mister micro

Spectrum

BASIC-ABENTEUER Band 1

Der fremde Planet

Dieses didaktisch hervorragende Konzept des Computer-Unterrichts für Kinder vermittelt Computerwissen auf einer spielerischen Ebene: das Kind lernt die BASIC-Programmierung für den ZX Spectrum, während es mit MISTER MICRO einen Ausflug in das 21. Jahrhundert unternimmt.

Der vorliegende Band 1 dieser Abenteuer-Serie enthält eine aufregende Geschichte, bei der unsere Helden auf ein verlassenes Raumschiff verschlagen werden. Um sich aus dieser Situation zu befreien, müssen sie (und auch das Kind) lernen, den Computer des Raumschiffs zu bedienen.

Dieses ungewöhnliche Abenteuer wurde erdacht, um die Grundlagen der ZX Spectrum-Programmierung auf spannende und spielerische Weise zu vermitteln — so lernt man am leichtesten.

Begleitend dazu enthält der Datenträger dieselben Programme wie der Computer des Raumschiffs und dazu noch einige Lernprogramme, die das Kind für seine spielerischen Lernerfolge belohnen.

Jeder erwähnte BASIC-Befehl wird im zweiten Teil des Buches ausführlich erläutert, so daß Sie immer gleichzeitig ein Nachschlagewerk zur Programmierung in Händen halten.

Buch und Kassette Best.-Nr. 3410

ISBN 3-88745-[illegible]

SYBEX

Der spannendste Weg zur Programmierung in BASIC

3410

FJ
5343
(1)

Spectrum BASIC-ABENTEUER Band 1 Der fremde Planet



mister micro

Spectrum

BASIC-ABENTEUER Band 1

SYBEX

Spectrum
BASIC Abenteuer
Band 1

Weitere Titel in dieser Serie:

Commodore 64 BASIC-Kurs
Best.-Nr. 3400 Buch + Kassette
Best.-Nr. 3401 Buch + Diskette

Commodore 64 Assembler-Kurs
Best.-Nr. 3402 Buch + Kassette
Best.-Nr. 3403 Buch + Diskette

Commodore 64 BASIC-Abenteuer
Best.-Nr. 3404 Buch + Kassette
Best.-Nr. 3405 Buch + Diskette

VC20 Assembler-Kurs
Best.-Nr. 3406 Buch + Kassette

VC20 BASIC-Abenteuer
Best.-Nr. 3407 Buch + Kassette

Apple II/IIe Assembler-Kurs
Best.-Nr. 3408 Buch + Diskette

Spectrum BASIC-Kurs
Best.-Nr. 3409 Buch + Kassette

mister micro

Spectrum
BASIC Abenteuer
Band 1:

Der fremde Planet

UNIVERSITÄTSBIBLIOTHEK
HANNOVER
TECHNISCHE
INFORMATIONSBIBLIOTHEK

SYBEX

BERKELEY · PARIS · DÜSSELDORF

Ihre mister micro-Garantie FJ 5343 (1)

Falls sich diese Programme nicht laden lassen, schicken Sie bitte **nur den Original-Datenträger** an uns zurück. Wir ersetzen ihn kostenlos!

SYBEX-Verlag GmbH
Vogelsanger Weg 111 · Postfach 300961
4000 Düsseldorf 30
Telefon (0211) 626441 · Telex 8588163

Originalausgabe in Englisch
Titel der englischen Ausgabe:
„ZX Spectrum BASIC Adventure Part I"
Original Copyright © 1983 by Glentop Publishers Ltd.

Deutsche Übersetzung: IPMC Ltd.
Umschlagentwurf: Daniel Boucherie
Zeichnungen: Nik Stanbury und Michael Albers
Satz: tgr – typo-grafik-repro gmbh, remscheid
Gesamtherstellung: Druckerei Hub. Hoch, Düsseldorf

Der Verlag hat alle Sorgfalt walten lassen, um vollständige und akkurate Informationen zu publizieren. SYBEX-Verlag GmbH, Düsseldorf, übernimmt keine Verantwortung für die Nutzung dieser Informationen, auch nicht für die Verletzung von Patent-, Lizenz- und anderen Rechten Dritter, die daraus resultieren.

ISBN 3-88745-410-3 Buch + Kassette
1. Auflage 1984

Alle deutschsprachigen Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlages reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Printed in Germany
Copyright © 1984 by SYBEX-Verlag GmbH, Düsseldorf

Inhaltsverzeichnis

So benutzt du dieses Buch am besten 7

So benutzt du die Programme am besten 8

Der Hintergrund zur Geschichte 10

Episode 1

Auf einem fremden Raumschiff 13
Die Tastatur ; CAPS SHIFT, ENTER, CLS; RUN/STOP; Laden (LOAD) eines Programms; LIST; PRINT; INPUT; CREWDATEN; LET; +; =; Speicherplatznamen

Episode 2

Unsere Helden ändern erfolgreich ein Programm ... 29
NEW; Zeilennummern; RUN

Episode 3

Die Getränkemaschine 41
GETRAENKE; GO TO; £; Zeichenfolgen/ Meldungen; IF-THEN

Episode 4

Eine aufregende Entdeckung 51
RND + − * / ; INT

Episode 5

Aleate erfindet ein Spiel 59
> ; < ; Zahlen-Ratespiel

Episode 6

Schleifen knüpfen 67
FOR-NEXT; STEP; TO

Episode 7

Start des Raumschiffs 77
SAUERSTOFF; LEN; CODE; CHR£;

Mister M erklärt 89

Lösungen 123

Stichwortverzeichnis 127

So benutzt du dieses Buch am besten

Dieses Buch handelt von einem Abenteuer – nein, eigentlich von zwei Abenteuern! Das eine spielt sich im Weltraum ab. Unsere Helden müssen nämlich lernen, wie man den Computer eines Raumschiffs bedient, damit sie von einem weit entfernten Planeten entfliehen und nach Hause kommen können.

Der Held des zweiten Abenteuers ist aber viel, viel wichtiger – das bist nämlich du selbst! Denn da lernst du, wie du deinen eigenen Computer bedienen kannst. Da will es nun der Zufall, daß der Raumschiff-Computer deinem Sinclair ZX Spectrum unwahrscheinlich ähnlich ist. Du kannst also gemeinsam mit Mister Micro, Aleate und SN7 lernen! Zum Glück hat einer der Weltraumwanderer, nämlich Mister Micro, ein Tagebuch geführt. Als er endlich nach Hause kam, hat unser fleißiger Freund eine ganze Reihe von Erklärungen für seinen Neffen Rolf aufgeschrieben, der der stolze Besitzer eines Sinclair ZX Spectrum ist. Diese Erklärungen werden dich sicher interessieren, und deshalb haben wir die meisten einmal für dich aufgeschrieben. Du findest sie im Abschnitt „Mister M erklärt". Wenn du also eines der verwendeten Computerwörter nicht so ganz verstehst, schaust du am besten gleich in diesem Abschnitt nach.

So benutzt du die Programme am besten

In unserer Geschichte finden die drei Helden eine Kassette, auf der einige Programme gespeichert sind. Diese Programme haben wir für dich kopiert. Im Buch erfährst du, wann du eines der Programme laden sollst.

