1:return                                   887
1240 next :return                                  304
2000 w$="c64 gewinnt!":pw=pw+1:wi=pi:goto         3440
     4000
3000 w$="sie gewinnen!":ow=ow+1:wi=oi              3829
4000 for j=n1%(i) to n2%(i)*4+n1%(i) step         4180
     n2%(i):x=j:xc=0
4010 if x>9 then xc=xc+1:x=x-9:goto 4010           2829
4015 if v%(j)=0 and wi=pi then print "x-           4123
     wert";xc+1:print "y-wert";x
4020 poke pn+x+xc*80,wi+128:next :if mk>           5771
     m1 then poke pn+ch+ch+rh*80,32
4035 if ff then poke pn+x1+x1+r2*80,32             2249
4037 bs=10:if wi=oi then bs=19                     2009
4040 print "[home]":for i=1 to bs:print :         4505
     print bl$:next :if wi=pi then print "
     [down]c64'er zug:"
4043 if mk>m1 and wi=pi then print "von :":       3153
     print "[2down]zu:"
4050 print "[home]":print "[rvs]";w$:print       2159
     "[rvs]punkte:[off]
4060 print no$;":";ow:print pn$;":";pw:            3584
     print "remis:";tg
4070 print "[down]wollen sie ":print              3858
     "nochmal":print "spielen?"
4075 gosub 3:gosub 4:gosub 3:gosub 4               1266
4080 get a$:if a$="" goto 4080                     1289
4090 if a$="j" goto 90                             1154
4095 print "[clr]":end                             495
6000 print "[clr,down]willkommen zu               5401
     [5spaces,rvs] 'go-moku' [off].":print
     "[down]mein name ist pit."
6010 print "[down]geben sie ihren namen           4982
      ein.":print "danach [rvs]return[off]
      ! [cbm @,3left]";:input a$
6020 if a$="[cbm @]" then print "[3up]";:          2427
     goto 6010
6030 no$=a$:oi=asc (a$)-64:pn$="pit":pi=16:        6352
     if oi=16 then pi=3:pn$="computer"
6050 print "[down]hallo[2spaces]";no$;:          2106
     print " !"
6055 print "[down]ihr zug wird durch den"        2216
6060 print "buchstaben[2spaces]'";left$          5274
     (no$,1);"' fuer ";no$;" gezeigt."
6070 print "[down]meine zuege sehen aus wie       5020
      ":print "buchstabe[5spaces]'";left$
     (pn$,1);
6080 print "' fuer ";pn$;".                      1865
6090 print "[down]anleitung[2spaces]?"           1508
6095 print "'j' oder 'n'."                       1852
6097 get a$:if a$="" goto 6097                     1530
7000 if a$="n" goto 90                             1282
7010 a1=1:gosub 100:print "[home,down]das          3252
      ist":print "das feld."
7020 print "[down]man gewinnt":print "durch       4781
     5 zu-":print "sammenhaeng-"
7030 print "ende felder,":print "egal in          5714
      wel-":print "che richtung":print "sie
      gehen."
7085 print "[down,rvs]return[off] ":print         3409
     "blaettert.":poke 198,0:wait 198,1
7100 print "[clr,down]wir haben ";m1;             6120
     "steine.":print "man muss";m1;"steine
     setzen."
7120 print "das feld muss unbesetzt sein."       2828
7130 print "nachdem alle steine":print "          6268
     [down]gesetzt sind":print "muss man
     mit den figuren"
7140 print "auf dem feld herumfahren."           2312
7150 print "das ziel der fuenf steine in         3024
      einer"
7160 print "reihe ist gleich.":print "           5415
     [down]gewinnt keiner in";m1+m2;
     "zuegen"
7170 print "ist das spiel remis."               2100
7180 print "[down]geben sie weiter, dann         4188
     [2spaces,rvs]return[off] druecken."
7280 print "[down,3spaces,rvs]return[off]        4135
      startet das spiel.":poke 198,0:wait
     198,1
7290 poke 198,0:goto 90                            1361
7400 sm=54272:hf=sm+1:vo=sm+24:sz=sm+4:for         5561
     i=sm to sm+24:poke i,0:next
7410 poke sm+5,8:poke sm+5,8:return                1570
8000 print "[home]":for i=1 to 21:print :         1576
     next
8005 mk=mk+1:if mk>m1+m2 then w$="game             4876
      tied.":tg=tg+1:goto 4035
8010 mc=0:if mk>m1 then print "[rvs]";             2083
8020 print tab(6)mk:return                         1043
63000 for i=50000 to 50035:read a:poke i,a:        2626
     next
63040 gosub 7400:sys 50000:print "[clr]";          4394
     chr$ (5);chr$ (142);chr$ (8):goto 9
```

5 Steine senkrecht oder diagonal hintereinander bedeuten den Sieg

Umleitungen im Computer

System: C64
Typ: Spiel
Sprache: Basic

Einen Sprungball muß man ins Ziel lenken, um zu gewinnen. Mit den Tasten „N" und „M" baut man die dafür notwendigen Ecken, Gänge und Winkel.

Schnell reagieren und die Mauern überlegt setzen: Nur so gewinnt man bei „Trapskrip". Nach Start mit RUN gibt man die Spieleranzahl (1 bis 9), Namen der Teilnehmer und das Schwierigkeitslevel ein. Danach zeichnet der 64'er das Spielfeld mit zwei Bällen: Geschoß und Ziel. Das Geschoß fliegt mit hoher Geschwindigkeit über's Spielfeld. Mit Taste „N" setzt man ein „/"-Winkel direkt vor das Geschoß und lenkt es damit gezielt auf das Ziel. Taste „M" setzt den „\"-Winkel. Nach jedem Durchgang zeigt „Trapskrip" die zuletzt benötigte Zeit, um den Zielball zu treffen und die bisherige Durchschnittszeit an. „E" beendet das Spiel. *F. Bastian*

```
1 goto 63030                                       635
2 poke vo,7:poke sz,33:poke hf,5:poke             3144
  sz,32:return
3 for mz=1 to 255 step 4:poke vo,7:poke           5363
  sz,33:poke hf,mz:poke sz,32:next mz:
  return
10 px=1:dim n%(9),pn$(9),ps%(9),pm%(9)            3350
20 print "[clr,down,9spaces]*** trapskrip         2374
    ***"
30 print "[2down,2spaces]*** treffen sie          3360
    das ziel[2spaces]-- w ***"
40 print "[down,2spaces](tasten 'n' und           2594
    'm' !!!)"
45 print "[down]c64 - version von f.              2944
    bastian"
50 print "[down,2spaces]anzahl der                4345
    spieler (1-9) ? ";:gosub 830:py=x
```


WWW.HOMECOMPUTERWORLD.COM

```
 60 print "[down,2spaces]'return' nach      4163
     nameneingabe druecken":gosub 890
 70 print :print :print tab(10)"'return'     2941
     druecken";
 80 get a$:if a$<>chr$ (13) then 80          2108
 90 pl=pl+1:if pl>py then pl=1               2133
100 b$="[rvs,40spaces]":v$="[23down]"        3789
110 ts=ti:tx=ts+6000:pc=1:pk=33268:d=1:rem   3621
     **  draw border *
120 print "[clr]"b$"[home,rvs,3spaces]       3768
    spieler"pl"[2spaces]-[2spaces]" left$
    (pn$(pl),10) spc(12);
130 for i=1 to 22:print "[rvs] [38right] "   6297
    ;
140 next :print "[rvs]druecke ! fuer ende    5718
     --[3spaces]level[4spaces]="(20-
    n%(pl))/2"[left]" left$ (b$,4);
150 t=int ( rnd (1)*900+32768):if (33248<    5389
    t) and (t<33288) then 150
160 td=1:nn=n%(pl)                           1834
170 if peek (t)<>32 then 150                 1616
180 poke t,87:rem start                       874
190 rem
200 for ii=1 to nn:if peek (pe)=81 then      2194
    poke pe,32
210 pe=pk:get i$:if i$<>"" then 310          2107
220 if peek (pk+d)<>32 then 260              1976
230 pk=pk+d:poke pk,81:next :poke t,32:t=    3202
    t+td*d
240 if peek (t)<>32 then td=-td:t=t+td*d     2647
250 poke t,87:goto 200                       1141
260 if ti>tx then ps=0:goto 450              1810
270 if peek (pk+d)=78 then pk=pk+d:d=d-sgn    4050
    (d)*41:gosub 2:goto 200
280 if peek (pk+d)=77 then pk=pk+d:d=sgn      6165
    (d)*(( abs (d) and 1)*40+( abs (d) and
    40)/40):gosub 2:goto 190
290 d=-d:if peek (pk-d)=87 then 440          2296
295 poke vo,7:poke sz,33:poke hf,200:poke    3531
    sz,32
300 goto 200:rem   volltreffer                721
310 if i$="n" then 350                       1509
320 if i$="m" then 380                       1483
330 if i$="!" then ps=0:goto 450             1999
340 goto 220                                  566
350 o=d:pc=pc+1:poke pk,78                   1814
360 d=d-sgn (d)*41:goto 200                  1463
370 if peek (pk+d)=78 then pk=pk+d           2064
380 o=d:pc=pc+1:poke pk,77                   1812
390 if o=-1 then d=-40:rem      "M"          1637
400 if o=40 then d=1                         1062
410 if o=1 then d=40                         1140
420 if o=-40 then d=-1                       1393
430 goto 200                                  558
440 ps=1000-12*(pc-1)-int ((ti-ts)/6):if     4737
    ps<0 then ps=1
445 gosub 3                                   243
450 ps%(pl)=ps                               1010
460 pm%(pl)=int ((pm%(pl)*(px-1)+ps)/px)     3309
470 print "[clr,down,3right,rvs,cbm a,       3864
    11shift *,cbm r,5shift *,cbm r,
    7shift *,cbm r,7shift *,cbm s]"
480 print "[3right,rvs,shift -,              4391
    shift-space]spieler[3spaces,shift -]
     lev [shift -] pkte[2spaces,shift -]
     dusnt [shift -]"
490 print "[3right,rvs,cbm q,11shift *,      3623
    shift +,5shift *,shift +,7shift *,
    shift +,7shift *,cbm w]"
500 print "[3right,rvs,shift -,off,          2743
    11spaces,shift -,5spaces,shift -,
    7spaces,shift -,7spaces,rvs,shift -]"
510 for i=1 to py                             784
520 print "[3right,rvs,shift -,off] "        6201
    pn$(i)(20-n%(i))/2" [shift -] "ps%(i)
    tab(29)"[shift -] "pm%(i) tab(37)"
    [rvs,shift -]"
530 next                                      130
540 print "[3right,rvs,shift -,off,          2743
    11spaces,shift -,5spaces,shift -,
    7spaces,shift -,7spaces,rvs,shift -]"
550 print "[3right,rvs,cbm z,11shift *,      3614
    cbm e,5shift *,cbm e,7shift *,cbm e,
    7shift *,cbm x,off]"
554 print spc(9)"[2down,rvs] spielende =     2420
     'x' "
560 print spc(9)"[3up,rvs]spiel nummer # "    2388
    px
570 print "[home]";:for kk=1 to 4+pl:print    4805
    "[down]";:next :print "[4right]*":for
    kk=pl to py:print "[down]";:next
580 if pl=py then gosub 600:for i=1 to py:    4810
    ps%(i)=0:next :px=px+1:goto 70
590 print "[5down]" spc(9)"[rvs]'return'      3427
     druecken":goto 80
600 print "[6down]" spc(7)"[rvs]level         4154
     aendern (j/n) ?[off] ";:gosub 770
610 if a$="n" then return                     1286
612 if a$="x" then 1100                       1422
614 if a$<>"j" then return                    1024
620 print "[clr,3down]spieler-nr.[2spaces]    4217
    ? ";:gosub 830:xx=x:if x>py then 620
630 a$=pn$(xx):aa=10                          1501
640 if mid$ (a$,aa,1)=" " and aa>1 then       3535
    aa=aa-1:goto 640
650 a$=left$ (a$,aa)+"'s ":print "[2down]"    4848
    a$"level zur zeit ist "(20-n%(xx))/2
660 print spc(len (a$))"durchschnitt ist"     2787
    pm%(xx)
670 print spc(len (a$))"letzte punktzahl :    3271
     "ps%(xx)
680 print "[down]neues level[10spaces](1-     4689
    9) ? ";:gosub 830:n%(xx)=20-x*2
690 print :print "[down]weitere                5414
     aenderungen (j/n)? ";:gosub 770:if
    a$="j" then 620
692 if a$<>"n" then 690                       1461
700 return                                     142
710 for i=1 to 100:get a$:if a$<>"" then      2183
    750
720 next :print "[rvs] [left]";                581
730 for i=1 to 100:get a$:if a$<>"" then      2183
    750
740 next :print "[off] [left]";:goto 710       953
750 if val (a$)=0 then 710                    1349
760 print "[off]"a$:s=val (a$):return         1578
770 for i=1 to 100:get a$:if a$<>"" then      1959
    810
780 next :print "[rvs] [left]";                581
790 for i=1 to 100:get a$:if a$<>"" then      1959
    810
800 next :print "[off] [left]";:goto 770      1049
810 if a$<>"j" and a$<>"n" and a$<>"x"        3059
    then 770
820 print "[rvs] [off]"a$:return              588
830 for i=1 to 100:get a$:if a$<>"" then      2088
    870
840 next :print "[rvs] [left]";                581
850 for i=1 to 100:get a$:if a$<>"" then      2088
    870
860 next :print "[off] [left]";                589
870 if a$="0" then 1100                       1417
872 if a$<"1" or a$>"9" then 830              1841
880 print "[off] [left]"a$:x=val (a$):        1575
    return
890 print "[down]":for k=1 to py:print "      6620
    [2spaces]name des spielers "k"?
     [10cbm @,10left]";
900 gosub 930                                 588
910 print :print "[2spaces]                   6899
    schwierigkeitslevel spieler"k"(1-9) ?
     ";:gosub 830:n%(k)=20-2*x
920 next :return                              304
930 b$=""                                     391
940 for j=1 to 50:get a$:if a$<>"" then       2822
    970
950 next :print "[rvs,cbm @,left]";:for j=    2949
    1 to 50:get a$:if a$<>"" then 970
960 next :print "[off,cbm @,left]";:goto      1161
    940
970 a=asc (a$):if a=13 then print "[off,      2374
    cbm @]":goto 1020
980 if a=20 and len (b$)>1 then b$=left$      5174
    (b$,len (b$)-1):print "[off,cbm @,
    2left,cbm @,left]";:goto 940
990 if a=20 and len (b$)=1 then b$="":        4484
    print "[off,cbm @,2left,cbm @,left]";:
    goto 940
1000 if a=32 or (64<a and a<98) then print    3425
    "[off]"a$;:b$=b$+a$:if len (b$)>9 then
    1020
1010 goto 940                                 588
1020 b$=b$+"[10spaces]":pn$(k)=left$ (b$,     3335
    10)+"[shift -] ":return
1030 end :rem ende von trapskrip              244
1100 print "[clr,4down,5spaces]*** ein        3979
     anderes spiel (j/n) ? ***"
1110 get z$:if z$="j" then run                1598
1120 if z$<>"n" then 1110                     1480
1130 print chr$ (147);"ade ...":end           1857
7400 sm=54272:hf=sm+1:vo=sm+24:sz=sm+4:for    5561
    i=sm to sm+24:poke i,0:next
7410 poke sm+5,8:poke sm+5,8:return           1570
63000 data 169,0,141,32,208,141,33,208,141,   5109
    0,4,141,1,4
63010 data 141,2,4,169,149,141,0,221,169,5,   4086
    141,24
63020 data 208,169,128,141,136,2,133,56,96,   4236
    96
63030 for i=50000 to 50035:read a:poke i,a:    2626
    next
63040 gosub 7400                              591
63050 sys 50000:print chr$ (147);chr$ (6)     2473
63060 goto 10                                  427
```

„Trapskrip“: Den Bildschirm verwinkeln

WWW.HOMECOMPUTERWORLD.COM

C64 ist clever!

System: C64
Typ: Spiel
Sprache: Basic

Der Commodore 64 fordert heraus. Wer schlägt ihn im Denkspiel „Cleverness"?

Nachdem der Spieler mit einer kurzen Melodie und den Spielregeln begrüßt wurde, baut der Computer das Spielgatter auf. Dieses Gatter umschließt 8 mal 8 Quadrate. Ziel ist es, die Quadrate einzukesseln.
Schließt man ein Quadrat vollkommen an seinen Gatterrändern ein, erhält man das Spielfeld und darf den nächsten Zug machen. Den Eingabecursor steuert man mit:

R = schräg links hoch
T = senkrecht nach oben
Y = schräg rechts hoch
F = nach links
G = stehenbleiben
H = nach rechts
V = schräg links runter
B = senkrecht nach unten
N = schräg rechts runter

RETURN setzt ein Mauerstück an die augenblickliche Stelle des Eingabecursors. Es ist zu vermeiden, ein Quadrat an 3 Seiten einzukesseln, da sonst der Computer in seinem nächsten Zug die vierte Seite schließt und somit das Feld vollends einkesselt.

Zu Beginn eines Spieles sollte man in der Mitte des Feldes beginnen, Mauern zu setzen. Gewonnen hat, wer am Schluß die meisten Spielfelder sein eigen nennt. Selten kommt es am Ende eines Spiels zum Unentschieden.

softteam

```
  1 goto 63030                                                    635
  2 poke vo,3:poke sz,33:poke hf,5:poke sz,32:return             3080
  3 for mz=1 to 255 step 4:poke vo,7:poke sz,33:poke hf,mz:poke sz,32:next mz:
    return                                                        5363
  4 for mz=255 to 1 step -4:poke vo,7:poke sz,33:poke hf,mz:poke sz,32:next mz:
    return                                                        5326
 10 data 169,0,141,32,208,141,33,208,141,0,4,141,1,4            5109
 20 data 141,2,4,169,149,141,0,221,169,5,141,24                 4086
 30 data 208,169,128,141,136,2,133,56,96,96                     4236
 50 gosub 60000                                                  579
100 crt=32768                                                    870
110 t=rnd (-ti)                                                 1042
120 tba=0:hz=1:vt=2:hu=1:co=2                                   2715
130 rs=6:cs=6:sq=rs*cs                                          2228
140 rf=11-rs:cf=19-cs                                           1916
200 r=rs+2:c=cs+2:dim lc(r,c),vh(r,c),rt(4,4)                   3228
210 t=rnd (-ti)                                                 1042
220 for i=0 to 3:for j=0 to 3:read rt(i,j):next j:next i        2888
230 data 3,4,2,6,4,5,1,7,2,1,0,8,6,7,8,9                        3340
240 dim sv(1):sv(0)=0:sv(1)=1                                   2128
250 q$="[3cbm t]EDC[shift *]FR[3cbm @]RF[shift *]CDE":q$=q$+q$:q$=q$+q$
                                                                4303
260 for i=1 to 9:read rm(i),cm(i):next i                        2827
270 data 1,-1,2,0,1,1, 0,-2, 0,0, 0,2, -1, -1, -2,0, -1,1      3864
900 gosub 7000                                                   575
910 pl=hu:sv(0)=0:sv(1)=1                                       2159
990 if pl=hu then gosub 1500:goto 1010                          2163
1000 gosub 5000:if dir=tba then 2000                            2173
1010 gosub 5500:gosub 5600                                      1257
1020 gosub 5800:if n=0 then pl=3-pl                             2432
1030 if sq>0 then 990                                           1044
1040 if hq>cq then 1070                                         1322
1050 if hq=cq then 1080                                         1332
1060 print "[home]c64 gewinnt mit ";cq;" gegen ":print "hq;"felder";:gosub4:
     goto1100                                                   5734
1070 print "[home]sie gewinnen mit ";hq;" gegen ";cq;"felder";:gosub 3:goto
     1100                                                       5189
1080 print "[home]unentschieden mit je ";hq;" feldern";:gosub 3:gosub 4
                                                                4545
1100 print "[left]."                                            486
1105 print "[38spaces,up]"                                      1613
1110 input "noch ein spiel ? j[3left]";t$                       2542
1120 if left$ (t$,1)="j" then 1200                              1705
1130 if left$ (t$,1)<>"n" then print "[up]";:goto 1105          2955
1140 end                                                         128
1200 for i=0 to rs+2:for j=0 to cs+2:lc(i,j)=0:vh(i,j)=0:next j:next i
                                                                4766
1210 sq=rs*cs:cq=0:hq=0                                         2116
1290 goto 900                                                    572
1500 r=sv(0):c=sv(1)                                            1342
1510 gosub 9000                                                  579
1520 pk$=chr$ ( peek (crt+40*(r+rf)+(c+cf))+128)                2945
1530 get t$:if t$<>"" then 1600                                 1925
1540 print mid$ ("GD",1+(c and 1),1);"[left]";                  2492
1550 print pk$;"[left]";                                         817
1560 goto 1530                                                   603
1600 if t$<>"r" and t$<>"t" and t$<>"y" and t$<>"f" and t$<>"g" and t$<>"h" then
     uf=1                                                       5657
1601 if t$<>"h" and t$<>"v" and t$<>"b" and t$<>"n" and uf=1 then uf=0:goto 1700
                                                                5215
1602 if t$="r" then t$="7"                                      1613
1603 if t$="t" then t$="8"                                      1693
1604 if t$="y" then t$="9"                                      1614
1605 if t$="f" then t$="4"                                      1691
1606 if t$="g" then t$="5"                                      1739
1607 if t$="h" then t$="6"                                      1532
1608 if t$="v" then t$="1"                                      1645
1609 if t$="b" then t$="2"                                      1531
1610 if t$="n" then t$="3"                                      1676
1611 t=val (t$):r=r+rm(t):c=c+cm(t)                             2378
1620 if r<0 or r>2*rs or c<0 or c>2*cs then r=r-rm(t):c=c-cm(t) 4598
1700 if t$<>chr$ (13) then 1800                                 1622
1705 gosub 2                                                     241
1710 row=int (r/2)+1:col=int (c/2)+1                            2862
1720 v=(r and 1):h=1-v:dir=hz:if v then dir=vt                  3436
1730 if vh(row,col) and (vt*v+hz*h) then 1530                   3400
1740 sv(0)=r:sv(1)=c                                            1945
1750 return                                                      142
1800 if t$<>" " then 1510                                       1345
1810 input "[home,20down]wirklich aufhoeren ?";af$:if af$<>"j" then 1510
                                                                5245
1820 print "[clr]":end                                          495
5000 mx=-1:dir=tba:row=col=-1                                   2498
5005 r=2*rs+1:c=0                                               1472
5010 for i=1 to rs                                               974
5020 :for j=1 to cs+1                                           1063
5025 ::gosub 5200                                                450
5030 ::if vh(i,j) and vt then 5070                              1728
```


WWW.HOMECOMPUTERWORLD.COM

```
5040 :::f=rt(lc(i,j-1),lc(i,j))                    2257
5050 :::if f<mx or (f=mx and rnd (1)>.5)           3570
     then 5070
5060 :::dir=vt:mx=f:row=i:col=j                    3121
5070 :::next j                                      590
5080 ::next i                                       258
5110 for i=1 to rs+1                               1108
5120 :for j=1 to cs                                 795
5125 ::gosub 5200                                   450
5130 ::if vh(i,j) and hz then 5170                 1840
5140 :::f=rt(lc(i-1,j),lc(i,j))                    1876
5150 :::if f<mx or (f=mx and rnd (1)>.5)           3572
     then 5170
5160 :::dir=hz:mx=f:row=i:col=j                    3120
5170 :::next j                                      590
5180 ::next i                                       258
5182 gosub 3                                        243
5185 gosub 9000:print "[39spaces]";                2104
5190 return                                         142
5200 gosub 9000:print left$ (q$,2*cs+1);           2505
5205 uf=uf+8:if uf>205 then uf=0                   2185
5207 poke vo,1:poke sz,33:poke hf,uf:poke          2960
     sz,32
5210 q$=mid$ (q$,2,255)+left$ (q$,1)               2040
5220 return                                         142
5390 :
5400 tj=ti+dl                                       884
5410 if ti<tj then 5410                            1372
5420 return                                         142
5490 :
5500 lc(row,col)=lc(row,col)+1                     3105
5510 if dir<>hz then 5540                          1283
5520 lc(row-1,col)=lc(row-1,col)+1                 3475
5530 vh(row,col)=vh(row,col) or hz:return          3489
5540 lc(row,col-1)=lc(row,col-1)+1                 3655
5550 vh(row,col)=vh(row,col) or vt:return          3491
5590 :
5600 r=(row-1)*2:c=(col-1)*2                       3178
5605 if (r and 1) or (c and 1) then stop           1108
5610 gosub 9000                                     579
5620 t$="[down,shift -]":if dir=hz then t$=        2359
     "[right,shift *]"
5630 print t$;:if pl=hu then return                1716
5640 dl=5:for qq=0 to 6                            1651
5650 gosub 5400:print "[left] ";                   1351
5660 gosub 5400:print "[left]";right$ (t$,         1982
     1);
5670 next qq:return                                 802
5690 :
5700 r=r*2-1:c=c*2-1:gosub 9000                    1732
5710 sq=sq-1                                        815
5720 if pl=hu then print "W";:hq=hq+1:gosub        3848
     3:gosub 3:return
5730 print "Q";:cq=cq+1:gosub 4:gosub 4:           2350
     return
5790 :
5800 n=0                                            371
5810 r=row:c=col:gosub 5900                        1878
5820 r=row-1:c=col:gosub 5900                      1885
5830 r=row:c=col-1:gosub 5900                      1918
5840 return                                         142
5890 :
5900 if lc(r,c)<>4 or vh(r,c) and 12 then          2535
     return
5910 vh(r,c)=vh(r,c) or 12*pl:n=n+1                3312
5920 gosub 5700:return                              833
5990 :
6000 print "[clr]";                                 495
6010 for i=1 to cs:print "[cbm r] ";:next           2411
     i:print "[cbm s]"
6020 print "[up,cbm a,down]"                         509
6030 for j=2 to rs                                  984
6040 for i=1 to cs:print "[shift +] ";:next         2145
     i:print "[cbm w]"
6050 print "[up,cbm q,down]"                         469
6060 next j                                         278
6070 for i=1 to cs:print "[cbm e] ";:next           2392
     i:print "[cbm x]"
6080 print "[up,cbm z,home]";                        723
6090 return                                         142
6900 end                                            128
7000 print "[clr]":l$="[cbm a] ":r$="             2307
     [cbm s] ":m$="[cbm r] "
7010 for i=2 to rs:l$=l$+"[cbm q] ":r$=r$+"        4249
     [cbm w] ":m$=m$+"[shift +] ":next i
7020 l$=l$+"[cbm z] ":m$=m$+"[cbm e] ":r$=         3161
     r$+"[cbm x] "
7030 for r=0 to 2*rs step 2                        1218
7040 c=0:gosub 9000:print mid$ (l$,r+1,2);         2248
7050 for c=2 to 2*cs-2 step 2:print mid$           3090
     (m$,r+1,2);:next c
7060 print mid$ (r$,r+1,1)                         1446
7070 next r                                         294
7080 return                                         142
7400 sm=54272:hf=sm+1:vo=sm+24:sz=sm+4:for         5561
     i=sm to sm+24:poke i,0:next
7410 poke sm+5,8:poke sm+5,8:return                1570
8990 :
9000 print "[home]";:r=r+rf:c=c+cf                 2020
9010 if r then print left$ ("[24down]",r);         3082
9020 if c then print left$ ("[39right]",c);        6488
9030 r=r-rf:c=c-cf                                 1501
9040 return                                         142
50000 print "[down,rvs]return[off] druecken         2237
      !!! "
50010 get z$:if z$="" then 50010                   1589
50020 return                                        142
60000 print "[clr,3down,4spaces]softteam           4549
      proudly presents":print :print :print
      "[2down]"
60005 print "[11spaces,rvs]cleverness 64           3295
      [off]":print :print "[2down]"
60010 print "[4spaces]c64 version von              3057
      softteam[10spaces]"
60011 print :print :gosub 3:gosub 4:gosub 3:       1966
      gosub 4
60020 input "[5down]erklaerungen ?[2spaces]n       2497
      [3left]";t$
60030 if left$ (t$,1)="n" then return              1680
60040 print "[clr,2down,rvs]cleverness[off,       2214
      2down]"
60050 print "spielen sie [rvs]cleverness           3952
      [off] gegen ihren"
60060 print "commodore 64.[2spaces]nehmen          3486
      sie die heraus-"
60070 print "forderung an !!![down]"               1491
60080 print "versuchen sie, die felder so          3292
      einzu-"
60090 print "kesseln, dass sie mit ihrem zug       3032
      das "
60100 print "gesamte quadrat umschliessen.         3411
      das feld"
60105 print "gehoert dann ihnen und sie            3360
      duerfen "
60110 print "nochmals ziehen.[down]"               2035
60120 print "waehrend der c64 denkt,               3363
      zeichnet er"
60130 print "ein wellenmuster auf den              4355
      bldschirm,"
60140 print "begleitet von ansteigenden            3809
      toenen."
60150 print "sene denkzeitliegt zwischen 5         3869
      sekunden"
60160 print "und maxmal einer minute."             2611
60170 print                                         153
60180 gosub 50000                                   577
60190 print "[clr,2down,rvs] cleverness[off]       4308
      c64-version by softteam[3spaces,down]
      "
60200 print "die steuerung erfolgt ueber :"        2890
60210 print "taste[2spaces]funktion:[down]"         1817
60220 print " f -[3spaces]links[4spaces]"           1467
60230 print " h -[3spaces]rechts"                   1348
60240 print                                         153
60250 print " t -[3spaces]hoch"                     1165
60260 print " b -[3spaces]runter"                   1355
60270 print                                         153
60280 print " y -[3spaces]schraeg rechts            2424
      hoch"
60290 print " r -[3spaces]schraeg links            2212
      hoch"
60300 print " v -[3spaces]schraeg links            2455
      runter"
60310 print " n -[3spaces]schraeg rechts           2758
      runter"
60330 print "[down,rvs]return[off] c64 nimmt       2485
      zug an"
60340 print "[down,rvs]space[off,2spaces]          2506
      beendet das spiel"
60350 print "(mit sicherheitsabfrage)"             2500
60360 gosub 50000                                   577
63000 return                                        142
63030 for i=50000 to 50035:read a:poke i,a:        2626
      next
63040 gosub 7400:rem sound                          662
63050 sys 50000:print chr$ (147);chr$ (5)          2471
63055 print chr$ (142);"[yel]"                     1192
63060 goto 50:rem zurueck zum hauptprogramm         644
```

Der Commodore C64 fordert heraus in „Cleverness“

WWW.HOMECOMPUTERWORLD.COM

RUN an die Geschütze

System: C64
Typ: Spiel
Sprache: Basic

Zwei Spieler sitzen sich in ihren Kampfgeschützen gegenüber. Schußwinkel und Schußkraft sind präzise einzugeben, damit man den Gegner zerstört.

Nachdem das Programm gestartet ist und die Namen der Spieler eingegeben sind, beginnt „Artillery" per Zufallszahlen ein Spielfeld zu entwerfen. Zwei Geschütze erscheinen am Bildschirm, ein Gebirgsrücken entsteht. Windstärke und -richtung zeigt das Programm in der ersten Zeile. Danach gibt Spieler 1 seine Schußwerte ein. Höhenunterschiede sind genauso zu berücksichtigen wie die momentanen Windverhältnisse. Um über die hohe Gebirgskette zu schießen, ist ein Schußwinkel von mindestens 80 Grad ratsam. Mit 6.5 ist die Schußkraft bei Rückenwind ausreichend, um das Geschoß an sein Ziel zu schleudern. Fliegt eine Granate höher, als das Spielfeld anzeigt, errechnet „Artillery" weiterhin die Koordinaten, denn oftmals taucht ein Geschoß wieder am Bildschirm auf und findet sein Ziel. Ein Spieldurchgang geht so lange, bis einer der beiden Kontrahenten zerstört ist. Danach muß man entscheiden, ob man weiterspielt oder endet. Dabei zeigt der Computer den aktuellen Spielstand an. Jedes Spielfeld wird neu entwickelt. Dadurch sind immer neue Schußwerte notwendig, um den Gegner zu vernichten.

softteam

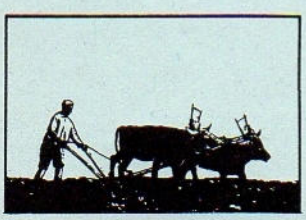
```
  1 gosub 63030:goto 5                                          1030
  2 poke vo,15:poke sz,33:poke hf,5:poke                        3467
    sz,32:return
  3 for mz=1 to 255 step 4:poke vo,7:poke                       5363
    sz,33:poke hf,mz:poke sz,32:next mz:
    return
  4 for mz=255 to 1 step -4:poke vo,7:poke                      5326
    sz,33:poke hf,mz:poke sz,32:next mz:
    return
  5 ho$="[home]"+s$+s$+"[home,rvs] ":s$="                       3954
    [home,37spaces]"
  6 print "[clr,3down,2right,rvs,3spaces]a                     2117
     r t i l l e r y[3spaces,off]"
  7 input "[home,7down]name des                                 4296
     1.spielers(links) : ?[3left]";n1$
  8 if n1$="?" then 7                                           1136
  9 input "[home,9down]name des                                 5079
     2.spielers(rechts) : ?[3left]";n2$
 10 if n2$="?" then 9                                           1144
 11 dim a(100),b(100),h(39),a1(2),b1(2),                        4000
    a2(2),b2(2),l(2)
 50 for i=1 to 500:next i                                       990
 53 print "[clr]":poke 59468,12:gosub 8000                     2337
 56 z$="[rvs,cbm +,up,left]":for x=1 to 5:                      2957
    z$=z$+z$:next x
 60 a$="[right]":for x=1 to 6:a$=a$+a$:                         4406
    next x:b$="[down]":for x=1 to 5:b$=b$+
    b$:next x
 80 n=2                                                         379
 90 rem