Die folgenden Programme sind im Text erklärt:

CREWDATEN SAUERSTOFF

GETRAENKE

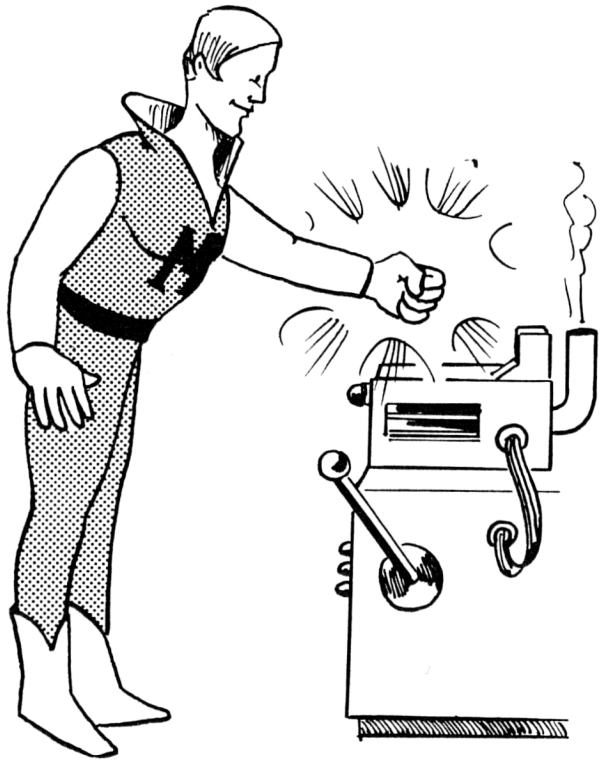
Außerdem findest du vier Lernprogramme, die dir bei einigen Kommandos helfen, die in BASIC verwendet werden. Ein Übersichtsprogramm (INHALT) und ein Wortratespiel (HANGMAN) sind auch noch mit auf der Kassette. Diese Programme kannst du benutzen, wann du willst – während du das Buch liest oder auch bevor du mit dem Lesen anfängst. Sie sind als zusätzliche Hilfe für dich gedacht. Benutze sie, wenn du meinst, daß du sie gebrauchen kannst:

INHALT	gibt dir einen Überblick über die gesamten Programme
PRINTES	hilft dir, das Kommando PRINT zu verstehen
NUMMERN	erklärt dir, wie die Zeilennummern in BASIC funktionieren
STRINGY	hilft dir bei dem Befehl TO

RATEN	ist ein Programm, in dem du etwas über das Programm für das Zahlen-Ratespiel erfährst, das Aleate im Kapitel 5 schreibt.

HANGMAN ist ein Wortrate-Programm

Der Hintergrund zur Geschichte...



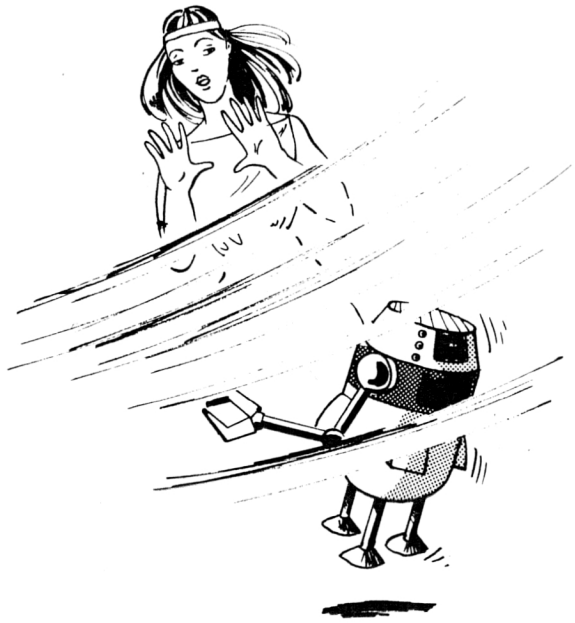
Mister Micro hat Anfangsprobleme mit seiner neuesten verbesserten Beam-Maschine, der Mark Nr. 41. Seit seinen ersten Schultagen hat sich Mister M auf die Technologie des Beamens spezialisiert.

Er war Schüler in München, bis eines Tages eines seiner mehr experimentellen Experimente schiefging und er die

ganze Schule nach Hapersdorf, einem kleinen verschlafenen Dörfchen mitten auf dem Lande, beamte. Mister Micros Mark Nr. 41 ist genau so unberechenbar wie alle seine anderen Maschinen, aber viel stärker! Wir kommen gerade dazu, wie Mister M versucht, zum Elektronikladen zu beamen, wo er sich ein kleines elektrisches Teil holen will. Wie immer will ihm die Maschine nicht so recht gehorchen. Um das Problem zu lösen, versetzt er ihr einen kurzen Schlag. Das hilft: Sie beginnt zu surren, und Mister M wird vom Beam erfaßt.

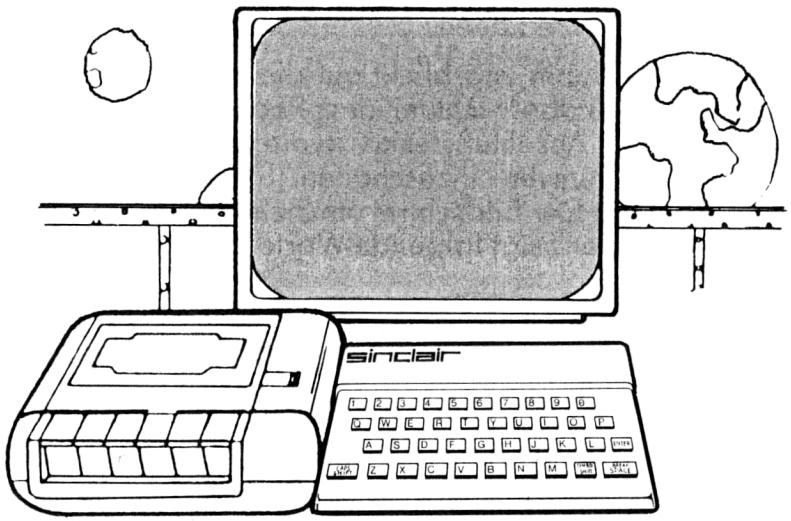
Währenddessen...

Aleate, eine junge Arfonitin (241 Jahre alt), und SN7, ein kleiner Wartungsroboter, haben Probleme mit dem Hyperantrieb ihres Raumschiffs. Sie wissen es noch nicht, aber Schuld an ihren Schwierigkeiten ist die Beam-Maschine unseres Mister M. Plötzlich gibt es einen Ruck, und sie werden davongebeamt.



EPISODE 1

Auf einem fremdem Raumschiff



Unsere Helden materialisieren sich in der Steuerzentrale eines verlassenen Raumschiffs von einem fremden Planeten.

„Nanu! Wo bin ich denn bloß?“ wundert sich Aleate, während sie sich vom staubbedeckten Boden erhebt, „und wer sind Sie?“

Mister M dreht den Kopf und reibt sich die Augen. „Ich? Ich heiße Mister Micro. Tut mir furchtbar leid, ich habe da wohl einen Fehler gemacht. Ich wollte gerade zum Elektronikladen beamen und ... und ...", Mister M sucht nach Worten. Inzwischen rollt SN7, der Wartungsroboter, quer durch den Raum.

„Aufstehen! Aufstehen! Es gibt viel zu tun!" wiederholt er wieder und wieder.

Aleate reißt sich zusammen. „Es sieht so aus, als ob wir zusammenarbeiten müssen, wenn wir etwas über dieses Schiff herausfinden wollen; sonst kommen wir nie zurück nach Hause."

„Seht mal", Mister Micro steht vor einer Schalttafel, „das sieht wie eine Art Tastatur aus. Damit kann man wohl den Hauptcomputer bedienen."

Aleate geht zu ihm und blickt auf die Tastatur. Sie sieht, daß das Stromkabel nicht mit dem Rechner verbunden ist, und steckt den Anschlußstecker in die Buchse ganz rechts am Computer, um ihn einzuschalten. (Schalte deinen Computer jetzt ein.) Der Bildschirm am Sichtanzeigegerät wird lebendig, und er zeigt folgende Worte an:

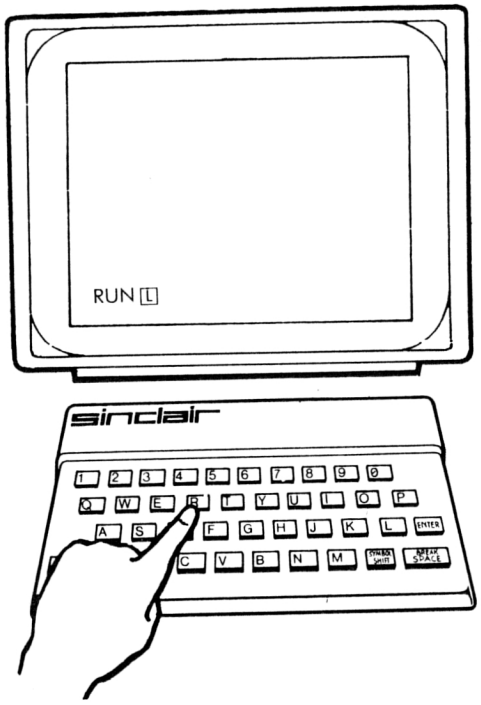


In der Zwischenzeit hat Mister M eine kleine Kassette auf dem Boden in der Ecke gefunden. „Was ist das?"

„Das ist eine Art gespeichertes Programm. Es ist offensichtlich für diese Computermaschine bestimmt", schnarrt SN7 mit seiner durchdringenden Stimme. „Wir könnten versuchen, es in den Kassettenrecorder zu schieben, der hier steht." Er schnappt sich das Band und legt es in den Kassettenrecorder.

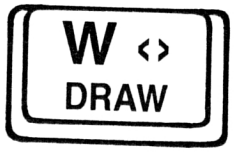
„So", sagt Aleate, „und was jetzt? Wie können wir den Computer dazu bringen, etwas zu tun?"

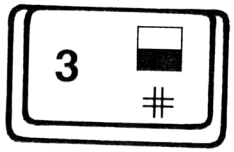
Sie blicken alle auf die Symbole auf der Tastatur. Einige kennen sie schon, andere jedoch nicht. Aleate drückt die Taste mit dem Buchstaben „R". Auf dem Bildschirm erscheint das Wort „RUN" und ein blinkendes L.



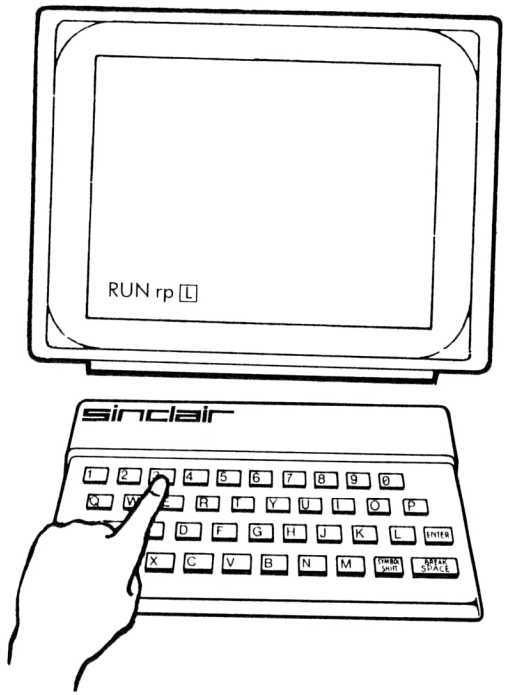
„Aha!" ruft Mister M. „Die meisten Tasten haben mehr als ein Symbol." Er drückt noch einmal auf die Taste „R". Im Gegensatz zu eben erscheint jetzt ein kleines „r" auf dem

Schirm. Laßt uns erst einmal nur die Symbole oben auf den Tasten ansehen.





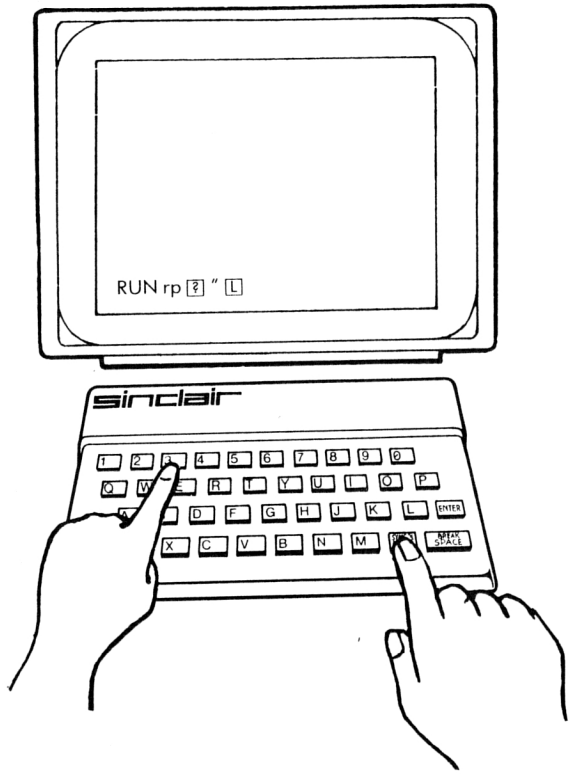
Es scheint so, daß der erste Tastendruck am Anfang einer Zeile das in Weiß geschriebene Wort auf der Taste wiedergibt. Erst, wenn das blinkende L erscheint, kann man die einzelnen Buchstaben auf den Tasten benutzen. Aleate drückt die Taste mit dem Buchstaben „P", und sofort erscheint der kleine Buchstaben p darauf.



„Wie können wir wohl die in Rot gedruckten Wörter und Symbole bekommen?" fragt Mister Micro.

SN7 weiß es: „SYMBOL SHIFT drücken und halten, SYMBOL SHIFT drücken und halten."

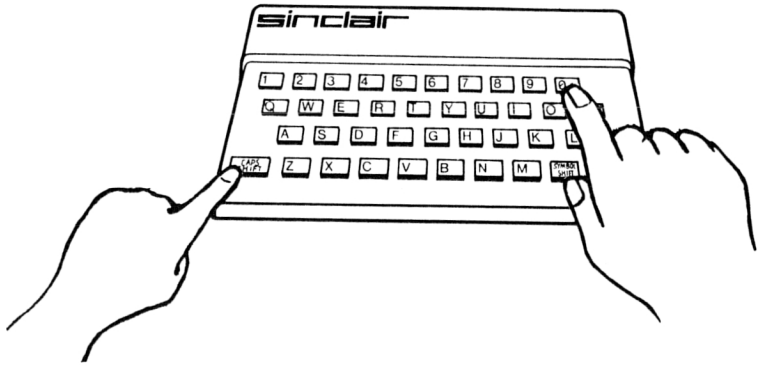
Mister Micro streckt den Finger aus und hält die Taste SYMBOL SHIFT gedrückt, während Aleate ihre Taste erneut drückt. Auf dem Bildschirm erscheint das rote Symbol der Taste. „Ich möchte mal wissen, was diese Taste macht", wundert sich der Doktor und drückt die mit ENTER bezeichnete Taste. Sogleich erscheint auf dem Bildschirm die Meldung:



„Na klar, er versteht offensichtlich nicht, was wir da eingetippt haben", meint er.

„Nun wollen wir doch einmal sehen, was für Informationen auf der Kassette gespeichert sind." Er schließt den Kassettenrecorder an eine Buchse auf der Rückseite des Computers an.

SN7 schließt sich selbst an eine nahegelegene Buchse mit der Bezeichnung „Roboter-Schnittstellenmodul" an. „Die Analyse ergibt, daß Niederdrücken von CAPS SHIFT und gleichzeitiges Betätigen der Taste DELETE die Symbole auf dem Bildschirm löscht. Eintippen von LOAD"" und Drükken der ENTER-Taste bewirkt, daß die Kassette aktiviert wird", bietet SN7 als Rat an.



Diesem Rat folgt Mister Micro. Der Bildschirm wird weiß. Jetzt startet er den Recorder. Nach einigen Sekunden tauchen an den Rändern des Bildschirms rote und blaue Streifen auf, der Recorder gibt hohe Töne von sich.

Auf dem Bildschirm erscheint die Meldung:

Program: CREWDATEN

Die Streifen haben sich jetzt blau und gelb verfärbt. Dann nimmt der Bildschirm seine frühere Farbe wieder an.

Die Meldung: 0 OK, 0 : 1 erscheint.

RUN gefolgt von ENTER startet das Programm, und die erste Frage erscheint ebenfalls.

Aleate beantwortet sie durch Drücken der Taste mit der Nummer 1. (Das solltest du jetzt auch tun.) Es passiert aber gar nichts.

„Versuch' es mal mit Drücken der Taste ENTER", rät Mister Micro. „Der Computer hat schon einmal darauf reagiert", fügt er hinzu.