130 x1=int (6+rnd (1)*8)                                        1851
140 x2=int (25+rnd (1)*8)                                       1744
150 l(1)=int (3+(x1-3)*rnd (1))                                 2427
160 l(2)=int (x2+1+(36-x2)*rnd (1))                             2620
170 h9=rnd (1)*11+1                                             1391
180 for i=1 to x1:h(i)=h9:next i                                1501
210 h9=rnd (1)*11+1                                             1391
220 for i=x2 to 39:h(i)=h9:next i                               2156
250 if abs (h(1)-h(39))<4 then 170                              2545
260 x3=x1+int ( rnd (1)*(x2-x1-6))+3                            2591
270 h(x3)=rnd (1)*11+9                                          1520
290 d1=(h(x3)-h(1))/2                                           1365
300 d2=(h(x3)-h(39))/2                                          1558
310 if d1*d2<=0 then 470                                        1558
320 j=-180                                                      623
330 k=180/(x3-x1+1)                                             1457
340 h9=h(1)+d1                                                  947
350 for i=x1+1 to x3-1                                          1434
360 j=j+k                                                       481
370 h(i)=h9+cos (j*[pi]/180)*d1                                 1790
380 next i                                                      276
390 j=0                                                         367
400 k=180/(x2-x3+1)                                             1472
410 h9=h(39)+d2                                                 1418
420 for i=x3+1 to x2-1                                          1464
430 j=j+k                                                       481
440 h(i)=h9+cos (j*[pi]/180)*d2                                 1806
450 next i                                                      276
460 goto 550                                                    584
470 d=(h(1)-h(39))/2                                            1722
480 j=0                                                         367
490 k=180/(x2-x1+1)                                             1456
500 h9=h(39)+d                                                  1218
510 for i=x1+1 to x2-1                                          1432
520 j=j+k                                                       481
530 h(i)=h9+cos (j*[pi]/180)*d                                  1771
540 next i                                                      276
550 print "[clr]";                                             495
560 w=int (100*( rnd (1)-.5))                                   1725
570 print left$ (b$,22);                                        999
580 for i=1 to 39:h(i)=int (h(i)+.5):next                       2636
    i
590 for x=1 to 39:if x=1 then 610                               2001
600 print "[right]";                                           533
610 print mid$ (z$,1,h(x)*4);                                   1858
650 for z=1 to h(x):print "[down]";:next z                      1691
660 next x:print "[off]"                                        994
670 print "[home]";                                            493
680 print mid$ (b$,1,22-h(1));:if l(1)=1                        2687
    then 700
690 print mid$ (a$,1,l(1)-1);                                   1649
700 print "[left,rvs,shift f]";left$ (n1$,                      2989
    1);"[cbm *,off,up,2left]N";
710 print "[home]";                                            493
720 print mid$ (b$,1,22-h(39));                                 1784
730 print mid$ (a$,1,l(2)-1);                                   1653
740 print "[left,rvs,shift f]";left$ (n2$,                      2929
    1);"[cbm *,off,up,2left]M";
750 print "[home]";                                            493
760 print mid$ (b$,1,24);                                       1312
770 print "[8spaces]";                                         829
780 k=1                                                         372
790 if w>=0 then 820                                            994
800 print -w;" km/h wind[2spaces]( <== )";                      3002
810 goto 840                                                    586
820 print w;" km/h wind[2spaces]( ==> )";                       2574
830 print "[home]";                                            493
840 print "[home]";:n=3-n:if n=1 then 880                       2545
850 print ho$;n2$+" ";:input "[home,                            6362
    15right]winkel:[3spaces]?[4spaces,
    7left]";e
```


WWW.HOMECOMPUTERWORLD.COM

```
855 print s$                                                    463
860 print ho$;n2$+" ";:input "[home,                          5858
    15right]pulver:[3spaces]?[3left]";v
865 print s$                                                    463
870 q%=3:goto 910                                             1000
875 print s$                                                    463
880 print ho$;n1$+" ";:input "[home,                          5743
    15right]winkel:[3spaces]?[3left]";e
885 print s$                                                    463
890 print ho$;n1$+" ";:input "[home,                          5794
    15right]pulver:[3spaces]?[3left]";v
900 q%=5                                                        526
910 k=k+1:if e=0 then 90                                      1420
930 a2(n)=a2(n)-2.86                                          1502
940 e=90-e                                                      881
950 if n=1 then 970                                           1024
960 e=-e                                                        386
970 x1=l(n)                                                     891
980 y1=h(l(n))+1                                              1201
990 t=0:m=0                                                     896
1010 v1=sin (e*[pi]/180)*v                                    2303
1020 v2=cos (e*[pi]/180)*v*.7                                 2539
1030 t=t+.5:m=m+1                                             1690
1040 v1=v1+(v1+w*.5)/1500                                     2205
1050 x=x1+(v1+w*.5/50)*t                                      2110
1060 x=int (x+.5)                                             1020
1070 y=y1+(v2-t/3)*t                                          1767
1080 y=int (y+.5)                                             1037
1090 z=23-y:print "[home]";                                   1544
1092 a(m)=z:b(m)=x:if x>39 or x<1 then 1350                   3215
1094 if y-h(x)<=0 then 1340                                   1755
1100 if z<0 then 1150                                         1092
1110 if z=0 then 1130                                         1087
1120 print left$ (b$,z);:if x=1 then 1140                     2199
1130 print left$ (a$,x-1);                                    1269
1140 print "*":gosub 2                                        886
1150 print "[home]";                                          493
1160 if m-4<1 then 1030                                       1452
1162 if a(m-4)=0 then 1180                                    1615
1164 if a(m-4)<0 or b(m-4)<=0 then 1030                       2641
1170 print left$ (b$,a(m-4));                                 1469
1180 print left$ (a$,b(m-4)-1);                               1731
1190 print " ";                                                 545
1200 print "[home]";                                          493
1220 goto 1030                                                583
1230 if m=1 then 1030                                         1005
1232 if b(m)<1 or b(m)>39 then 840                            2007
1233 rem schwierigkeitsgrad liegt bei '1.5'
     !!!
1234 gosub 4:if abs (b(m-1)-l(n))<1.5 then                    3241
     1250
1240 goto 1260                                                  615
1250 if t>1 then 1510                                         1079
1260 if abs (b(m-1)-l(3-n))<1.5 then 1560                     2976
1270 if t<1 then 1030                                         1023
1280 goto 1460                                                  623
1290 print "[home]"                                           314
1300 for i=0 to 1000:next                                       928
1310 gosub 8000:goto 90                                         950
1340 print "[home]";                                          493
1350 for j=4 to 1 step -1                                     1120
1360 if m-j<=0 then 1440                                      1407
1370 if a(m-j)=0 then 1410                                    1466
1380 if a(m-j)<0 or b(m-j)<=0 then 1440                       2650
1400 print left$ (b$,a(m-j));:if b(m-j)=1                     3079
     then 1420
1410 print left$ (a$,b(m-j)-1);                               1564
1420 print " ";                                                 545
1430 print "[home]";                                          493
1440 next j:if m=1 goto 1600                                  1650
1450 if b(m)<1 or b(m)>39 then 1610                           2137
1455 goto 1230                                                  591
1460 if b(m-1)=1 then 1470                                    1530
1465 print left$ (a$,b(m-1)-1);                               1719
1470 print left$ (b$,23-h(b(m-1)));                           2650
1472 for z=1 to 3:print "[up,left]£W/[up,                     4305
     3left]£!/";:for x=1 to 5:next x
1474 print "[3left,down,3spaces,up,3left,                    3971
     3spaces,2left,2down]";:for x=1 to 5:
     next x:next z:print "[cbm o]";
1480 goto 1600                                                  583
1510 print "[home]";                                          493
1520 print left$ (a$,l(n)-1);                                 1426
1530 print left$ (b$,24-h(l(n)));                             2126
1540 for z=1 to 9:print "[up,left,rvs,                        6031
     cbm *]Q[shift £,off,up,4left]£[cbm b]!
     [rvs,cbm b,off]/[5left,up]£M!N/[2left,
     2down]";"[left]";
1541 w%=8                                                       556
1542 for x=1 to 5:next x                                      1179
1544 print " [3left,up,5spaces,5left,up,                      3185
     5spaces,3left,up,4down]";
1545 for x=1 to x:next x:next z                               1360
1546 for x=1 to 50:next x                                     1231
1548 gosub 3:gosub 3:gosub 3                                    957
1550 goto 1290                                                  639
1560 print "[home]";                                          493
1570 print left$ (a$,l(3-n)-1);                               1429
1580 print left$ (b$,24-h(l(3-n)-1));                         2565
1590 goto 1540                                                  611
1600 rem
1610 w=w+int (10*( rnd (1)-.5))                               1729
1620 print "[home]";left$ (b$,24);"                          1762
     [8spaces]";
1630 if w>=0 then print "[3spaces,3left]";                    4998
     w;" km/h wind[2spaces]( ==> )";:goto
     840
1640 print "[3spaces,3left]";-w;" km/h wind                  3780
     [2spaces]( <== )";:goto 840
7400 sm=54272:hf=sm+1:vo=sm+24:sz=sm+4:for                    5561
     i=sm to sm+24:poke i,0:next
7410 poke sm+5,8:poke sm+5,8:return                           1570
8000 print "[clr,3down,rvs,3spaces]a r t i                   1733
      l l e r y[3spaces,off]"
8010 print "[5down]";                                         620
8050 if q%=5 and w%=8 then s1=s1+1                            2624
8055 if q%=3 and w%=8 then s2=s2+1                            2481
8100 print n1$;tab(20);s1;tab(24)"siege !!"                   2739
8200 print "[3down]";                                         716
8300 print n2$;tab(20);s2;tab(24)"siege !!"                   2751
8500 print "[5down,4right] noch ein spiel ?                   3634
      (j oder n)
8600 get q$:if q$="" then 8600                                1570
8700 if q$="n" then end                                       1304
8750 q%=0:w%=0:a%=0                                           1304
8800 return                                                     142
63000 data 169,0,141,32,208,141,33,208,141,                   5109
     0,4,141,1,4
63010 data 141,2,4,169,149,141,0,221,169,5,                   4086
     141,24
63020 data 208,169,128,141,136,2,133,56,96,                   4236
     96
63030 for i=50000 to 50035:read a:poke i,a:                   2626
     next
63040 gosub 7400                                                591
63050 sys 50000:print chr$ (147)                              1728
63060 return                                                    142
63200 rem commodore 64 version
63220 rem <c> by  softteam
```

Unter Beschuß: Artillery



»BEBAUEN UND BEWAHREN – das ist der Auftrag, den Gott den Menschen für die Schöpfung mitgegeben hat – nun ist er selbst für sie zur Bedrohung geworden.
Besonders hart treffen die Folgen des Raubbaus an der Schöpfung die arme Bevölkerung in den Entwicklungsländern. Sie verliert ihre Lebensgrundlage... Viele Partner von ›Brot für die Welt‹ versuchen mit großem Einsatz die Ursachen der Not zu erkennen und zu beheben. Wir wollen sie mit unseren Spenden tatkräftig unterstützen.«
(Auszug)

Postfach 476, 7000 Stuttgart 1

Brot für die Welt

Postgiro Köln 500500-500

WWW.HOMECOMPUTERWORLD.COM

Mit dem Kopf durch die Wand

System : C16/116/Plus4
Typ : Spiel
Sprache: Basic

In „Break-Out" geht es darum, einen Ball gegen Mauerstücke zu feuern, um damit die Wand einzureißen. Je nach Aufschlagwinkel kommt der Ball direkt oder angeschnitten zurück. Nur mit viel Geschick gelingt es, die Mauer komplett niederzureißen

Mit Joystick oder Tastatur steuert man seine Fahrbasis am unteren Bildschirmrand. Trifft der Ball auf dieses Spielbrett, schleudert er wieder auf die Wand zu und trifft dort (hoffentlich) einen Stein. Ist die Mauer komplett abgeschossen, ertönt eine Sieges-Hymne. Danach erscheint eine weitere Mauer. Allerdings ist der Sprengball diesmal schneller und das Spiel daher schwieriger. Gute Nerven und schnelle Reaktionen sind Grundlage, um den auf Kassette gespeicherten High-Score zu brechen.

Deshalb ballert's im C16

Zeile 120 bis 180
— Anfangsbildschirm zeichnen, Variablendefinition
Zeile 190
— Zufallsrichtung des Balls am Anfang des Spiels bestimmen
Zeile 200
— Ballbewegung: neue Position auf Kollision untersuchen, Ball zeichnen
Zeile 210 bis 320
— 1. Ballroutine:
  — Ball auf dem Boden
  — Richtungsänderung des Balles nach Kollision mit Wand oder Schläger
Zeile 330 bis 350
— Joystick/Tastaturroutine
Zeile 360 bis 380
— Spieler (Schläger) zeichnen und bewegen
Zeile 390 bis 400
— Steine zeichnen
Zeile 410 bis 430
— Score erhöhen, anzeigen
Zeile 440 bis 470
— 2. Ballroutine
  — Richtungsänderung des Balls nach Abschuß eines Steins
Zeile 480 bis 510
— Lebensabzugsroutine
Zeile 520 bis 560
— Endroutine
— Highscore?
— Highscore abspeichern?
Zeile 570 bis 575
— Ballwarteschleife verringern
— Freileben geben
Zeile 580 bis 680
— Datas für Vorspann und Hymne
Zeile 690 bis 720
— Hymne spielen
Zeile 730 bis 820
— Vorspann:
  — Highscore einladen?
  — Steuerung mit Tastatur/Joystick
Zeile 830
— Highscore auf Band schreiben

Wichtige Variable

HS	= Highscore
N$	= Name des Highscoreinhabers
L	= Leben
SC	= Score
S	= Position des Schlägers
J	= Ballwarteschleife (vom Level abhängig)
V	= Anzahl der abgeschossenen Steine pro Level
Q	= Scoremultiplikator (vom Level abhängig)
A	= Position des Balls
M	= Richtung des Balls
JT	= Joystick/Tastaturvariable

Martin Goes

```
 10 rem "    U----------------------I
 20 rem "    |                      |
 30 rem "    |   b r e a k - o u t  |
 40 rem "    |                      |
 50 rem "    |    von martin goes   |
 60 rem "    |   philosophenweg 2   |
 70 rem "    |    7400 tuebingen    |
 80 rem "    |                      |
 90 rem "    |alle rechte vorbehalten|
100 rem "    |                      |
110 rem "    |    ein  spiel der    |
111 rem "    | i m a g o software   |
112 rem "    |                      |
113 rem "    J----------------------K
114 :
115 :
116 :
117 :
118 :
119 :
```

```
120 vol 8:gosub 730                                   900
130 q=-1:l=9:sc=0:s=20:j=40:v=0:color 0,1             3443
140 q=q+1:color 1,8:scnclr :for i=3112 to             4256
    3151:poke i,64:poke i+920,64:next
150 poke 3112,91:poke 3151,91:poke 3992,              3836
    91:poke 4031,91
160 for i=3152 to 3952 step 40:poke i,66:             4695
    poke i+39,66:next :gosub 390
170 char 1,0,0," ":print "[11right]li:";l;            5508
    "[5right]high:";hs
180 s=20:char 1,s,23,"[2cbm +]":a=3972-               4010
    160:o=4010
190 y=int ( rnd (0)*2)+1:if y=1 then m=-              4354
    41:else if y=2 then m=-39
200 gosub 330:a=a+m:poke a-m,32:poke a,81:            6123
    b=peek (a+m):if b=32 then 200:else if
    b=98 then 410
210 sound 1,168,3:if b=66 then 250:else if            5011
    b=64 then 220:else if b=102 then 300:
    else 320
```


WWW.HOMECOMPUTERWORLD.COM