Aleate drückt ENTER, und auf dem Bildschirm erscheint eine neue Frage. Aleate betätigt die Taste mit der 3 (drücke die Taste auch), aber anstatt ENTER zu drücken, ruft sie: „O nein! Ich wollte doch Nummer 2 tippen. Was nun?"

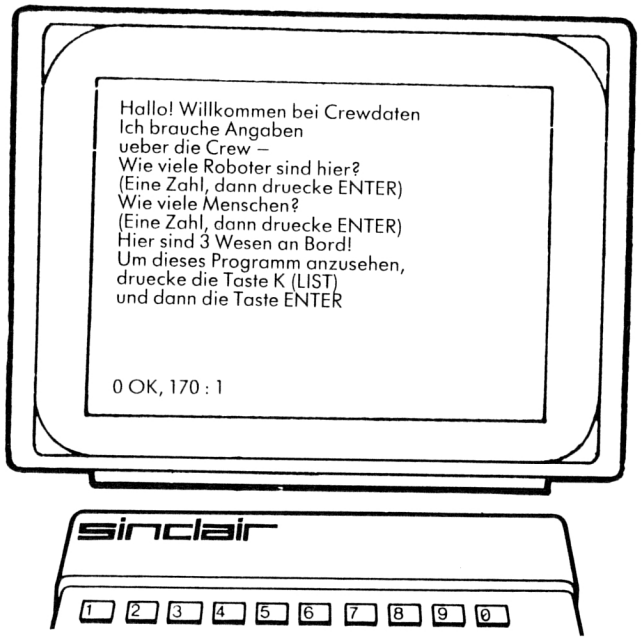
Sie sehen sich die Tastatur genau an. In der oberen Reihe ganz rechts entdecken sie eine Taste mit der Bezeichnung



Das heißt löschen", rät Mister M. Aleate drückt die Taste zusammen mit CAPS SHIFT, und die 3 verschwindet vom Bildschirm.

„Prima", sagt sie und betätigt die Taste mit der 2 und dann die Taste ENTER. Eine Meldung erscheint mit der Angabe, wie viele Crew-Mitglieder an Bord des Schiffs sind.

Schließlich zeigt der Bildschirm die Meldung 0 OK, 170 : 1 an.



„Was soll das wohl bedeuten?" fragt Mister Micro.

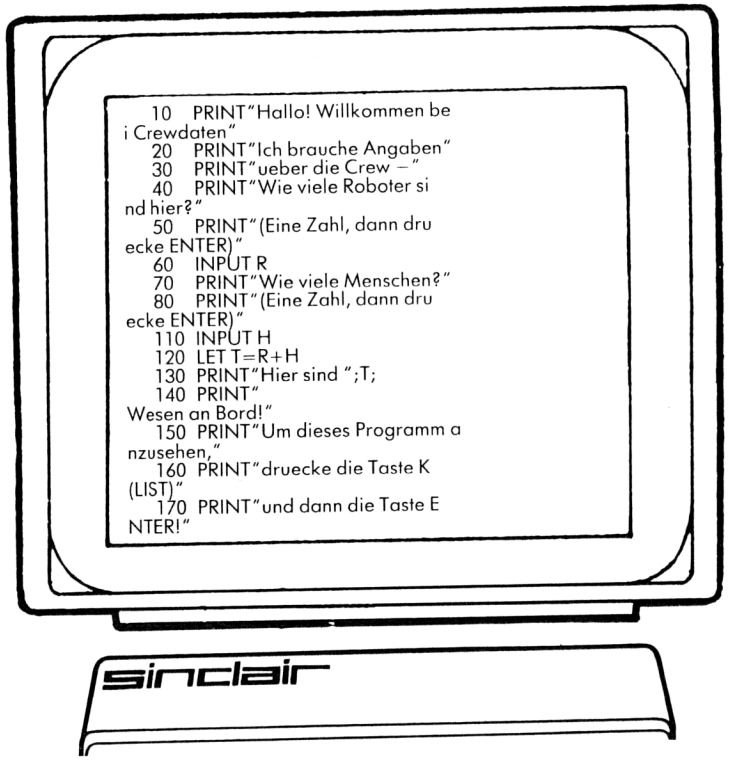
„Nun", zögert Aleate, „vielleicht heißt das, daß der Computer zu irgend etwas bereit ist!"

„Er ist bereit für uns! Er ist bereit für uns!" flötet SN7 dazwischen. „Das ist klar", Mister M kratzt sich am Kopf. „Aber WAS wollen wir denn tun!"

„Na, er sagt uns ja, wir sollen den Befehl LIST eintippen. Probieren wir es." Aleate beugt sich vor und tippt LIST. Es passiert aber gar nichts. (Hast du schon LIST auf deinem Computer getippt?)

„Ja, da hat der Computer also überhaupt nicht reagiert!" Mister M überlegt. „Vielleicht sollten wir die Taste ENTER immer dann drücken, wenn wir dem Computer gesagt haben, daß er etwas tun soll." Er streckt die Hand zur rechten Seite der Tastatur aus und drückt ENTER.

Und schon erscheint eine Liste auf dem Bildschirm:

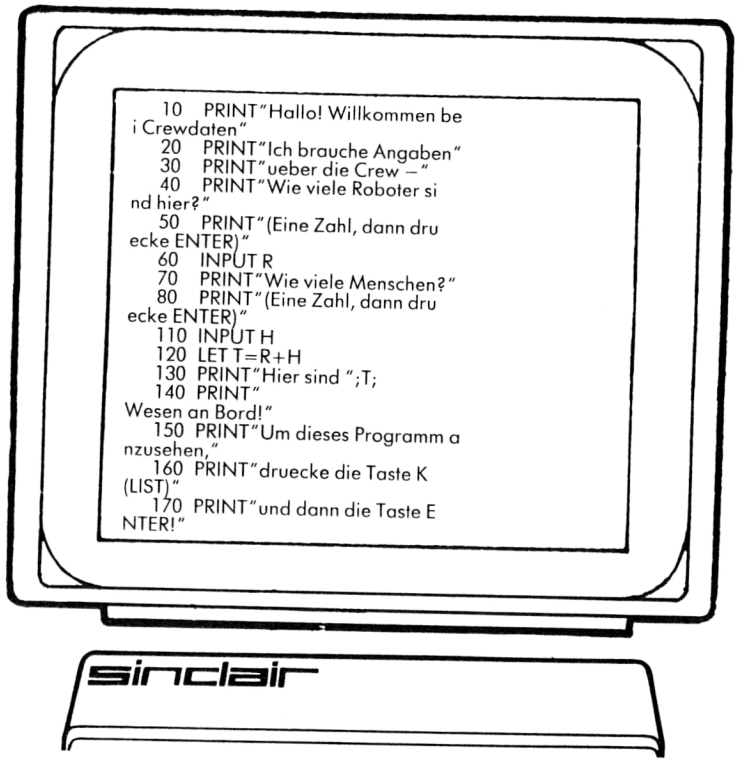


„Vielleicht eine Geheimsprache?" meint Mister Micro. „Glaube ich nicht", antwortet Aleate, „mir scheint, dies ist das in Frage kommende Programm."

„Aha", Mister Micro geht ein Licht auf. „Also, jede dieser Zeilen scheint mit einer Zahl zu beginnen, und zwar in aufsteigender Reihenfolge. Höchst sonderbar!" Er hat eine Idee: „Wir können annehmen, daß jede Zeile eine Art Anweisung ist: Jede Zeile sagt dem Computer, daß er etwas Bestimmtes ausführen soll."

„O.K.", übernimmt Aleate das Wort, „dann sollte der Computer eine dieser Anweisungen verstehen können, wenn wir sie für sich eintippen."

„Genau", antwortet Mister Micro. „Sehen wir uns mal Zeile 10 an. Du erinnerst dich: Das erste, was das Programm auf unseren Bildschirm schrieb, war „Hallo willkommen bei Crewdaten". Es benutzt solche Schreibanweisungen mit dem Wort PRINT auch in anderen Zeilen dieses Programms."



„Also", meint Aleate, „dann müßte der Computer ja dieses PRINT-Kommando verstehen. Das heißt, er soll etwas schreiben oder ausgeben."