```
220 if b=64 and m=41 then 480                    1740
230 if b=64 and m=39 then 480                    1755
240 if b=64 and m=-41 then m=39:goto 200         2733
250 if b=64 and m=-39 then m=41:goto 200         2568
260 if b=66 and m=41 then m=39:goto 200          2390
270 if b=66 and m=-39 then m=-41:goto 200        2643
280 if b=66 and m=39 then m=41:goto 200          2435
290 if b=66 and m=-41 then m=-39:goto 200        2748
300 if b=102 and m=39 then m=-41:goto 200        2568
310 if b=102 and m=41 then m=-39:goto 200        2673
320 if b=91 then m=-m:goto 200                   1795
330 for i=0 to j:next :on jt goto 350,340        2094
340 get a$:if a$="[right]" then n=2:goto         5191
    360:else if a$="[left]" then n=-2:goto
    360:else return
350 if joy (1)=7 then n=-2:else if joy (1)       3898
    =3 then n=2:else return
360 s=s+n:if s>=1 and s<=37 then 380:else        3658
    s=s-n
370 char 1,s,23,"[2cbm +]":char 1,s-n,23,"       4824
    [2spaces]":poke 3992,91:poke 4031,91:
    return
380 char 1,s-n,23,"[2spaces]":char 1,s,23,       2971
    "[2cbm +]":return
390 for i=1 to 3:char 1,0,1+i*3,"[red,           5588
    4right,3cbm i,2spaces,3cbm i,2spaces,
    3cbm i,2spaces,3cbm i,2spaces,3cbm i,
    2spaces,3cbm i,2spaces,3cbm i]":next
400 return                                        142
410 sc=sc+10+10*q:v=v+1:for i=a+m-2 to a+m        6995
    +2:poke i,32:next :if v=21 then v=0:
    goto 570
420 char 1,0,0," ":print sc:sound 1,810,          4777
    10:sound 2,854,10:sound 1,854,10
430 sound 2,881,10                               1067
440 if m=-41 then m=39:goto 200                  2067
450 if m=-39 then m=41:goto 200                  2022
460 if m=41 then m=-39:goto 200                  2059
470 if m=39 then m=-41:goto 200                  2149
480 l=l-1:for i=3993 to 4030:poke i,32:          2754
    next
490 for i=1000 to 100 step -40:sound 3,i,        4873
    2:next :char 1,0,0," ":if l<0 then 520
500 print "[11right]li:";l                       2247
510 goto 180                                      588
520 print "[clr,rvs,yel,11spaces]b r e a k       4151
     - o u t[12spaces]":for t=1 to 10:get
    r$:next
530 if sc>hs then hs=sc:input "[pur,3down]       4984
     wie heissen sie ";n$:if n$="" then n$
    ="---"
540 print "[4down,cbm 3,5right]highscore:        2997
    [right]";n$,hs
550 print "[cbm 4,6down] highscore               5869
     abspeichern ? (j/n) ":get key a$:if
    a$="j" then gosub 830
560 goto 120                                      564
570 j=j-10:if j<=0 then j=0:s=20                 2146
575 l=l+1:if l>=9 then l=9                       1592
580 data v,o,n[2spaces],m,a,r,t,i,n              3420
    [2spaces],g,o,e,s
590 data 1,881,10,2,0,0,1,834,20,2,0,0,1,        4777
    810,10,2,881,10
600 data 1,796,20,2,834,20,1,768,10,2,810,       7546
    10,1,739,20,2,796,20,1,768,10,2,768,10
610 data 1,796,5,2,739,10,1,768,5,1,739,5,       5170
    2,644,10,1,1022,5
620 data 1,796,10,2,739,10,1,834,20,2,796,       5677
    20,1,854,10,2,810,10
630 data 1,834,20,2,796,20,1,810,10,2,768,       7201
    10,1,796,30,2,739,30,1,768,20,2,721,20
640 data 1,881,10,2,0,0,1,834,20,2,0,0,1,        4777
    810,10,2,881,10
650 data 1,796,20,2,834,20,1,768,10,2,810,       7546
    10,1,739,20,2,796,20,1,768,10,2,768,10
660 data 1,796,5,2,739,10,1,768,5,1,739,5,       5170
    2,644,10,1,1022,5
670 data 1,796,10,2,739,10,1,834,20,2,796,       5677
    20,1,854,10,2,810,10
680 data 1,834,20,2,796,20,1,810,10,2,768,       7162
    10,1,796,40,2,739,40,1,768,40,2,721,40
690 restore                                       140
700 for i=1 to 13:read a$:next :for u=1 to       5614
    34:read a,b,c,d,e,f:sound a,b,c:sound
    d,e,f
710 next :sound 1,796,110:sound 2,881,110        2432
720 for t=1 to 2000:next :goto 140               1426
730 color 0,1                                     640
740 print "[clr,rvs,yel,11spaces]b r e a k       2618
     - o u t[12spaces]"
750 restore                                       140
760 print :print :print :for i=1 to 13:          3523
    read a$:print a$;:sound 3,900,5
770 for u=1 to 150:next :next :print              1380
780 print "[5down,cyn]highscore einladen ?        5240
     (j/n)":get key a$:if a$="j" then 790:
    else 800
790 open 1,1,0,"highscore":input#1,n$:           3111
    input#1,hs:close 1
800 print :print "[wht,2down] j- joystick        3504
    [3spaces]k- keyboard":get key a$
810 if a$="k" then jt=2:else jt=1                2450
820 return                                        142
830 open 1,1,1,"highscore":print#1,n$:           3517
    print#1,hs:close 1:return
```

Destruktiv: Mauern fällen in „Break-Out"

Wir danken



Das deutsche Seenotrettungswerk wird ausschließlich durch freiwillige Zuwendungen finanziert. Zum Beispiel auch durch Ihre Spende.

Deutsche Gesellschaft zur Rettung Schiffbrüchiger

Werderstr. 2, 2800 Bremen 1 · Postscheck Hamburg

(BLZ 20010020) 7046-200 Wir danken für die gespendete Anzeige

WWW.HOMECOMPUTERWORLD.COM

FOG — Nebel des Grauens

System: C16/116/Plus 4
Typ: Spiel
Sprache: Basic

Dichter Nebel zieht auf. Gespenstische Wesen begleiten die Wolken. Einzige Hoffnung: Hilfe aus der Luft. Man muß Leuchtraketen durch die dünnsten Nebelschichten schießen.

Nachdem der Computer den Schwierigkeitsgrad, Eingabegerät (Joystick/Tastatur) und High-Score-Bedarf kennt, beginnt das Spiel. Nebel entsteht auf dem Spielfeld. Am unteren Bildschirmrand befindet sich der Spieler. Nach Abfeuern der Leuchtrakete muß man das Geschoß durch zahlreiche Nebelpunkte nach oben steuern. Stößt die Rakete auf ein Nebelpartikel, gibt es keine Punkte. Erreicht man den oberen Bildschirmrand, naht Rettung. Danach beginnt das Spiel in einem höheren Level. Nach 10 Runden gibt es einen Freischuß.

So funktioniert's!

Zeile 10 bis 40: Vorspann
— Spiel mit Joystick oder Tastatur
— Schwierigkeitsgrad
— Highscore von Kassette einladen?
Zeile 50 bis 100
Bildschirm zeichnen (Nebel, Abschußrampe)
Zeile 110
Rakete zeichnen, alte Raketenposition löschen
Zeile 111 bis 136
Joystick beziehungsweise Tastaturabfrage
Zeile 140 bis 160
Festsetzen der neuen Raketenposition und Überprüfung nach Crash, Bonuspunkten, Ziel Zeile 170
Lebensabzugsroutine
Zeile 190 bis 270
Abschußrampe löschen
Zeile 300
Crasheffekt
Zeile 310 bis 320
Rakete im Ziel
Zeile 330
Nebel auflockern
Zeile 350
Variablenreset für Neubeginn
Wichtige Variablen:

JT	=	Joystick/Tastaturvariable
Q	=	Schwierigkeitsgrad
K	=	Dichte des Nebels
LI	=	Leben
M,N	=	Richtunng der Spielerbewegung
X,Y	=	Nebelposition
O	=	Abschußrampenort
A,B	=	Raketenposition
SZ,SC	=	Scorevariablen
N$	=	Name des Highscoreinhabers

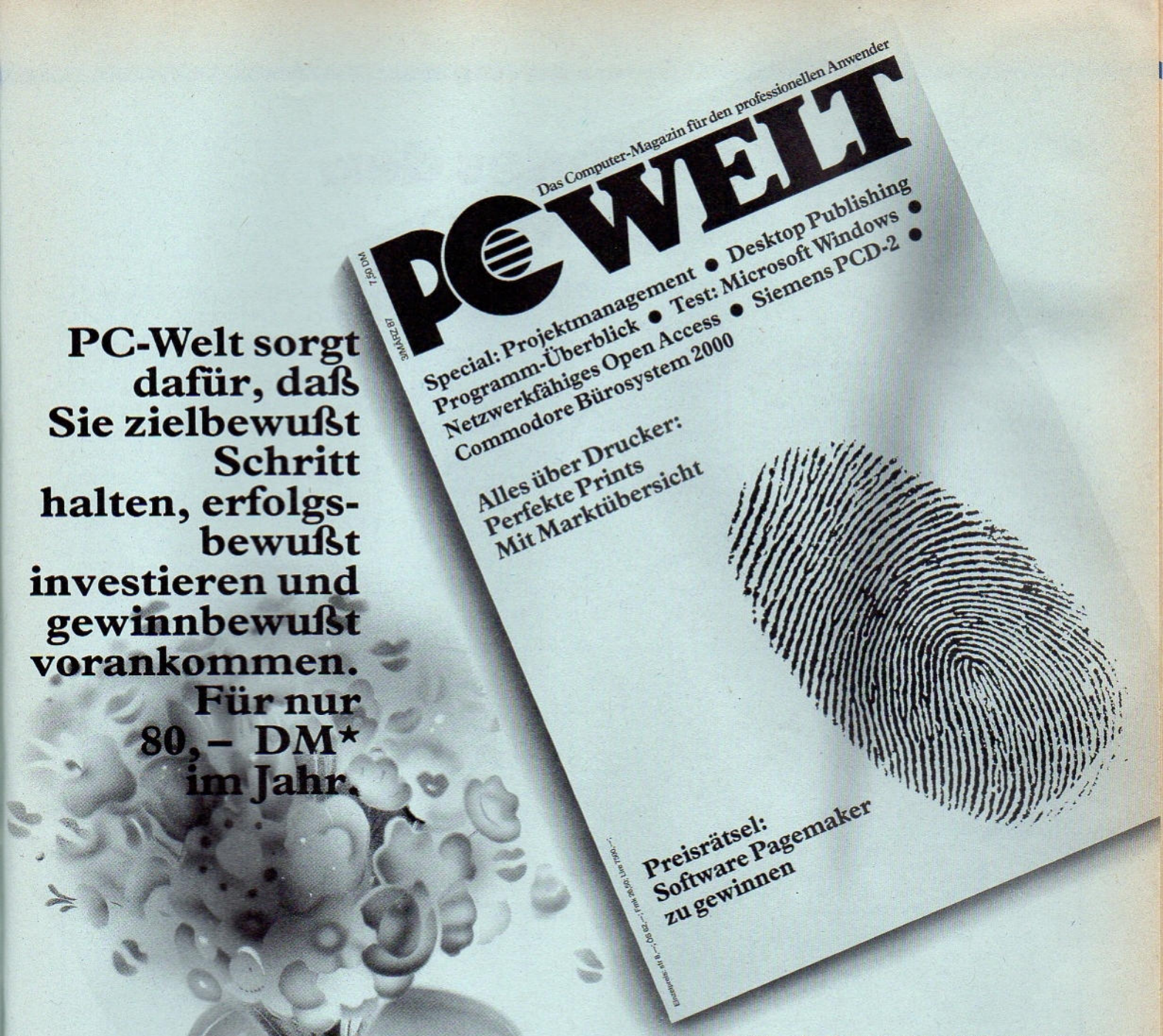
Martin Goes

```
10 graphic 0:color 0,2:print "[clr,rvs,          4234
   flashon,blk,7spaces]d a n g e r o u s
   [3spaces]f o g[8spaces]"
15 print "[2down]press j for joystick or        4745
    k for keyboard":get key a$
16 if a$="k" then jt=2:else jt=1                 2450
20 input "[3down,red,2spaces]difficulty          4656
   [4spaces](0-3) ";q:if q<0 or q>3 or q<
   >int (q) then 20
30 print "[2down,blu] shall i load the          6107
    highscore (y/n)":get key a$:if a$="y"
   then 40:else 50
40 open 1,1,0:input#1,n$:input#1,hs:close        2104
   1
50 graphic 3,1:k=250+q*70:vol 8:li=3             3131
60 color 1,2:color 3,3,4:color 0,1:color         5123
   2,7,4:m=0:n=2:z=1:color 4,10
70 y=int ( rnd (0)*100)+30:if y/2<>int (y        3805
   /2) then 70
80 x=int ( rnd (0)*159):draw z,x,y:s=s+1:        4599
   if s<>k then 70
90 b=188:o=int ( rnd (0)*150)+5:draw 2,o-        6956
   1,191 to o+1,191:draw 2,o-2,192 to o+
   2,192
100 draw 2,o,193 to o,194:draw 2,o-2,195         7297
   to o+2,195:draw 2,o-1,196 to o+1,196:a
   =o
110 draw 3,a,b:if m<>0 then draw 0,a-m,b         3832
   to a-m,b+1:else draw 0,a,b+4
111 on jt goto 120,135                            936
120 if joy (1)=0 then 140:else if joy (1)=       5689
   1 then m=0:n=2:else if joy (1)=3 then
   m=1:n=0
130 if joy (1)=7 then m=-1:n=0                   2292
131 goto 140                                      572
135 get a$:if a$="" then 140:else if a$=         5560
   [up]" then m=0:n=2:else if a$="[right]
   " then m=1:n=0
136 if a$="[left]" then m=-1:n=0                 2171
140 a=a+m:b=b-n:locate a,b:if rdot (2)=0         4274
   and a>0 and a<159 and b>5 then 110
160 if rdot (2)=1 or a<0 or a>159 then           5592
   170:else if rdot (2)=3 then 180:else
   if b<5 then 310
170 li=li-1:gosub 300:if li<0 then 190:          3922
   else gosub 280:goto 90
180 li=li+.1:for i=500 to 900 step 50:           3364
   sound 1,i,2:next :goto 110
190 graphic 0:print "[clr,flashon,7spaces]       5612
   sie erreichten"s2*15+sc"punkte":for i=
   1 to 10:get a$
200 next :sc=s2*15+sc:if sc>hs then hs=sc:       5497
   input "[2down] enter your name please"
   ;n$
210 print "[2down,rvs,cyn,8spaces]h a l l        2391
   [3spaces]o f[3spaces]f a m e[9spaces]"
220 if n$="" then n$="---"                       1851
230 print "[red,3down,8right]";n$,hs             2278
240 print "[2down,blu,6spaces]press s to         4892
    save the highscore[5spaces]"
250 print "[down,2left,3spaces]press any         4797
    other key to continue":for i=1 to 10:
   get a$:next
260 get key a$:if a$<>"s" then 350               2068
270 open 1,1,1,"highsore":print#1,n$:            3814
   print#1,hs:close 1:goto 250
280 draw 0,o-1,191 to o+1,191:draw 0,o-2,        5898
   192 to o+2,192:draw 0,a,b-1 to a,b+4
290 draw 0,o,193 to o,194:draw 0,o-2,195         7105
   to o+2,195:draw 0,o-1,196 to o+1,196:
   return
300 for i=900 to 500 step -50:color 0,2:         3674
   sound 1,i,2:color 0,1:next :return
310 sound 1,569,20:sound 2,684,20:sc=sc+3:       6644
   z=1:if sc=15 then s2=s2+1:sc=0:goto
   330
320 s=0:k=40+q*20:gosub 280:goto 70               2322
330 z=0:for i=1 to 50:color 0,2:sound 1,         4473
   int ( rnd (0)*1000),1:color 0,1:next :
   s=0
340 k=40+2*q*20:gosub 280:goto 70                2387
350 s2=0:sc=0:s=0:k=0:goto 10                    2185
```

FOG — undurchsichtiges Spiel im C16

WWW.HOMECOMPUTERWORLD.COM

PC-Welt sorgt dafür, daß Sie zielbewußt Schritt halten, erfolgsbewußt investieren und gewinnbewußt vorankommen. Für nur 80,– DM* im Jahr.



Erfolge sind berechenbar, im Computerzeitalter mehr als je zuvor. Machen Sie PC-Welt zu Ihrem Informations-Kapital. Jeden Monat mit dem kompletten Wissen über PCs & Co. PC-Welt vermittelt Ihnen erfolgreich den aktuellen Stand allen PC-Know-hows — bis ins Detail. Praxisecht und professionell — inclusive aller wichtigen Wettbewerber. Mit den objektiven und aussagefähigen Tests. So gründlich, aufwendig und umfassend, wie es nur ein Monatsmagazin kann. PC-Welt verhilft zu den richtigen Entscheidungen. Darum sollten Sie sich für ein Jahresabonnement der PC-Welt entscheiden — für nur 80,— Mark im Jahr*. Dabei sparen Sie pro Heft 83 Pf. im Vergleich zum Kiosk-Preis. Das Entscheidende wissen. PC-Welt.

* Inlandspreis (inkl. MwSt. und Vertriebskosten)

CW PUBLIKATIONEN

Ihre starke Verbindung zur Computerwelt

Postfach 40 04 29, D-8000 München 40

WWW.HOMECOMPUTERWORLD.COM

Wer findet den Ausweg?

System: C64
Typ: Spiel
Sprache: Basic

Dieses Spielprogramm entwirft ein dreidimensionales Labyrinth. Ziel ist es, mit wenigen Schritten den Ausgang zu finden.

Zirka 30 Sekunden benötigt „Superlabyrinth" um den Irrgarten zu konstruieren. Danach präsentiert es die Gänge des Labyrinthes in dreidimensionaler Form.
Mit den Tasten:
„I" = ein Schritt nach vorne
„J" = nach links drehen
„L"= nach rechts drehen
„K"= um 180 Grad drehen

geht man dem Ziel entgegen. Mit der Taste „?" erhält man die zweidimensionale Ansicht des Labyrinthes. Ziel, Ausgangspunkt und die bisher begangene Wegstrecke sind zu erkennen. Will man den nächsten Schritt gehen, nachdem der Computer die Vogelperspektive gezeigt hat, muß man diesen blind fahren. Fairerweise sollte man auf die Mogelfunktion „?" verzichten. Bei jedem Start entwickelt das Programm per Zufallszahl andere Spielfelder. Alle Grafiken haben die Commodore-Grafikzeichen zum Ursprung und werden direkt in den Bildschirm gePOKEt.

F. Bastian