Sie tritt an die Tastatur heran und tastet ein:

PRINT"Hallo"

Der Computer regt sich nicht! „Drücke mal ENTER", sagt Mister Micro. Das macht Aleate. Und auf dem Bildschirm erscheint das Wort:

Hallo

„Das klappt ja prima!" freut sich Mister Micro.

„Moment mal. Wir wollen etwas anderes ausprobieren. Ich möchte, daß er „ICH MAG EIS" ausgibt. Aber wie bekomme ich die Zwischenräume zwischen die Wörter?" Sie blickt auf die Tastatur. „Auf dieser Taste hier rechts unten steht SPACE drauf – Zwischenraum, das bedeutet das englische Wort „SPACE". Das könnte also die richtige Taste sein." Ihre Vermutung stimmt tatsächlich, und sie tippt:

PRINT"Ich mag Eis"

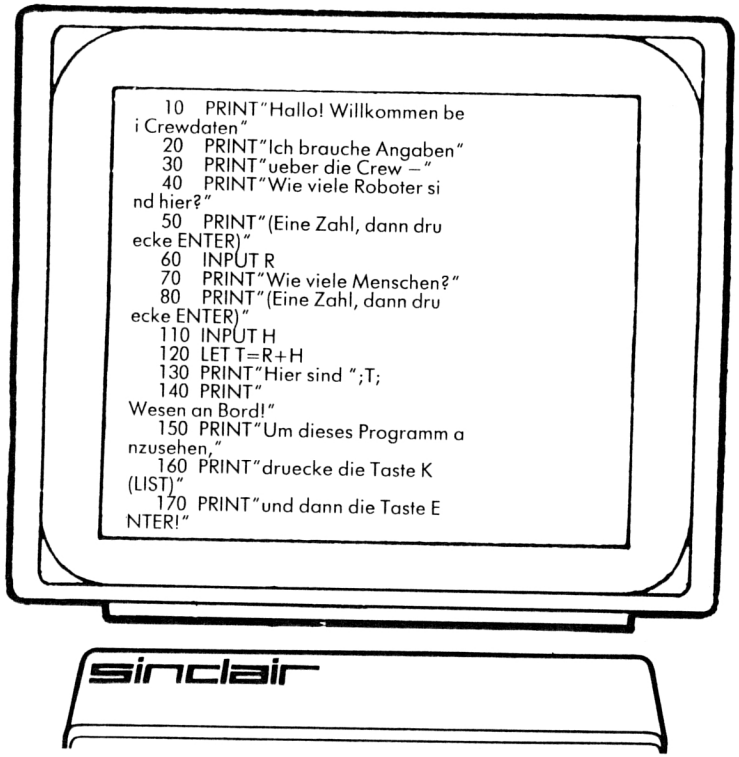
(Schreibe das auch mal auf deinem Computer!) Sie drückt dann ENTER, und auf der Anzeige erscheint die Meldung:

Ich mag Eis

Aleate folgert daraus: „Wir können also alles schreiben, was wir wollen. Dafür brauchen wir nur erst das Wort PRINT einzugeben, dann das, was geschrieben werden soll, in Anführungszeichen."

„So, jetzt verstehen wir also einen Befehl", sagt Mister M, „aber wie steht's dann mit Zeile 60? Moment ... sieh doch! Das Programm fragt, wie viele Roboter an Bord sind. Erinnerst du dich? Wir haben doch eine Zahl eingegeben, nicht wahr, als wir danach gefragt wurden! Vielleicht ist INPUT eine Anweisung, damit Informationen oder Daten eingegeben werden sollen, zum Beispiel eine Zahl oder so."

„Ja. Aber was bedeutet wohl das R?" fragt Aleate.



„Also, das ist mir auch nicht so ganz klar", sagt Mister Micro. Er macht ein ratloses Gesicht.

„Wo geht die Zahl hin, wenn du eine Zahl eingibst. Wo geht sie hin?" zwitschert SN7 und rollt über den Boden zu Aleate und Mister M.

„Ach ja! Die muß an einen Speicherplatz im Speicher des Computers gehen", meint unser Held.

„Da müssen ja aber Tausende von solchen Speicher- oder Gedächtnisplätzen sein", kratzt sich Aleate am Kopf. „Wie kann man diese unterscheiden? Vielleicht hat jeder einen bestimmten Namen, und R ist vielleicht der Name eines Speicherplatzes!"

„Ja, ich glaube, du hast recht", schmunzelt Mister Micro.

„Ich weiß, wie wir mal ein Experiment machen können." Aleate tritt an die Tastatur heran. „Was passiert, wenn wir dem Computer sagen:

```
PRINT"R"
```

Ich würde erwarten, daß dann der Buchstabe R auf den Bildschirm geschrieben wird." Sie versucht es und hat Erfolg. „Was passiert jetzt aber, wenn ich

```
PRINT R
```

schreibe? Ich will das mal ausprobieren." Sie tippt es ein, und der Computer schreibt eine 1 auf den Bildschirm.

„Das ist die Zahl, die wir eingegeben haben, als uns der Computer fragte, wie viele Roboter an Bord seien", sagt Mister M. SN7 dreht eine Runde im Raum, eine Sieges- oder Ehrenrunde!

„Ach so", unterbricht Aleate, „wenn wir PRINT R sagen, weisen wir also den Computer an, den Inhalt des Speicherplatzes R zu schreiben."

„Und wenn das R nun innerhalb von Anführungszeichen steht?" fragt Mister Micro.

„Das ist dann etwas anderes", antwortet Aleate, „wenn der Computer etwas von Anführungszeichen umgeben sieht, dann schreibt er dies genau so auf den Bildschirm, wie es eingetippt wurde."

„Was soll bloß dieses Wort INPUT?" fragt Mister Micro nachdenklich.

„Na, das ist recht einfach!" ruft Aleate. „Er wartet darauf, daß wir ihm sagen, wie viele Roboter oder Personen hier an Bord des Raumschiffs sind – das Wort INPUT ist eine Aufforderung an den Computer, zu warten, bis wir Daten oder Informationen eingegeben haben."

„Richtig", stimmt Mister Micro zu, „und er speichert die Informationen an dem Speicherplatz, der in unserem Fall R oder H heißt."

„Stimmt", sagt Aleate.

„Jetzt zur Zeile 120", Mister M blinzelt ein bißchen. „Das scheint so etwas wie ein mathematischer Ausdruck zu sein: LET T=R+H muß bedeuten, daß man die Zahlen in R und H zusammenzählen und die Antwort dann an den Speicherplatz T geben muß."

„Ja", sagt Aleate und klopft Mister Micro auf die Schulter.

„Nun wollen wir mal versuchen", fährt Aleate fort, „dieses LET-Wort zu verwenden." Sie schreibt:

```
LET X=7
```

und drückt die Taste ENTER. (Du solltest jetzt das gleiche tun). Die Meldung 0 OK, 0 : 1 erscheint.

„Was hat das gebracht?" fragt Mister M.

„Nun", erklärt Aleate, „der Computer scheint diese Anweisung akzeptiert zu haben, und ich hoffe, er hat eine 7 an den Speicherplatz mit dem Namen X gegeben. Das können wir mit dem, was wir bereits gelernt haben, einmal nachprüfen. Wenn wir

```
PRINT X
```

eintippen, sollte eine 7 erscheinen."

„Richtig, aber der Computer weiß, daß X eine 7 ist, von der oberen Zeile, die ja LET X=7 heißt. Was passiert, wenn wir den ganzen Bildschirm löschen und dann PRINT X eintasten?" fragt Mister Micro.

„Gut. Machen wir's mal." Aleate ist ein bißchen ratlos. „Aber wie können wir denn den Bildschirm löschen?"

SN7 rollt heran. „Die Taste CLS drücken und danach die ENTER-Taste!"

Mister Micro führt den Rat von SN7 aus, und der Bildschirm wird gelöscht. „Und nun wollen wir PRINT X probieren." Mister Micro führt aus, was Aleate vorgeschlagen hat. Er tippt

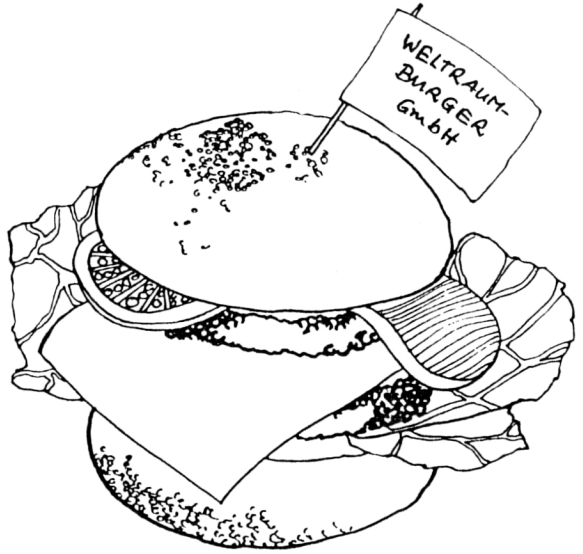
```
PRINT X
```

ein und drückt ENTER.

„Das klappt ja", lacht Aleate, als eine 7 auf dem Bildschirm erscheint. (Hast du es auch probiert?)

„Essenszeit!" Aleate zieht ein paar Weltraumburger aus der Tasche und ein Stück Mondkäse.

„Ach, herrlich", Mister M läuft das Wasser im Munde zusammen. Sie setzen sich und verzehren ihre Weltraumburger.

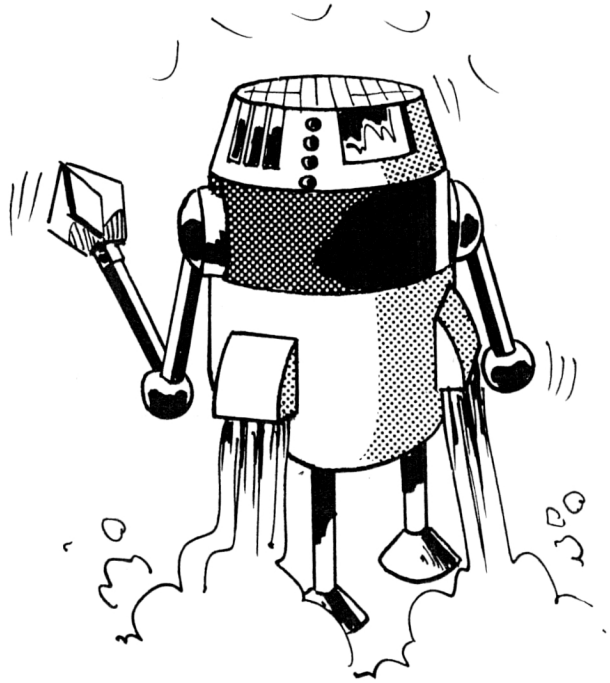


„Essen! Essen! Schrecklich! Schrecklich!" piepst SN7 und rast verzweifelt hin und her. Der Gedanke an Weltraumburger in seinen elektrischen Leitungen ist einfach zu viel für SN7!

Vielleicht möchtest du dir jetzt das Programm „PRINTES" (auf Seite B der Kassette) anschauen. Damit kannst du die Anwendung der Anweisung PRINT üben.

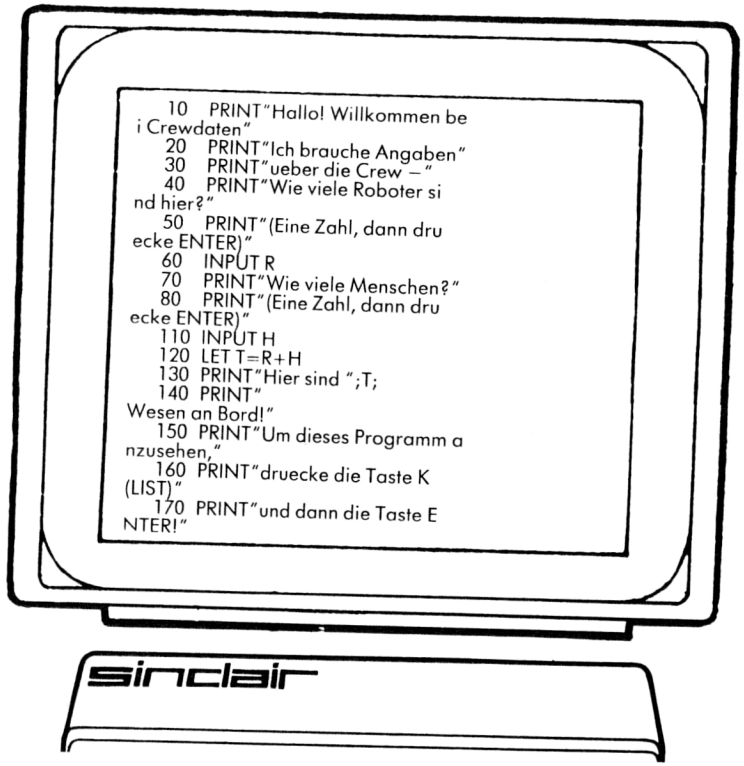
EPISODE 2

Unsere Helden ändern erfolgreich ein Programm



„Nun wollen wir uns das Programm doch noch einmal ansehen", sagt Mister Micro. Er tippt auf die Taste mit der Aufschrift LIST und drückt dann ENTER. Wie er erwartet hat, erscheint nun das Programm auf dem Bildschirm.

Wenn du deinen Computer nach dem Kapitel 1 abgeschaltet hast, mußt du das Band wieder zurückspulen und warten, bis es angehalten hat. Dann tippst du LOAD"". Nun die Taste ENTER drücken und dann warten, bis der Computer fertig ist. Wenn er OK anzeigt, schreibst du LIST und drückst dann ENTER.



„Was bedeuten aber diese Zahlen am Anfang jeder Zeile?" fragt Aleate. Mister Micro denkt gerade über das gleiche Problem nach.

Dann verkündet er strahlend: „Mir scheint, jede Zeile ist eine Anweisung an den Computer. Und zusammen bilden die Zeilen eine Liste von Anweisungen. Deshalb das Wort LIST, mit dem wir den Computer dazu gebracht haben, die Liste der Anweisungen darzustellen."

„Dann ist es wohl möglich, beliebige Listen in den Computer einzugeben", meint Aleate.

„Ja schon, aber nicht alle Listen sind auch Anweisungslisten", erwidert Mister M, „eine Einkaufsliste zum Beispiel enthält keine Anweisungen, sondern sie ist einfach eine Auflistung von Dingen. Wenn es sich um eine Liste mit Anweisungen handelt, die der Computer verstehen kann, dann nennen wir diese Liste ein Programm."

„Aber was ist denn mit den Zahlen?" unterbricht Aleate.

„Geduld." Mister M fährt fort: „Eine Anweisungsliste muß nun immer in einer bestimmten Reihenfolge geschrieben werden. So haben zum Beispiel die Anweisungen zum Essen eines Weltraumburgers

ISS DEINEN WELTRAUMBURGER
ENTFERNE DIE VERPACKUNG

eine bestimmte Reihenfolge. Wenn du die Reihenfolge vertauschst, wird dein Weltraumburger bestimmt nicht schmecken. Wie sagen wir also dem Computer, in welcher Reihenfolge er etwas ausführen soll? Ganz einfach. Gib jeder Anweisung eine Nummer wie z. B. 500 oder 300. Der Computer führt die Anweisung mit der niedrigsten Nummer zuerst aus und arbeitet sich so zu der Anweisung mit der höchsten Nummer vor."

„Dann", sagt Aleate, „können wir in dem Beispiel mit dem Weltraumburger schreiben:

500 ISS DEINEN WELTRAUMBURGER
300 ENTFERNE DIE VERPACKUNG

und der Computer würde erst Zeile 300 ausführen und dann Zeile 500."

„Genau", antwortet Mister M, „und wir könnten leicht eine weitere Anweisung zwischen die beiden Anweisungen einschieben, wie

```
400 VERGISS NICHT, SOSSE AUF DEN
    WELTRAUMBURGER ZU GEBEN
```

Der Computer würde dann die ganze Liste in die richtige Reihenfolge sortieren und ein schönes sauberes Programm bilden:

```
300 ENTFERNE DIE VERPACKUNG
400 VERGISS NICHT, SOSSE AUF DEN
    WELTRAUMBURGER ZU GEBEN
500 ISS DEINEN WELTRAUMBURGER
```

Einfach, nicht wahr?"