```
  1 print chr$ (147):print chr$ (5):poke 53280,0:poke 53281,0:gosub 7400:goto 90   4972
  2 rem c64 version by f. bastian                                                  
  4 rem <c> 1986                                                                   
  5 :                                                                              
 10 mz=mz+1:if mz>250 then mz=250                                                  2576
 12 poke vo,7:poke sz,33                                                           1600
 14 poke hf,mz                                                                      704
 16 poke sz,32                                                                      640
 18 return                                                                          142
 20 for xm=20 to 5 step -1                                                         1195
 22 poke vo,15                                                                      666
 24 poke sz,33                                                                      672
 26 poke hf,xm                                                                      719
 28 poke sz,32:next xm                                                             1089
 29 return                                                                          142
 30 poke vo,15                                                                      666
 32 poke sz,33                                                                      672
 34 poke hf,5                                                                       500
 36 poke sz,32                                                                      640
 38 return                                                                          142
 40 for mz=1 to 200 step 4:poke vo,7:poke sz,33:poke hf,mz:poke sz,32:next mz: return   5228
 90 pg$="suplab":nm$="13":gosub 62000                                              2770
100 h=12:v=7:h1=h+1:w=36864:v1=38912                                               3825
110 i=rnd (-ti)                                                                    1031
130 dim x1(4),n2(4),y1(4),xr(4)                                                    1972
140 for i=0 to 4:read x1(i),n2(i),y1(i),xr(i):next i                               3905
150 data 8,20,-4,31,8,12,1,27,12,6,5,24,15,2,8,22,17,0,10,21                       5281
200 cx=int ( rnd (1)*h)+1:cy=int ( rnd (1)*v)+1                                    3273
210 c=cx:r=cy:s=0                                                                  1379
220 for i=0 to (v+1)*h1:poke w+i,0:poke v1+i,0:next                                3975
230 print chr$ (147);"ich konstruiere !!!";                                        3233
300 da=0:n=0:av=v1+c+r*h1:aw=w+c+r*h1:poke aw,1:s=s+1:if s>=h*v then 900            6137
320 print mid$ ("[rvs,off]",(s and 1)+1,1);"v[left]";:gosub 10                     2989
410 if c>1 and peek (aw-1)=0 then da=da+2:n=n+1                                    3130
420 if c<h and peek (aw+1)=0 then da=da+1:n=n+1                                    3033
430 if r>1 and peek (aw-h1)=0 then da=da+8:n=n+1                                   2867
440 if (r<v and peek (aw+h1)=0) then da=da+4:n=n+1                                 2972
450 n=int ( rnd (1)*n)+1                                                           1470
460 on da+1 goto 500,850,800,600,750,610,620,630,700,640,650,660,670,680,690,695   7399
500 s=s-1                                                                          373
510 c=c+1:if c>h then c=1:r=r+1:if r>v then r=1                                    3207
520 on peek (w+c+r*h1)+1 goto 510,300                                              2637
600 on n goto 800,850                                                              729
610 on n goto 750,850                                                              801
620 on n goto 750,800                                                              796
630 on n goto 750,800,850                                                          1254
640 on n goto 700,850                                                              721
650 on n goto 700,800                                                              716
660 on n goto 700,800,850                                                          1174
670 on n goto 700,750                                                              848
680 on n goto 700,750,850                                                          1306
690 on n goto 700,750,800                                                          1226
695 on n goto 700,750,800,850                                                      1414
700 r=r-1:av=av-h1:poke av,peek (av) or 1:goto 300                                 3120
750 poke av,peek (av) or 1:r=r+1:av=av+h1:goto 300                                 2750
800 c=c-1:av=av-1:poke av,peek (av) or 2:goto 300                                  2456
850 poke av,peek (av) or 2:c=c+1:av=av+1:goto 300                                  2647
900 i=rnd (1)>.5:j=rnd (1):k=rnd (1)>.5                                            2680
920 if i then x=int (j*h)+1:y=v+(v-1)*k                                            2707
930 if i=0 then y=int (j*v)+1:x=h+(h-1)*k                                          3058
931 if (cx-x)↑2+(cy-y)↑2<(h↑2+v↑2)/9 then 900                                      3962
950 sx=x:sy=y                                                                      1188
1001 print "[clr]                                                                 299
1010 for i=w to w+(v+1)*h1:poke i,0:next                                           2762
1020 d=int ( rnd (1)*4)                                                            1304
1030 t4=ti:poke (w+x+y*h1),1                                                       2167
1045 if f=0 then gosub 3000:print "[home]";int ( sqr ((x-cx)↑2+(y-cy)↑2)*10+.5)/10   5919
1049 if cx=x and cy=y then 6000                                                    1989
1050 get q$:if q$="" goto 1050                                                     1458
1060 if q$="q" or q$="?" then k=ti:goto 6100                                       3065
1070 if q$<>"i" and q$<>"j" and q$<>"k" and q$<>"l" then goto 1050                 4021
1075 if q$="k" then d=d+2                                                          1386
1080 if q$="j" then d=d-1                                                          1346
1090 if q$="l" then d=d+1                                                          1402
1100 if d<0 then d=d+4                                                             1348
1110 if d>3 then d=d-4                                                             1365
1120 if q$<>"i" then 1045                                                          1497
1130 gosub 30                                                                      435
1200 av=v1+x+y*h1:aw=w+x+y*h1                                                      2503
```


WWW.HOMECOMPUTERWORLD.COM

```
1201 on d goto 1260,1310,1360                                    1710
1210 if y>1 and ( peek (av-h1) and 1) then                       4199
     y=y-1:poke aw-h1,1:goto 1045
1220 goto 1400                                                    575
1260 if x<h and ( peek (av) and 2) then x=                       4281
     x+1:poke aw+1,1:goto 1045
1290 goto 1400                                                    575
1310 if y<v and ( peek (av) and 1) then y=                       4769
     y+1:poke aw+h1,1:goto 1045
1340 goto 1400                                                    575
1360 if x>1 and ( peek (av-1) and 2) then                        5278
     x=x-1:poke aw-1,1:goto 1045
1400 dn=23:gosub 7000                                            1283
1410 print tab(16);"[7cbm @,7left,down,rvs]                      5666
     geht nicht[up]":gosub 20:for i=1 to
     1000:next
1420 print tab(16);"[up,11spaces,7left,                          3578
     down,7spaces,up]":goto 1049
2000 print "[clr] ";                                             437
2020 for i=1 to h:print "[3cbm @]";:next :                       2143
     print "[2down]"
2100 for j=1 to v:print "[cbm m,up,left,                         3084
     cbm m,up,left,cbm m,2down]";:for i=1
     to h
2125 k=peek (v1+i+j*h1)                                          1343
2130 print mid$ ("[2cbm @,shift @,2spaces,                       2491
     cbm m,3cbm @,3spaces]",1+3*k,3);
2140 if k<2 then print "[up,left,cbm m,up,                       1991
     left,cbm m,2down]";
2200 print "[up,left]";                                          664
2210 if peek (w+i+j*h1)=1 then print "                           2612
     [left].";
2250 if i=sx and j=sy then print "[left,                         2688
     rvs] [off]";
2260 if i=cx and j=cy then print "[left,                         2191
     cbm +]";
2270 if i=x and j=y then print "[left]";                         4059
     mid$ ("↑>v<",d+1,1);
2280 print "[right,down]";                                       610
2290 next :print "[2down]":next :return                          885
3000 s=0:print "[clr]                                            1034
3005 av=v1+x+y*h1                                                1176
3010 on d goto 3200,3400,3600                                    1532
3025 if sy=y-s and sx=x then gosub 5100                          2575
3026 if cy=y-s and cx=x then gosub 5000                          2399
3027 k=peek (av-(s+1)*h1):if k=0 or k=2                          2618
     goto 3100
3030 if peek (av-s*h1) and 2 then gosub                          2706
     4000:goto 3040
3035 gosub 4200                                                   577
3040 if peek (av-1-s*h1) and 2 then gosub                        3290
     4500:goto 3050
3045 gosub 4700                                                   597
3050 s=s+1:if s>4 goto 3900                                      1468
3060 goto 3025                                                    659
3100 if peek (av-s*h1) and 2 then gosub                          2899
     4300:goto 3110
3105 gosub 4800                                                   601
3110 if peek (av-1-s*h1) and 2 then gosub                        3281
     4400:goto 3120
3115 gosub 4900                                                   605
3120 return                                                       142
3200 if cy=y and cx=x+s then gosub 5000                          2391
3210 if sy=y and sx=x+s then gosub 5100                          2555
3226 k=peek (av+s):if k=0 or k=1 goto 3300                       1900
3230 if peek (av+s)=2 then gosub 4200:goto                       2204
     3240
3235 gosub 4000                                                   569
3240 if peek (av+s-h1) and 1 then gosub                          3046
     4500:goto 3250
3245 gosub 4700                                                   597
3250 s=s+1:if s>4 goto 3900                                      1468
3260 goto 3200                                                    571
3300 if peek (av+s)=0 then gosub 4800:goto                       2097
     3310
3305 gosub 4300                                                   581
3310 if peek (av+s-h1) and 1 then gosub                          2980
     4400:goto 3320
3315 gosub 4900                                                   605
3320 return                                                       142
3400 if cy=y+s and cx=x then gosub 5000                          2367
3410 if sy=y+s and sx=x then gosub 5100                          2543
3426 k=peek (av+s*h1):if k=0 or k=2 then                         3183
     goto 3500
3430 if peek (av-1+s*h1) and 2 then gosub                        3144
     4000:goto 3440
3435 gosub 4200                                                   577
3440 if peek (av+s*h1)=1 then gosub 4700:                        2818
     goto 3450
3445 gosub 4500                                                   589
3450 s=s+1:if s>4 goto 3900                                      1468
3460 goto 3400                                                    579
3500 if peek (av-1+s*h1) and 2 then gosub                        3151
     4300:goto 3510
3505 gosub 4800                                                   601
3510 if peek (av+s*h1)=0 then gosub 4900:                        2739
     goto 3520
3515 gosub 4400                                                   585
3520 return                                                       142
3600 if cx=x-s and cy=y then gosub 5000                          2389
3610 if sx=x-s and sy=y then gosub 5100                          2565
3625 k=peek (av-(s+1)):if k=0 or k=1 then                        3323
     goto 3700
3630 if peek (av-s-h1) and 1 then gosub                          2946
     4000:goto 3640
3635 gosub 4200                                                   577
3640 if peek (av-s) and 1 then gosub 4500:                       2375
     goto 3650
3645 gosub 4700                                                   597
3650 s=s+1:if s>4 then 3900                                      1588
3660 goto 3600                                                    587
3700 if peek (av-s-h1) and 1 then gosub                          3139
     4300:goto 3710
3705 gosub 4800                                                   601
3710 if peek (av-s) and 1 then gosub 4400:                       2303
     goto 3720
3715 gosub 4900                                                   605
3720 return                                                       142
3900 dn=11:gosub 7000:print tab(19)"MN                           3457
     [down,2left]NM[home]":return
4000 dn=y1(s):gosub 7000                                         1632
4015 if s=0 then print tab(xr(s));"[cbm @]"                      1898
4020 if s>0 and s<4 then for i=1 to 4-s:                         3742
     print tab(xr(s)+5-s)"[cbm g]":next
4045 if s>0 then print tab(xr(s)) mid$ ("                        2454
     [4cbm @,cbm g]",s)
4070 if s=4 then print tab(xr(s))"[2cbm g,                       2355
     down,2left,2cbm g]"
4071 q$="":if s>0 then q$=mid$ ("[3right,                        2610
     cbm g]",s)
4072 if s<4 then for i=0 to n2(s)+1:print                        2961
     tab(xr(s))"[cbm g]"q$:next
4075 if s>0 then print tab(xr(s)) mid$ ("                        2439
     [4cbm t,cbm g]",s)
4095 if s>0 and s<4 then for i=1 to 4-s:                         3742
     print tab(xr(s)+5-s)"[cbm g]":next
4096 if s=0 then print tab(xr(s))"[cbm t]"                       1860
4098 print "[home]                                               297
4099 return                                                       142
4200 dn=y1(s):gosub 7000                                         1632
4215 if s=0 then print tab(xr(s))"N":goto                        2294
     4230
4220 for i=1 to 5-s:print tab(xr(s)+5-s-                         3552
     i)"N":next
4230 dn=n2(s)+2:gosub 7100                                       1739
4235 if s=0 then print tab(xr(s))"M":goto                        2279
     4249
4240 for i=0 to 4-s:print tab(xr(s)+i)"M":                       3056
     next
4249 print "[home]                                               297
4250 return                                                       142
4300 j=3-s:if s=0 then j=0                                       1554
4305 print "[home]";                                             493
4306 if s>0 goto 4328                                            1203
4307 print tab(20);:print "[down]";:for i=                       4018
     1 to 18:print "[cbm t]";:next :print "
     [cbm t]"
4310 dn=n2(0):gosub 7100                                         1191
4320 print tab(20);:for i=1 to 19:print "                        3563
     [cbm @]";:next :goto 4398
4328 dn=y1(s):gosub 7100                                         1640
4329 for i=1 to 5-s:print tab(j+xr(s)+2)"                        2818
     [cbm g]":next
4330 print tab(20);:for i=19 to j+xr(s):                         3937
     print "[cbm t]";:next :print "[cbm g]"
4370 if s<4 then for i=1 to n2(s):print                          3320
     tab(j+xr(s)+2)"[cbm g]":next
4380 print tab(20);:for i=19 to j+xr(s):                         3810
```


WWW.HOMECOMPUTERWORLD.COM