„Ja, schon", Aleate ist etwas verwundert, „aber wäre es nicht einfacher gewesen, die Zeilen einfach mit 1, 2, 3 usw. zu numerieren?"

„Zuerst scheint das zwar logischer zu sein", antwortet Mister Micro, „doch wenn wir unsere Anweisungen mit 1, 2, 3 usw. bezeichnet hätten, wie könnten wir dann zusätzliche Anweisungen einfügen? In unserem Beispiel haben wir unsere ersten beiden Anweisungen mit 300 und 500 bezeichnet. So konnten wir später eine weitere Anweisung zufügen, nämlich 400:

```
400 VERGISS NICHT, SOSSE AUF DEN
    WELTRAUMBURGER ZU GEBEN
```

Wenn wir nun die ersten beiden Anweisungen mit 1 und 2 numeriert hätten, dann könnten wir die zusätzliche Anweisung nicht mehr dazwischenschieben!"

„Also", sagt Aleate, „weil das uns auf dem Band gegebene Programm in großen Schritten numeriert ist, können wir Extrazeilen einfügen, wo wir wollen. Probieren wir's mal aus." Sie tippt:

```
15 PRINT"DAS MACHT SPASS"
```

und drückt ENTER. (Das solltest du jetzt auch mal probieren.)

„Jetzt soll der Computer mal dieses Programm ausführen. Ich denke, dafür muß ich RUN tippen (weil das im Englischen soviel wie „laufen" heißt)." Sie tippt den Befehl

```
RUN
```

und drückt ENTER.

Auf dem Bildschirm erscheint:



„Sehr schön. Aber viel Sinn ergibt das ja nicht gerade", sagt Mister Micro.

„Mister Micro! Das ist doch unglaublich. Ich könnte verzweifeln. Da ist es uns nun gelungen, das Programm dieses Raumschiffs zu ändern, und Sie können nur kritisieren!"

„Tut mir furchtbar leid", entschuldigt sich Mister M. „Könnten wir denn jetzt tatsächlich ein neues Programm erstellen?" fragt er.

Aleate antwortet: „Warum eigentlich nicht? Nur müssen wir noch herausfinden, wie wir das alte Programm loswerden, bevor wir ein neues eingeben. Aber wie wohl?"

„NEW, NEW, das ist das Wort", gibt SN7 einen seiner seltenen Kommentare.

„Sei still!" Aleate mag laute Unterbrechungen nicht. SN7 rollt in eine Ecke. Mister Micro tritt vor und tippt auf die Taste mit der Aufschrift NEW auf der Tastatur. Dann drückt er ENTER. Auf dem Bildschirm steht nun zu lesen:

2 Variable not found, 60 : 1

„Oh!" sagt Mister M. „Was bedeutet das wohl?"

„Dumm! Dumm!" ruft SN7 aus seiner Ecke. „Beantworte doch die Frage erst! Erst."

„Ach so – er erwartet noch immer die Antwort auf seine Frage – die Anzahl der Roboter", sagt Mister Micro.

„Aber das haben wir ihm doch schon gesagt!" ruft Aleate aus.

„Richtig", sagt Mister M, „er scheint es aber zu vergessen, wenn wir RUN schreiben. Moment mal", er kratzt sich am Kopf. „Na, klar! Wenn wir RUN tippen, sagen wir dem Computer, er soll das Programm wieder ablaufen lassen,

egal was war. Selbst wenn er schon über die Besatzung Bescheid weiß, fragt er uns erneut, weil wir ihm das so befohlen haben!"

„Richtig", sagt Aleate, „und wir haben eine falsche Antwort gegeben, der Computer hat nämlich eine Zahl erwartet. Das mußte ja zu einem Fehler führen!" „Genau so ist es", ruft Mr. M, „aber jetzt können wir endlich NEW eingeben." Er drückt die Taste mit der Aufschrift NEW und danach die ENTER-Taste.

Auf dem Bildschirm erscheint

© 1982 Sinclair Research Ltd

Als nächstes schreibt er

LIST

und drückt wieder ENTER. Das Programm wird nicht wieder auf dem Bildschirm aufgelistet: Es ist verschwunden!

„Siehst du wohl! Siehst du wohl!" ruft SN7. „Wieder recht gehabt! Ich sagte ja, NEW macht's! Wieder recht gehabt!" Das stimmt, also beschwert sich niemand.

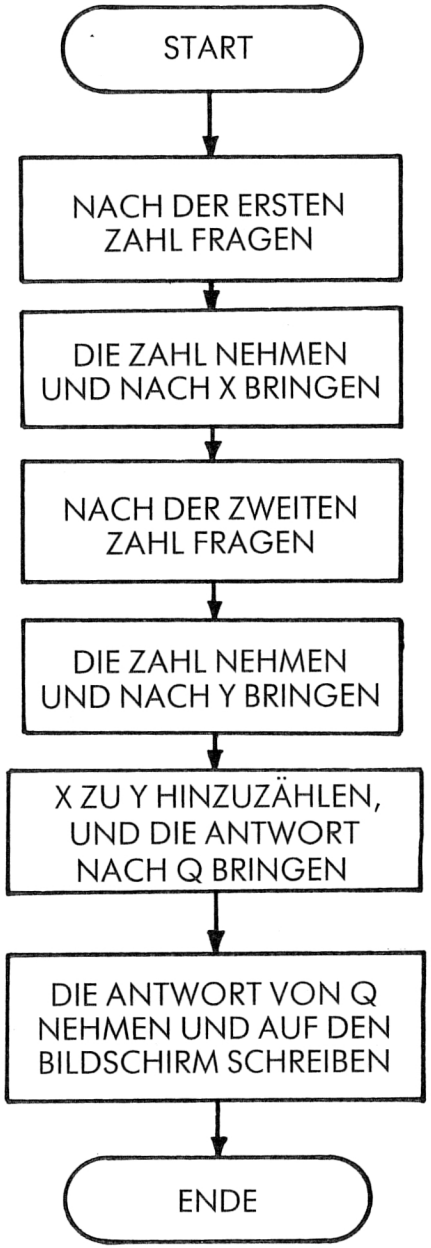
„O.K., entschuldige bitte, SN7. Du hattest tatsächlich recht", sagt Aleate. „Nun wollen wir also ein Programm schreiben. Wir wollen's ganz einfach machen, vielleicht erstmal nur zwei Zahlen zusammenzählen. Darf ich mal Ihren Schreibblock borgen, Mister Micro?"

„Na klar." Mister M reicht Aleate Schreibblock und Bleistift.

„Ich male jetzt mal eine einfache Skizze von dem, was das Programm tun soll."

Sie beginnt zu zeichnen.

Und so sieht die Skizze aus:



„Jetzt können wir dies in ein Programm umwandeln", fährt sie fort. „Ich will Zeilennummern in Schritten von jeweils zehn verwenden, damit ich immer noch weitere Zeilen einfügen kann, wenn ich sie brauche." Und hier sind die Programmzeilen, die Aleate schreibt, mit ihren Bemerkungen zu den einzelnen Zeilen.

„Zuerst fragen wir mit PRINT
nach der ersten Zahl."

```
10 PRINT"Wie lautet die erste Zahl"
```

(Hast du es auch eingetippt? Drücke nach jeder Programmzeile ENTER!)

„Dann verwenden wir INPUT,
damit der Computer die Zahl
an den Speicherplatz X bringt."

```
20 INPUT X
```

„Als nächstes fragen wir
nach der zweiten Zahl."

```
30 PRINT"Wie lautet die zweite Zahl"
```

„Dann verwenden wir wieder INPUT,
damit der Computer die Zahl
an den Speicherplatz Y bringt."

```
40 INPUT Y
```

„Jetzt haben wir also beide
Zahlen; die zählen wir nun
zusammen und bringen die
Antwort nach Q."

```
50 LET Q=X+Y
```

„Nun brauchen wir nur noch die Antwort mit PRINT schreiben zu lassen."

```
60 PRINT Q
```

„Das ganze Programm sieht dann so aus:"

```
10 PRINT"Wie lautet die erste Zahl"
20 INPUT X
30 PRINT"Wie lautet die zweite Zahl"
40 INPUT Y
50 LET Q=X+Y
60 PRINT Q
```

Aleate tippt RUN und drückt ENTER, damit das Programm ablaufen kann.