```
      print "[cbm @]";:next :print "[cbm g]"
4385 for i=1 to 5-s:print tab(j+xr(s)+2)"    2818
      [cbm g]":next
4398 print "[home]                           297
4399 return                                  142
4400 print "[home]";                        493
4406 if s>0 goto 4430                       1132
4407 print "[down]";:for i=1 to 11+xl(0):   3382
      print "[cbm t]";:next :print "[cbm t]"
4410 dn=n2(0):gosub 7100                    1191
4420 for i=1 to 12+xl(0):print "[cbm @]";:  3117
      next :goto 4398
4430 dn=yl(s):gosub 7100                    1640
4431 for i=1 to 5-s:print tab(xl(s))"       2790
      [cbm m]":next
4432 print tab(xl(s));:print "[cbm m]";:for 4853
      i=xl(s) to 17:print "[cbm t]";:next :
      print "[cbm t]"
4470 if s<4 then for i=1 to n2(s):print     3051
      tab(xl(s))"[cbm m]":next
4480 print tab(xl(s));:print "[cbm m]";:for 4471
      i=xl(s) to 17:print "[cbm @]";:next :
      print "[cbm @]"
4485 for i=1 to 5-s:print tab(xl(s))"       2790
      [cbm m]":next
4498 print "[home]                          297
4499 return                                  142
4500 dn=yl(s):gosub 7000                    1632
4515 if s=0 then print tab(xl(s));"[cbm @]" 1892
4520 if s>0 and s<4 then for i=1 to 4-s:    3363
      print tab(xl(s))"[cbm m]":next
4545 if s>0 then print tab(xl(s)) left$ ("  2689
      [cbm m,4cbm @]",6-s)
4565 if s=4 then print tab(xl(s))"[2cbm m,  2616
      2left,down,2cbm m]"
4570 q$="":if s>0 then q$=mid$ ("[3right,   2642
      cbm m]",s)
4572 if s<4 then for i=0 to n2(s)+1:print   3040
      tab(xl(s))"[cbm m]"q$:next
4575 if s>0 then print tab(xl(s)) left$ ("  2659
      [cbm m,4cbm t]",6-s)
4595 if s>0 and s<4 then for i=1 to 4-s:    3363
      print tab(xl(s))"[cbm m]":next
4596 if s=0 then print tab(xl(s))"[cbm t]"  1854
4597 print "[home]                          297
4599 return                                  142
4700 dn=yl(s):gosub 7000                    1632
4715 if s=0 then print tab(xl(s));"M":goto  2584
      4730
4720 for i=1 to 5-s:print tab(xl(s)+i)"M":  3255
      next
4730 dn=n2(s)+2:gosub 7100                  1739
4735 if s=0 then print tab(xl(s))"N":goto   2377
      4749
4740 for i=0 to 4-s:print tab(xl(s)+5-s-    3671
      i)"N":next
4749 print "[home]"                         314
4750 return                                  142
4800 gosub 4200                              577
4815 dn=yl(s)+5-s:gosub 7000                2506
4820 print tab(20);                          802
4825 if s<4 then for i=1 to xr(s)-21:print  2991
      "[cbm t]";:next
4830 print "P[left,down]";                   815
4835 if s<4 then for i=1 to n2(s):print "   2841
      [cbm m,down,left]";:next
4840 print "[shift @,2left]";               1182
4850 if s<4 then for i=1 to xr(s)-20-1:     3602
      print "[cbm @,2left]";:next
4851 print "[home]                          297
4855 return                                  142
4900 if s=0 then j=5:goto 4910              1668
4902 j=0                                     367
4910 gosub 4700                              597
4915 dn=yl(s)+5-s:gosub 7000                2506
4920 print tab(19);                          870
4925 if s<4 then for i=1 to 18-xl(s)-5+s+j: 3777
      print "[cbm t,2left]";:next
4930 print "O[left,down]";                   811
4935 if s<4 then for i=1 to n2(s):print "   2833
      [cbm g,down,left]";:next
4940 print "L";                             468
4950 if s<4 then for i=1 to 18-xl(s)-5+s+j: 3377
      print "[cbm @]";:next
4951 print "[home]                          297
4955 return                                  142
5000 if s=0 or s=4 then return              1147
5010 dn=18:gosub 7000                       1355
5020 if s=3 then print tab(20);"[3up].      1671
5030 if s=2 then print tab(19);"[rvs,up,    2978
      shift £,off,shift £,2left,down,cbm +]N
      [up,cbm g]
5040 if s=1 then print tab(18);"[down]      5891
      [3cbm @,4left,down]N[2spaces]N[cbm g,
      5left,down,3cbm +] [cbm g,5left,down,
      3cbm +]N
5050 return                                  142
5100 if s=0 or s=4 then return              1147
5110 dn=20:gosub 7000                       1235
5120 if s=3 then print tab(19);"[5up,rvs,   2935
      2cbm i,up,2left,off,2cbm i]
5130 if s=2 then print tab(16);"[3up,rvs,   2397
      shift £,6spaces,cbm *]
5141 if s=1 then print tab(13);"[rvs,       2175
      shift £,12spaces,cbm *]
5142 if s=1 then print tab(12);"[rvs,       2429
      shift £,14spaces,cbm *]
5150 return                                  142
6000 k=ti:for i=1 to 7:poke 245,22:print :  3928
      print tab(15)"super !!!!"
6005 gosub 40                                437
6010 poke 245,22:print :print tab(15)"      3255
      [10spaces]":next i:manni=1
6100 gosub 2000                              565
6110 print tab(10);"[up]ein anderes? ";     2629
6115 sw=1:tm=0                              1117
6120 get t$:if t$<>"" then 6200             1877
6130 if ti>tm then print mid$ ("? ",sw,1);" 5016
      [left]";:tm=ti+15:sw=3-sw
6140 goto 6120                               589
6200 if t$="j" or t$=cr$ then print "ja":   2369
      run
6205 if t$="n" and manni=1 then end         2213
6210 if t$="n" then print chr$ (147)"fahren 6409
      sie nun einen schritt blind !":goto
      1050
6220 goto 6120                               589
7000 print "[home]";                        493
7100 if dn>0 then for z=1 to dn:print "     2305
      [down]";:next
7110 return                                  142
7400 rem sid initialisieren
7410 sm=54272:hf=sm+1:vo=sm+24:sz=sm+4      3576
7420 for i=sm to sm+24:poke i,0:next         2316
7425 poke sm+5,8:poke sm+5,8                1399
7430 return                                  142
62000 print "[clr,down]";tab(9);"programmnr. 3463
      ";nm$;"[2spaces]";pg$
62010 print chr$ (147);" copyright (c) 1986 4185
      [2spaces]by f. bastian"
62020 for i=1 to 10:print "[4shift *]";:next 1389
62030 print "[down]suchen sie einen ausgang  3222
      aus dem"
62031 print :print "[12spaces,rvs]           2310
      superlabyrinth":print
62032 print "[down]ihre steuerungen:"        1899
62034 print "[down,2spaces]i = ein schritt   2794
      vorwaerts"
62035 print "[down,2spaces]j = nach links    1995
      drehen"
62036 print "[down,2spaces]l = nach rechts   2438
      drehen"
62038 print "[down,2spaces]k = um 180 grad   1879
      drehen"
62040 print "[down,2spaces]? = hilfe !"      1454
62049 print "[3down,rvs]return[off]          2109
      druecken"
62050 get t$:if t$="" then 62050             1681
62060 qk=198:cr$=chr$ (13)                   1965
62070 print "[clr]":goto 100                  676
```

Super ist, wer das Labyrinth in 20 Schritten durchquert

WWW.HOMECOMPUTERWORLD.COM

Mit High-Speed über's Spielfeld

System: C16/116/Plus4
Typ: Action-Spiel
Sprache: Basic

„Speedy" überzeugt durch Geschwindigkeit und Action. Schnelle Reaktionen und gute Nerven sind gefordert.

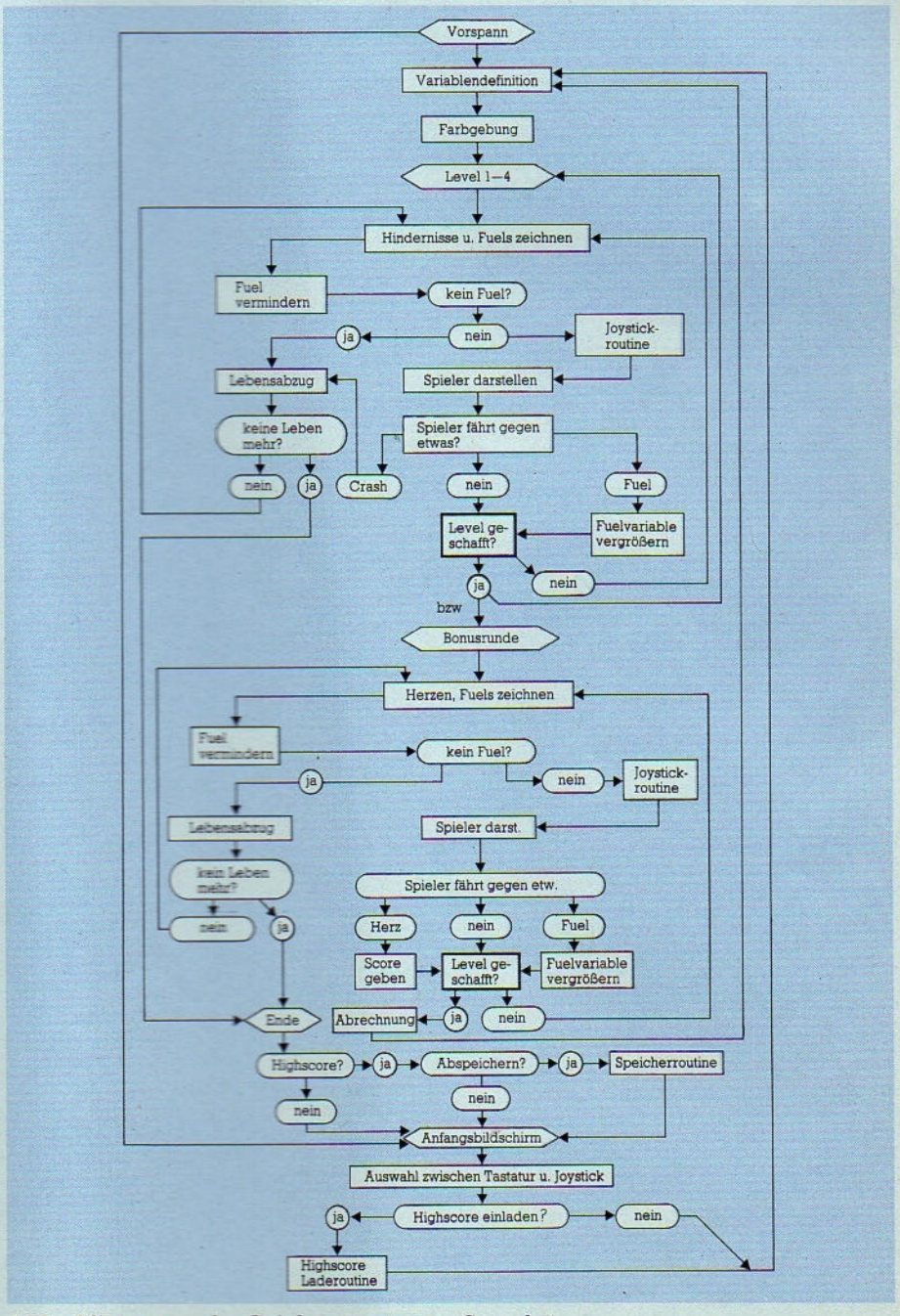
„Speedy" ist ein schnelles Hindernisrennen. Über fünf Runden muß der Spieler extreme Schwierigkeiten meistern.
Nach Start mit RUN fragt der Computer, ob man einen gespeicherten High-Score laden will. Danach entscheidet man: Joystick oder Tastatur. Im ersten Level muß der Spieler, durch eine Schlange von Kreuzen dargestellt, den entgegenkommenden Hindernissen ausweichen, die ihm auf dem ganzen Bildschirm entgegenfliegen.

Jetzt geht's ab

Im zweiten Level muß er auf einer engen Straße versuchen, den Hindernissen auszuweichen, ohne im Straßengraben zu landen. Das nächste Level, eine enge Kurvenstrecke, ist ohne Hindernisse. Hat man den vierten Level, überstanden, wird man mit einer Bonusrunde belohnt, in der man so viele Herzen wie möglich fressen muß. In keinem Level darf das Tanken vergessen werden. Gefährlich ist das in der Bonusrunde, wenn der Spieler die vorbeisausenden Tankstellen nicht beachtet und ohne Sprit in den Pampas sitzt. Ist alles geschafft, erscheinen Punkteaufstellung, gefahrene Kilometer und verbliebene Fahrzeuge. Auf Knopfdruck geht's von vorne los, allerdings mit verkürzten Reaktionszeiten und somit schwieriger. Nach jedem Level erhält man zwei weitere Fahrzeuge.
Unterroutinen des Programmes:

160 Initialisierung
290 Ebene I, Level I
400 Steuerung (Joystick/Tastatur)
530 Abfrage der Tankreserve
650 Spieler getroffen/Ende?
750 Ebene II, Level II
930 Ebene III, Level III
1160 Ebene IV, Level IV
1400 Ebene V, Level V
1560 Ziel, Level erhöhen
1730 Spielende erreicht
1860 Highscore speichern
1960 Copyright-Anfangsbild
2070 Eingabegerät-Abfrage
2170 Highscore laden

M. Goes



Ablaufdiagramm des Spielprogrammes „Speedy"

WWW.HOMECOMPUTERWORLD.COM

```
10 rem " U------------------------------I
20 rem " |     s p e e d y              |
30 rem " |                              |
40 rem " |           von                |
50 rem " |                              |
60 rem " |        martin goes           |
70 rem " |     philosophenweg 22        |
80 rem " |      7400 tuebingen          |
90 rem " |                              |
100 rem "J------------------------------K
110 :
120 :
130 :
140 :
150 :
160 color 0,1                                    640
170 gosub 1960:l=5:sc=0:op=0                     1829
180 i=0:s=0:color 1,2:f=200:y=1024               2713
190 scnclr :a=3092+160+op:color 0,1              2367
200 :
210 :
220 :
230 :
240 rem "   U----------------I
250 rem "   | erste ebene    |
260 rem "   J----------------K
270 :
280 :
290 for i=1 to 1000:char 1,0+int ( rnd           4223
    (0)*40),24,"[yel]Q[down]":gosub 400
300 if int ( rnd (0)*9)=0 then t=4032+int        4749
    ( rnd (0)*40):poke t-y,188:poke t,134
310 poke a,86:poke a-y,2:if peek (a+40)<>        4519
    32 or f<=0 then gosub 530
320 f=f-1:next :goto 750                         1156
330 :
340 :
350 rem    ***********************
360 rem    *  joystick/tastatur  *
370 rem    ***********************
380 :
390 :
400 on jt goto 410,420                            933
410 get a$:if a$="" then return :else 430        1905
420 if joy (1)=0 then return :else if joy        2712
    (1)<>0 then 440
430 if a$="[right]" then m=1:goto 450:else       5206
    if a$="[left]" then m=-1:goto 450:else
    m=0:goto 450
440 if joy (1)=7 then m=-1:else if joy           3863
    (1)=3 then m=1:else m=0
450 a=a+m:return                                  702
460 :
470 :
480 rem    ************************
490 rem    *  reaktion / kein fuel  *
500 rem    ************************
510 :
520 :
530 if f=<0 then char 1,26,24,"[rvs,red]         3793
     low on fuel [off]":goto 650
540 if peek (a+40)=102 then a=a-m:goto 650       2923
550 if peek (a+40)=134 then f=f+50:char 1,       4978
    0,24,"f:":print f"[up]":sound 1,500,
    10:return
560 if peek (a+40)=65 or peek (a+40)=81 or       3962
    peek (a+40)=113 then 650
570 return                                        142
580 :
590 :
600 rem    ************************
610 rem    *  lebensabzug (ende ?)  *
620 rem    ************************
630 :
640 :
650 for o=900 to 100 step -25:sound 3,o,3:       2326
    next
660 l=l-1:if l=0 then 1730:else f=200:char       4457
    1,0,24,"l:":print l"[up]":return
670 :
680 :
690 :
700 rem "   U----------------I
710 rem "   | zweite ebene   |
720 rem "   J----------------K
730 :
740 :
750 f=200:l=l+2:b=15:for i=1 to 25:char 1,       4361
    b,24,"[wht,shift -,cbm +,7spaces,
    cbm +,shift -,down]":next
760 c=25:a=3092+160+op                           2065
770 for i=1000 to 2000:char 1,b,24,"[wht,        3380
    shift -,cbm +,7spaces,cbm +,shift -,
    down]"
780 if int ( rnd (0)*4)+1=2 then char 1,b+       4070
    2+int ( rnd (0)*7),23,"[grn]A[wht]"
790 if int ( rnd (0)*14)+1=2 then char 1,        4972
    b+2+int ( rnd (0)*7),23,"[cbm 1,rvs,
    flashon]f[flashoff,off,wht]"
800 poke a,86                                     749
810 gosub 400                                     566
820 if peek (a+40)<>32 or f<=0 then gosub        2476
    530
830 f=f-1:next :goto 930                         1152
840 :
850 :
860 :
870 :
880 rem "   U----------------I
890 rem "   | dritte ebene   |
900 rem "   J----------------K
910 :
920 :
930 b=15:a=3092+160+op:color 0,9:color 1,        3552
    7,2
940 gosub 950:goto 960                            961
950 for i=1 to 25:char 1,b,24,"[shift -,         3661
    cbm +,cbm 7,rvs,7spaces,off,cbm +,
    shift -,down]":next :return
960 l=l+2:f=200:for u=2000 to 3000                2055
970 v=int ( rnd (0)*2)+1:if v=1 then b=b+        4534
    1:else if v=2 then b=b-1:else print v
980 if b<1 then b=b+1:else if b>26 then b=       2596
    b-1
990 char 1,b,24,"[shift -,cbm +,rvs,             4695
    7spaces,off,cbm +,shift -,down]":poke
    a,214:if int ( rnd (0)*17)=1 then
    gosub 1070
1000 gosub 400                                    566
1010 if peek (a+40)<>160 or f=0 then gosub       2251
    1030
1020 f=f-1:next :goto 1160                       1179
1030 if f=0 then 1040:else if peek (a+40)=       3123
    134 then gosub 530:return
1040 gosub 1050:l=l-1:if l=0 then end :else      4756
    char 1,0,24,"l:":print l"[up]":gosub
    950:goto 1060
1050 for o=900 to 100 step -25:sound 3,o,3:      2497
    next :return
1060 a=3072+160+op+b+3:return                    1992
1070 char 1,b+int ( rnd (0)*8)+2,24,"[up,        3288
    cbm 1,rvs,flashon]f[off,flashoff,down,
    cbm 7]":return
1080 :
1090 :
1100 :
1110 rem "   U----------------I
1120 rem "   | vierte ebene   |
1130 rem "   J----------------K
1140 :
1150 :
1160 b=15:scnclr :color 0,3,5:l=l+2:for i=       5278
    1 to 25:char 1,b,24,"[cbm 7,shift -,
    cbm +,7spaces,cbm +,shift -,down]":
    next
1170 c=25:f=200                                  1241
1180 a=3092+160+op:b=15:for i=3000 to 3500:      4995
    char 1,b,24,"[cbm 7,shift -,cbm +,
    7spaces,cbm +,shift -,down]"
1190 if int ( rnd (0)*20)+1=2 then 1200:         2839
    else 1210
1200 char 1,b+2,23,"[wht,7cbm e,cbm 7]":         5156
```


WWW.HOMECOMPUTERWORLD.COM

```
      char 1,b+2+int ( rnd (0)*7),23," ":
      gosub 1260
1210  if int ( rnd (0)*14)+1=2 then char 1,    4972
      b+2+int ( rnd (0)*7),23,"[cbm 1,rvs,
      flashon]f[flashoff,off,wht]"
1220  poke a,86:poke a-1024,2                  2155
1230  gosub 400                                 566
1240  if peek (a+40)<>32 or f<-0 then gosub    2476
      530
1250  f=f-1:next :goto 1400                    1143
1260  poke a,86:poke a-1024,2                  2155
1270  if peek (a+40)<>32 or f<-0 then gosub    2476
      530
1280  for u=1 to 5                              878
1290  char 1,b,24,"[cbm 7,shift -,cbm +,       4551
      7spaces,cbm +,shift -,down]":poke a,
      86:poke a-1024,2:gosub 400
1300  if peek (a+40)<>32 or f<-0 then gosub    2476
      530
1310  f=f-1:next :return                        710
1320  :
1330  :
1340  :
1350  rem "   U--------------------I
1360  rem "   I  fuenfte ebene    I
1370  rem "   J--------------------K
1380  :
1390  :
1400  f=200:a=3092+160+op:vol 8:l=l+2:color    6603
      0,1:char 1,15,21,"[5down,rvs,wht]
       bonus round [off]"
1410  for x=1 to 500:char 1,0+int ( rnd (0)*  4157
      40),24,"[yel]S[down]":gosub 400
1420  if int ( rnd (0)*9)=0 then t=4032+int    4749
      ( rnd (0)*40):poke t-y,188:poke t,134
1430  poke a,86:poke a-y,2:if peek (a+40)<>    4666
      32 or f<-0 then gosub 1450
1440  f=f-1:next :goto 1510                    1155
1450  if peek (a+40)=83 then s=s+10:sound 1,   3952
      900,10:else gosub 530
1460  return                                    142
1470  :
1480  :
1490  :
1500  :
1510  rem "  U----------------------I
1520  rem "  I  ziel erreicht !!!!!  I
1530  rem "  J----------------------K
1540  :
1550  :
1560  scnclr :for g=1 to 15:get a$             1065
1570  color 0,1:print "[rvs,red,6spaces]sie    3922
       haben das ziel erreicht[7spaces]"
1580  print :print "[3down,3right,grn]          1587
       score:"
1590  print "[4spaces,5cbm t]"                  1190
1600  print "[pur]"i"-meter "                   1283
1610  print "[yel]"s"-bonus round":print "      4646
      [cbm 1]"sc"-score der vorrunde"
1620  sc=sc+(s+i):print "[blu,2down,2right]    3737
      gesamt-score:"sc
1630  print "[rvs,2down,wht] die naechste       4123
       runde wird etwas schwerer "
1640  print "[2down,right,cbm 8,7spaces]        3704
      druecken sie 'space'[9spaces]"
1650  get key a$:if a$<>" " then 1650:else     3742
      op=op+120:goto 180
1660  :
1670  :
1680  rem   ***************
1690  rem   *  spielende  *
1700  rem   ***************
1710  :
1720  :
1730  color 0,1:scnclr :sc=sc+s+i:print "      5142
      [rvs,pur,9spaces]das spiel ist zu ende
      [10spaces]"
1740  print "[3down,2right,wht]score:"sc:for   2920
      t=1 to 10:get a$:next
1750  if sc>hs then hs=sc:input "[cyn,2down]    4492
       wie heissen sie";n$
1760  if n$="" then n$="---"                    1851
1770  print "[cbm 1,2down]highscore:"n$,hs      1816
1780  print "[cbm 4,2down] highscore saven      5289
       ?(j/n)":get key a$:if a$="j" then
      1860:else 1880
1790  :
1800  :
1810  rem   ********************
1820  rem   *  highscore saven  *
1830  rem   ********************
1840  :
1850  :
1860  open 1,1,1,"highscore":print#1,n$:       3343
      print#1,hs:close 1
1870  :
1880  print :print "[down,right,cbm 3,          4408
      7spaces]druecken sie eine taste
      [6spaces]":get key a$:goto 170
1890  :
1900  :
1910  rem   ******************
1920  rem   *   spielanfang   *
1930  rem   ******************
1940  :
1950  :
1960  vol 8:print "[clr,rvs,red,11spaces]s     2682
      [2spaces]p[2spaces]e[2spaces]e
      [2spaces]d[2spaces]y[13spaces]"
1970  data b,y ,m,a,r,t,i,n ,g,o,e,s            3032
1980  print "[3down,yel]"                        789
1990  restore :for e=1 to 12:read u$:print      4339
      u$;:sound 3,900,4:for t=1 to 170:next
      :next
2000  :
2010  :
2020  rem   ********************************
      **
2030  rem   * spiel mit joystick o. tastatur
       *
2040  rem   ********************************
      **
2050  :
2060  :
2070  print :print :print :print "[cbm 6]      3897
      druecken sie 'k' fuer keyboard"
2080  print "[13spaces]'j' fuer joystick"       2517
2090  get key a$:if a$="k" then jt=1:else       2058
      jt=2
2100  :
2110  :
2120  rem   ***********************
2130  rem   *  highscore einladen  *
2140  rem   ***********************
2150  :
2160  :
2170  print "[2down,cbm 1] highscore            4727
       einladen ?(j/n)":get key a$:if a$<>
      "j" then return
2180  open 1,1,0,"highscore":input#1,n$:       2840
      input#1,hs
2190  print :print "[down,right,cbm 3,          4170
      7spaces]druecken sie eine taste
      [6spaces]":get key a$:return
```

Mit Überschallgeschwindigkeit durch den Computer

WWW.HOMECOMPUTERWORLD.COM

Sternbilder im Computer

System: C128
Typ: Lernspiel
Sprache: Basic

Sternbilder vom Himmel auf den Bildschirm holen: Das macht „Starpic". Dabei stehen 30 Konstellationen zur Auswahl.

In klaren Nächten sind sie besonders gut zu erkennen: Sternbilder. Unzählige Sagen und Mythen über die Figuren am Himmelszelt sind bekannt. So wurde in Griechenland gerne die Geschichte Cassiopeias, Königin Äthiopiens, erzählt. Sie behauptete, schöner zu sein als die Göttin der Schönheit, und wurde deshalb in den Sternenhimmel verbannt. Heutzutage erfreuen sich die Geschichten immer noch großer Popularität; doch nur wenige wissen die einzelnen Sternanhäufungen richtig einzuordnen. „Starpic" verfügt über Koordinaten von 30 Sternbildern und zeichnet diese am Bildschirm. Ein Konstellationenquiz bringt spielend Lernerfolg.

Sternbilder:

Leier —	Widder
Zwillinge —	Stier
Kepheus —	Schütze
Drache —	Wassermann
Waage —	Kleiner Wagen
Skorpion —	Herkules
Delphin —	Großer Hund
Fuhrmann —	Löwe
Fische —	Orion
Pegasus —	Schwan
Jungfrau —	Großer Wagen
Adler —	Cassiopeia
Krebs —	Steinbock
Pfeil —	Nördliche Krone
Perseus —	Hase

Mit Taste „4" beendet man das Programm.

softteam

```
  5 print chr$ (14)                                        697
 10 rem starpic
 60 hl=2331:pl=2325:fa=2322                               2443
 70 fc=253:bc=252                                         1483
 80 graphic 0,1:color 0,1:print "[wht]"                   1512
100 dim x(25),y(25),x$(80)                                1893
120 rem "Sternenkonstellationen erraten
130 rem
131 print "[clr]"                                          316
140 gosub 2551                                            641
160 graphic 0,1:color 0,1:print "[wht]"                   1512
161 print "Dieses Programm soll Ihnen                      3758
     helfen, die"
180 print "Sternenbilder am Himmel zu                     4436
     erkennen."
200 gosub 10000                                           569
210 print                                                 153
220 print "Geben Sie nur den Namen"                       2533
240 print "des Sternbildes ein: z.B.                      3106
     Stier."
260 print "Bitte waehlen Sie 1, 2, 3 oder                 3157
     4."
280 print "1 - Eine bestimmte                             4008
     Konstellation sehen."
300 print "2 - Ein Zufallskonstellationen                 3898
     - Quiz."
320 print "3 - Ueberblick"                                1858
330 print "4 - Ende":print                                1500
335 poke 208,0                                            836
340 print "Ihre Wahl: ";:get z$:if z$=""                  5208
    or val (z$)<1 then print :print "[2up]
    ":goto 340
345 z=val (z$)                                            650
350 if z>4 then print "[clr]":goto 160                    1618
355 if z=4 then 25000                                     1119
360 if z=1 then 2180                                      1002
380 print :print "Mit Hinweisen (J/N)? ";                 3190
385 poke 208,0                                            836
390 get hi$:if hi$<>"j" and hi$<>"n" then                 3360
    goto 390
395 print "[clr]"                                          316
400 if z=2 then 2380                                      1138
410 if z=3 then char ,1,1,"um in das menue                 6380
     zurueckzukommen[15spaces]bitte ' ← '
     druecken."
411 print chr$ (14)                                       697
415 if z=3 then for i=1 to 1000:next i                    1751
420 rem "read Sternenanzahl
440 read n                                                291
460 if n=0 then print "Keine                              6317
     Konstellationen mehr erreichbar.":
    restore :for i=1 to 999:next :goto 160
480 for i=1 to n:read x(i),y(i):next i                    2156
500 rem y-trapert invertieren
520 for i=1 to n:y(i)=-y(i):next i                        2346
540 rem high x und high y finden
550 graphic 1,1:color 1,2:print "[clr]":                  1698
    rem "Hintergrund schwarz
560 x9=x(1):y9=y1:x1=x(1):y1=y(1)                         3645
580 for k=1 to n                                          893
600 if x(k)>x9 then x9=x(k)                               2159
620 if x(k)<x1 then x1=x(k)                               2187
640 if y(k)>y9 then y9=y(k)                               2243
660 if y(k)<y1 then y1=y(k)                               2271
680 next k                                                280
700 rem werte uebersetzen um negative zahl
    en zu verhindern
720 for i=1 to n:x(i)=x(i)-x1:y(i)=y(i)-                  4269
    y1:next i
740 rem ausmass ohne verzerrung errechnen
760 s1=39/(x9-x1)                                         1442
780 s2=38/(y9-y1)                                         1560
800 if s1>0.75*s2 then s1=0.75*s2:goto 840                3369
820 s2=s1/0.75                                            935
840 rem gr
860 rem punkte zeichnen
880 for i=1 to n                                          889
900 x(i)=x(i)*s1:y(i)=y(i)*s2                             1985
920 rem
930 for i1=10 to 11:for i2=30 to 31                       1894
940 draw 1,x(i)*299/39+i1,y(i)*164/39+i2                  3468
960 next i2,i1,i                                          934
980 if z=3 then read b$:read h$:char ,0,0,                3934
    b$:if hi$="j" then char ,0,24,h$
981 poke 208,0                                            836
982 if z=3 then get a$:if a$="" then 982                  1892
983 if z=3 and a$="←" then goto 160                       1838
984 if z=3 then goto 440                                  932
1000 if z=1 then gosub 9000:goto 2180                     1622
1020 if z<>2 then read h$                                 1242
1040 print :print :print                                  733
1060 rem speed=0
1070 char ,0,0,"benennen sie diese                        4238
     konstellation:"
1080 if hi$<>"j" then 1130                                1142
1100 char ,1,24,h$                                        857
1110 print chr$ (14);                                     903
1130 gosub 11000:y$=xx$:gosub 20000:c$=y$                 2567
```


WWW.HOMECOMPUTERWORLD.COM

```
1140 print :print                                                    371
1150 y$=b$:gosub 20000:b1$=y$                                       1936
1160 if b1$=c$ then 1200                                            1337
1180 goto 1260                                                      615
1200 char ,1,24,"richtig ![3spaces]weiter                           3681
      so !!![13spaces]"
1210 print chr$ (14)                                                697
1220 for i=1 to 1000:next i                                         1010
1230 gosub 9000                                                     579
1240 on z goto 2180,2380,440                                        1438
1260 char ,1,23,"leider falsch ":char ,1,                           5244
      24,"die konstellation war: "+b$
1270 print chr$ (14);                                               903
1280 for i=1 to 1300:next i                                         1202
1290 print "[clr]" chr$ (14):char ,1,23,                           4492
      "fuer das menue ' ← ' druecken."
1292 char ,1,24,"eine andere taste fuehrt                           3311
      weiter."
1293 print chr$ (14);                                               903
1295 poke 208,0                                                     836
1297 get a$:if a$="" then 1297                                      1642
1298 if asc (a$)=95 then print "[clr]":goto                         1944
      160
1299 print "[clr]"                                                  316
1300 on z goto 2180,2380,440                                        1438
1305 data 6,0,22,7,12,7,27,8,20,20,0,20,10,                         3735
      "leier"
1310 data "ihr hellster stern ist die wega"                         2891
1320 data 6,46,209,17,177,64,102,149,117,                           6379
      198,90,159,29,"zwillinge"
1340 data "sie beinhaltet castor und                                3215
      pollux"
1345 data 5,0,0,0,43,33,5,37,45,28,92,                              3963
      "kepheus"
1350 data "cassiopeia's ehemann"                                    2123
1360 data 14,15,59,69,163,157,235,233,192,                          8067
      266,138,254,69,283,37,354,39,377
1362 data 45,412,115,362,178,345,210,357,                           5075
      243,400,226,"drache"
1380 data "der gegner aller ritter"                                 2289
1385 data 6,0,0,5,10,16,75,40,97,43,20,65,                          4881
      65,"waage"
1390 data "sie gehoert zu den                                       3777
      tierkreiszeichen"
1400 data 13,60,80,44,58,35,37,63,16,112,                           7291
      17,148,29,150,72,154,116,177,177,189
1402 data 198,263,200,260,234,243,272,                              5152
      "skorpion"
1420 data "der rote gigant antares ist in                           3090
      ihm"
1425 data 5,0,3,8,0,30,0,8,10,14,7,                                 3207
      "delphin"
1430 data "gelehriges meeressaeugetier"                             2805
1440 data 5,81,7,155,33,147,175,70,189,42,                          5462
      113,"fuhrmann"
1460 data "auch wagenlenker genannt"                                2837
1465 data 21,0,0,8,0,11,9,12,2,23,3,24,19,                          5659
      32,5,33,37,38,3,38,32,38
1467 data 41,43,27,49,32,52,0,80,8,90,0,90,                         7452
      9,95,12,98,2,105,10,114,14,"fische"
1470 data "schwimmt im himmelsmeer"                                 2726
1480 data 8,67,98,73,243,225,248,237,111,                           7401
      405,0,303,63,479,-24,455,62,"pegasus"
1500 data "man beachte ein grosses viereck"                         3421
1505 data 11,0,102,20,80,22,126,42,22,52,                           6754
      90,70,60,112,20,112,50,117,80,137,9
1507 data 172,0,"jungfrau"                                          1639
1510 data "spica ist ihr einziger heller                            3917
      stern"
1520 data 7,0,0,0,59,8,83,18,107,113,149,                           5146
      52,76,77,31,"adler"
1540 data "ihr auffaelligster stern heisst                          3592
      altair"
1545 data 7,0,10,8,17,8,37,10,21,11,28,22,                          4977
      0,22,17,"krebs"
1550 data "bei ihr liegt ein offener                                3598
      sternhaufen"
1560 data 4,85,205,375,165,545,45,585,125,                          4187
      "pfeil"
1580 data "klein und liegt neben dem adler"                         2731
1585 data 6,0,3,5,0,14,3,37,10,47,0,47,10,                          4126
      "widder"
1590 data "liegt relativ parallel zur                               3471
      ekliptik"
1600 data 9,11,196,92,230,82,49,205,40,217,                         7548
      34,228,34,229,26,242,26,239,34,"stier"
1620 data "besteht aus dem schoenen                                 3701
      siebengestirn"
1640 data 7,10,35,43,0,45,43,65,90,105,73,                          6421
      140,77,143,41,"schuetze"
1660 data "die mitte der galaxis ist dort"                          2807
1665 data 18,0,52,8,4,13,2,19,39,22,4,22,                           6810
      43,27,65,33,41,41,50,43,69,45
1667 data 10,47,58,50,80,54,0,54,9,57,82,                           6255
      63,80,71,87,"wassermann"
1670 data "ist zwischen fisch und                                   3435
      steinbock"
1680 data 7,0,0,0,58,61,0,60,56,118,64,170,                         6223
      90,209,136,"kleiner wagen"
1700 data "beinhaltet den polarstern"                               2468
1720 data 6,0,0,25,105,21,182,91,170,64,                            5238
      104,103,47,"herkules"
1740 data "bei ihr ist kugelsternhaufen                             3229
      m13"
1760 data 7,0,0,59,0,43,141,61,181,86,130,                          6140
      143,116,29,24,"grosser hund"
1780 data "sie hat als hellsten stern den                           3628
      sirius"
1800 data 9,0,0,85,24,73,77,259,33,250,78,                          7410
      211,98,203,135,243,179,270,167,"loewe"
1820 data "stern regulus liegt auf der                              3714
      ekliptik"
1840 data 10,0,0,91,2,41,30,41,40,41,50,27,                         6985
      77,42,82,58,90,14,171,90,146,"orion"
1860 data "beruehmt durch ein nebelfeld"                            2778
1880 data 6,0,0,65,69,175,55,127,122,60,                            5238
      176,159,175,"schwan"
1900 data "so naeherte sich zeus der leda"                          3638
1920 data 7,0,0,53,44,99,53,157,57,194,21,                          6882
      262,62,246,117,"grosser wagen"
1940 data "das wohl bekannteste sternbild"                          2749
1960 data 5,14,65,53,29,93,43,124,4,159,44,                         4481
      "cassiopeia"
1980 data "sie war aethiopiens koenigin"                            2931
2000 data 13,23,145,42,136,48,115,55,88,65,                         6381
      74,113,44,141,21,161,36,244
2002 data 131,250,151,258,152,169,139,105,                          5640
      132,"steinbock"
2020 data "3 seiner sterne sind gut                                 3637
      sichtbar"
2040 data 9,8,96,4,63,57,18,76,11,90,85,                            5817
      114,12,133,33,117
2042 data 58,102,118,"noerdliche krone"                             2886
2060 data "utensil eines koenigs"                                   2943
2080 data 13,43,9,50,97,176,27,150,92,48,                           6766
      207,30,209,295,194,39,179,89
2082 data 171,121,190,168,223,154,227,177,                          5476
      253,"perseus"
2100 data "dort gibt es 2 sternhaufen"                              2325
2120 data 4,0,0,11,61,54,20,64,54,"hase"                            3102
2140 data "schmeckt als braten recht gut"                           3063
2160 data 0                                                         227
2180 graphic 0,1:print "Welche                                      5849
      Konstellation wollen Sie ? ":print
      "=>";:gosub 11000
2181 print "[clr]Bitte warten.[2spaces]Ich                          4900
      suche.":y$=xx$:gosub 20000:d$=y$
2182 if d$="" then 2180                                             1281
2200 read n                                                         291
2218 if n<>0 then 2240                                              1202
2220 print "Diese Konstellation ist mir                            3508
      unbekannt"
2222 print "Bitte versuchen Sie eine                               5320
      andere.":restore :for i=1 to 890:next
      i:goto 2180
2240 for i=1 to n:read x(i):read y(i):next                          2568
      i
2260 for i=1 to n:y(i)=-y(i):next i                                 2346
2280 read b$,h$                                                     772
2290 y$=b$:gosub 20000:b1$=y$                                       1936
2300 if b1$=d$ then 2340                                            1388
2320 goto 2200                                                      569
2340 restore                                                        140
2360 goto 540                                                       580
2380 r=int (30*rnd (1)+1)                                           1226
2400 for k=1 to r                                                   766
2420 read n                                                         291
2440 for j=1 to n:read x(j),y(j):next j                             2043
2460 for i=1 to n:y(i)=-y(i):next i                                 2346
2480 read b$,h$                                                     772
2500 next k                                                         280
2520 restore                                                        140
2540 goto 540                                                       580
2550 end                                                            128
2551 print "[clr]"                                                  316
2552 print "[2down,3spaces]S[2spaces]I                             2355
```

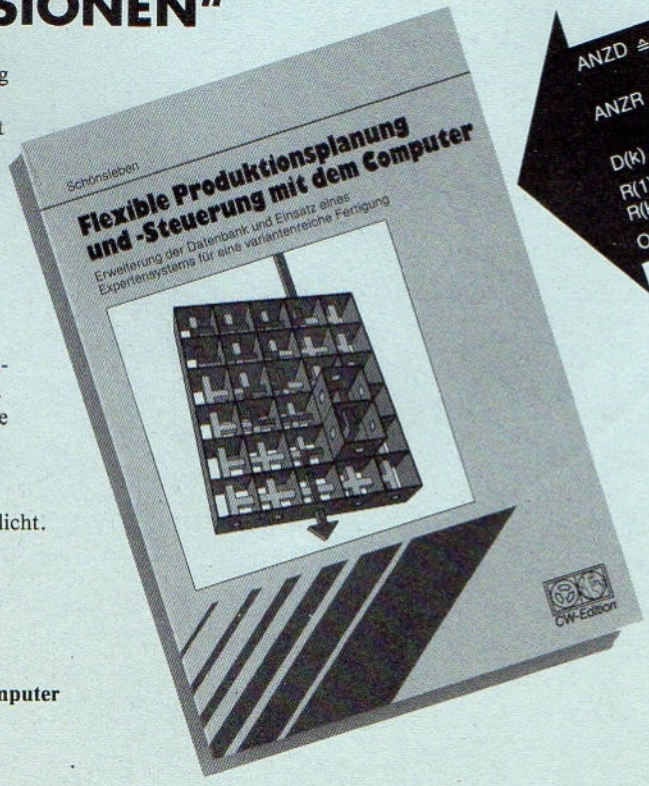

WWW.HOMECOMPUTERWORLD.COM

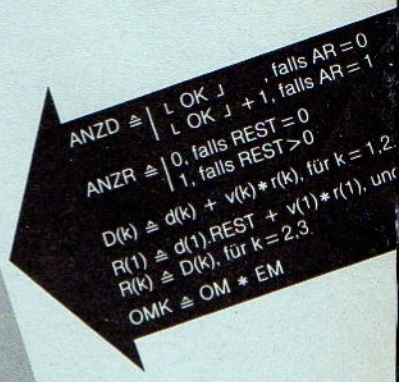
```
     [2spaces]A[2spaces]R[2spaces]P
     [2spaces]I[2spaces]C[6spaces]1
     [2spaces]2[2spaces]8"
 2553 print "[3down]"                                  510
 2554 print "[3spaces]by[4spaces]Frank                1780
      Bastian"
 2556 print "[5down] published by[4spaces]             2181
      RUN"
 2590 x=0                                              381
 2610 print "[4down]Bitte Taste druecken.";            2988
 2620 get a$:if a$="" then 2620                        1552
 2640 print "[clr]":return                             688
 9000 poke 208,0                                       836
 9001 get a$:if a$="" then 9001                        1512
 9002 graphic 0,1:print "Fuer das Menue ' +            3939
      ' druecken."
 9005 print "Eine andere Taste fuehrt                  3406
      weiter."
 9010 poke 208,0                                       836
 9020 get a$:if a$="" then 9020                        1512
 9030 if asc (a$)=95 then print "[clr]":goto           1944
      160
 9040 return                                           142
10000 print                                            153
10005 print "Loewe,[4spaces]Widder,[5spaces]          2908
      Zwillinge"
10010 print "Hase,[5spaces]Schwan,[5spaces]           2810
      Steinbock"
10020 print "Leier,[4spaces]Kepheus,                  3227
      [4spaces]Wassermann"
10030 print "Waage,[4spaces]Delphin,                  3069
      [4spaces]Cassiopeia"
10040 print "Adler,[4spaces]Pegasus,                  3533
      [4spaces]Kleiner Hund"
10050 print "Krebs,[4spaces]Schuetze,                 3710
      [3spaces]Kleiner Wagen"
10060 print "Pfeil,[4spaces]Perseus,                  3345
      [4spaces]Grosser Wagen"
10070 print "Stier,[4spaces]Skorpion,                 4238
      [3spaces]Noerdliche Krone"
10080 print "Orion,[4spaces]Fuhrmann,"                2182
10090 print "Drache,[3spaces]Jungfrau"                2022
10100 print "Fische,[3spaces]Herkules"                1963
10200 return                                           142
11000 rem "---Eingaberoutine fuer Input
11010 xx$="":                                          801
11020 get x$:if x$="" then 11020                       1569
11030 if x$="[left]" or x$=chr$ (20) then if           5562
      len (xx$)>0 then xx$=left$ (xx$,len
      (xx$)-1):goto 11100
11040 if x$=chr$ (13) then if xx$<>"" then             2419
      return
11050 if x$=chr$ (13) then if xx$="" then              2311
      11020
11060 if ( asc (x$) and 127)<32 then 11020             2306
11070 xx$=xx$+x$                                       955
11100 char ,0,1,xx$+" "                                1111
11110 goto 11020                                       601
20000 y1$="":xa=len (y$):for xb=1 to xa:y1=            4618
      asc ( mid$ (y$,xb,1)) and 127
20010 if y1<>32 then y1$=y1$+chr$ (y1)                 2574
20020 next :y$=y1$:return                              647
25000 print "[clr]E[shift-space]N                     1145
      [shift-space]D[shift-space]E"
```

„Starpic": Sternenpracht auf dem Bildschirm

FLEXIBLE DB-ORGANISATION CONTRA „STÜCKLISTEN-EXPLOSIONEN"

Immer wenn die computerunterstützte Produktionsplanung und -steuerung mit wenig stabilen Rahmenbedingungen leben muß, ist die DV-technische Seite besonders gefordert — in kleineren und mittleren Betrieben oft überfordert. Häufige Produktwechsel, Störungen an den Werkzeugen oder Maschinen oder Unterbrechungen bei der Materialversorgung stellen typische PPS-Problemfälle dar. Und geradezu wenn eine hohe Anzahl von Varianten innerhalb des Erzeugungsspektrums gegeben ist, fressen Stücklisten und Bedarfsrechnung Speicherkapazität und Rechnergeschwindigkeit auf.
Mit der vorliegenden Arbeit werden in der Praxis entwickelte und erprobte Datenstrukturen und Algorithmen gezeigt. Unter anderem wird ein Expertensystem für variantenreiche Produktionsprogramme realisiert, das durch seine Verknüpfungstechniken redundante Daten abbaut, „Stücklisten-Explosion" vermeidet und somit die wirtschaftliche Verarbeitung großer Datenmengen im PPS-Bereich ermöglicht.





Prof. Dr. Paul Schönsleben
Flexible Produktionsplanung und -Steuerung mit dem Computer
144 Seiten, München 1985
Best.-Nr.: 1 CW 53-2 DM 58,—

CW EDITION
Fachbücher für die Computerwelt

WWW.HOMECOMPUTERWORLD.COM

Auf Ordnung programmiert

RUN

Sammelbox

Ab sofort herrscht Übersicht!
Dafür sorgen diese attraktiven Sammel-Boxen aus stabilem Kunststoff, in denen jeweils 6 Ausgaben Platz finden.
So haben Sie alle RUN-Hefte übersichtlich und griffbereit zur Hand, wenn Sie etwas nachlesen wollen. Mit 2 Sammel-Boxen können Sie einen kompletten Jahrgang der Hefte zusammenfassen und vor Beschädigung geschützt aufbewahren – für nur DM 9,80 inkl. MwSt. und Versandkosten je Box.

WWW.HOMECOMPUTERWORLD.COM

Jede Box nur DM

9,80



CW PUBLIKATIONEN

Füllen Sie bitte gleich den Bestellschein aus und senden Sie ihn noch heute an uns ab:

CW Publikationen, Verlagsgesellschaft mbH
– Vertrieb RUN –
Postfach 40 04 29 · D-8000 München 40

Bestellschein

Bitte ausfüllen und ggf. mit Verrechnungsscheck in frankiertem Umschlag schicken an:
CW Publikationen Verlagsgesellschaft mbH – Vertrieb RUN – Postfach 40 04 29, D-8000 München 40

Zahlungsweise (Zutreffendes bitte ankreuzen):

☐ Verrechnungsscheck über DM _____ liegt bei.

☐ Den Gesamtbetrag in Höhe von DM _____ habe ich auf Postgiro-Konto Nr. 233900-808, Postgiroamt München, mit dem Vermerk »Sammelbox RUN« einbezahlt.

Ja, bitte schicken Sie mir umgehend ____ Boxen (bitte Anzahl eintragen) zum günstigen Stückpreis von nur DM 9,80 inkl. MwSt. und Versandkosten.

Name Vorname

Straße/Nr.

PLZ Ort

Datum/Unterschrift

R11

Hier bitte abtrennen!

IHR WEGWEISER FÜR DIE 68000ER-WELT

Im Zeitschriftenhandel erhältlich



Ausführliche Informationen über Amiga, Atari ST und Macintosh. Objektive und aussagekräftige Tests, verbunden mit umfangreichen Marktübersichten. So gründlich, aufwendig und umfassend, wie es Ihnen nur ein starkes Team von Fachleuten bieten kann. Nutzen Sie dieses Know-how für sich.

Information hat einen Namen: ICON

CW PUBLIKATIONEN

Ihre starke Verbindung zur Computerwelt

Postfach 40 04 29, D-8000 München 40

BESTELLSCHEIN: JA, ich bestelle das ICON Sonderheft direkt beim Verlag.

Bestellschein bitte ausfüllen und ggf. mit unterschriebenem Verrechnungsscheck in einem ausreichend frankierten Umschlag schicken an: CW-Publikationen Verlagsgesellschaft mbH – Vertrieb Postfach 40 04 29, D-8000 München 40

Zahlungsweise (Zutreffendes bitte ankreuzen):
Verrechnungsscheck über DM 16,– (DM 14,– Heftpreis + DM 2,– Rückporto) liegt bei.
Den Gesamtpreis in Höhe von DM 16,– (DM 14,– Heftpreis + DM 2,– Rückporto) habe ich auf Postgiro-Konto Nr. 2339 00-808, Postgiroamt München, mit dem Vermerk »ICON« überwiesen

* Inlandpreise

Name/Vorname

Straße/Nr.

PLZ/Ort

Unterschrift Datum

SH 3

WWW.HOMECOMPUTERWORLD.COM