Auf dem Bildschirm erscheint:

```
Wie lautet die erste Zahl
```

Aleate drückt auf die Taste mit der Zahl 5 und dann auf ENTER.

Der Computer fragt:

```
Wie lautet die zweite Zahl
```

Aleate tippt 7 und drückt dann wieder die ENTER-Taste.

Der Computer antwortet mit

```
12
```

„Das hat funktioniert!" ruft Mister M. „Du hast's geschafft!"

Aleate lacht, und nachdem sie eine Weile nachdenklich auf den Bildschirm geschaut hat, sagt sie: „Wir könnten das noch besser verständlich machen, wenn wir eine Meldung eingeben würden, wie z. B.:

```
60 PRINT"DIE SUMME IST
";Q
```

Aleate tippt das ein und läßt das Programm laufen, wobei sie diesmal eine 6 für die erste Zahl und eine 8 für die zweite eingibt. Der Computer antwortet mit der Meldung:

```
DIE SUMME IST 14
```

„Klasse!" ruft Mister Micro, „aber warum hast du den Strichpunkt zwischen die Meldung und das Q auf Zeile 60 gesetzt?"

„Ich denke, daß der Strichpunkt zum Trennen der Meldung von der Zahl verwendet wird", schlußfolgert Aleate.

„Hm. Ich glaube, du hast wahrscheinlich recht", bemerkt Mister Micro. „Aber ich kann mir vorstellen, daß er noch mehr bedeutet. Wenn du dir die anderen PRINT-Zeilen in CREWDATEN (10 bis 50, 70, 80 und 130 bis 170) ansiehst, kannst du feststellen, daß einige gleich hinter dem zweiten Anführungszeichen aufhören und der Computer dann mit einer neuen Zeile beginnt. Ich denke, der Strichpunkt sagt ihm, daß er auf derselben Zeile weiterschreiben soll."

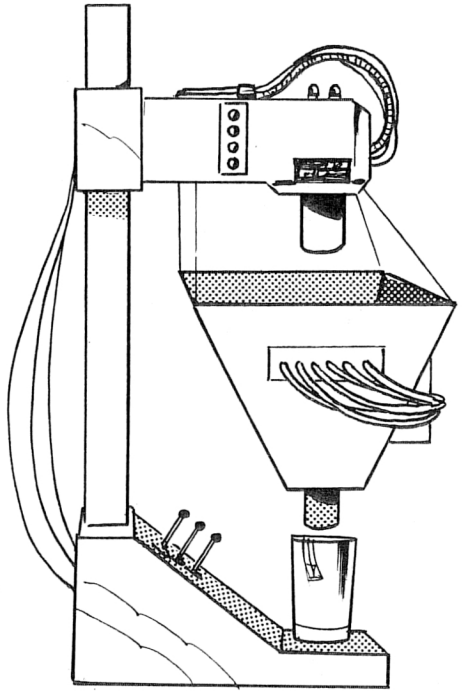
Sie experimentieren mit den PRINT-Zeilen und stellen fest, daß es stimmt, was Mister M vermutet hat.

(Vielleicht möchtest du das auch einmal ausprobieren und auch mit dem Zahlen-Eingabeprogramm ein bißchen experimentieren.)

An dieser Stelle findest du es vielleicht ganz nützlich, dir einmal das Programm NUMMERN (auf der B-Seite der Kassette) anzusehen. Damit kannst du das Eingeben von Zeilennummern in Programme praktisch üben.

EPISODE 3

Die Getränkemaschine



„Was nun?" fragt Mister Micro.

„Sollen wir mal nachsehen, ob noch etwas anderes auf dem Band ist?" fragt Aleate.

„Warum nicht?" antwortet Mister M. Er drückt die Taste LOAD und danach "", gefolgt von ENTER. Dann schaltet er den Recorder ein.

(Wenn du dein Band zurückgespult hast, solltest du stattdessen folgendes eintippen:

LOAD"GETRAENKE"

und dann ENTER drücken und den Recorder einschalten. Wenn der Computer fertig ist (erkennbar an der Meldung 0 OK 0 : 1 unten auf dem Bildschirm), dann drücke auf die Taste

RUN

und danach ENTER.

Sofort erscheint die Meldung

Wenn du die in der Ecke
stehende Getraenkemaschine
benutzen willst, musst du
mir das Codewort sagen

am Bildschirm.

„Ach das ist also diese komische Maschine da", bemerkt Mister M.

„Ja, ein schönes eiskaltes Getränk wäre gerade das Richtige für mich", Aleate hat Durst. „Aber wie können wir das Codewort finden?"

„SN7 muß uns helfen", sagt Mister M, „komm her, schließe dich wieder an."

SN7 rollt heran und schließt sich wieder an die Roboter-Schnittstelle an. Ein paar Minuten lang hört man nur Pieptöne und dann: „Hinweis! Hinweis! Hinweis! Es ist ein Wort

mit sechs Buchstaben – etwas aus Wasserstoff und Sauerstoff!"

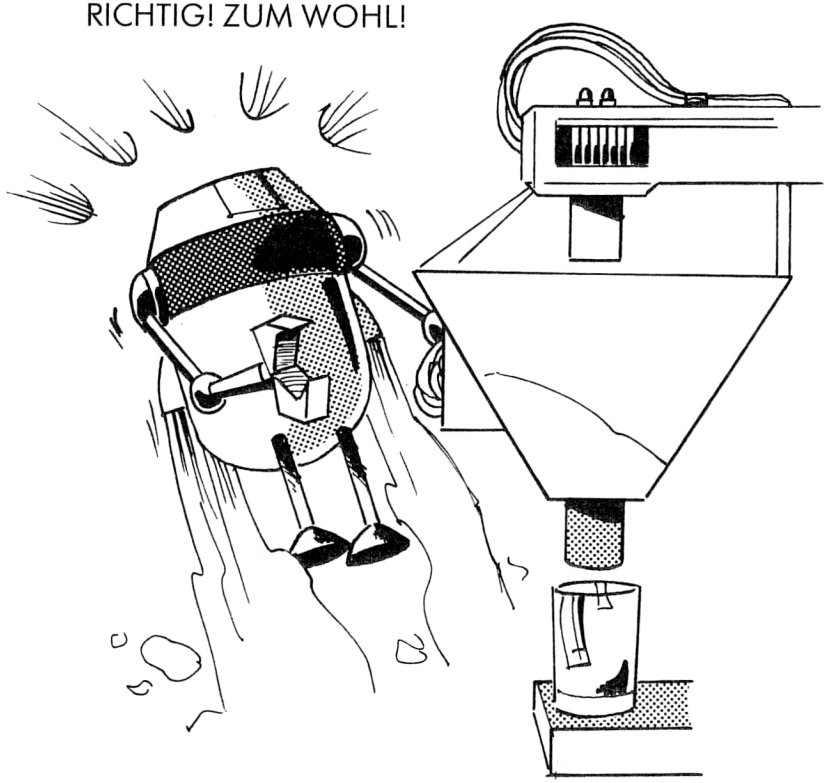
„Du weißt, was es ist, nicht wahr?" ruft Aleate, „sag's uns doch! Wir sind durstig!"

(Sieh, ob du es erraten kannst. Tippe deine Lösung ein, und drücke dann ENTER. Wenn es richtig ist, wird der Computer dir das mitteilen; wenn nicht, stellt er dir die Frage erneut.)

„O.K.", antwortet der Roboter und zieht den Stecker wieder vom Modul ab, „Wasser!"

Sie tippen das Wort Wasser ein und drücken ENTER. Der Computer antwortet mit

RICHTIG! ZUM WOHL!



(Achte bitte darauf, daß du das Wort Wasser genau so eingibst, wie es hier steht, also ein großes W und den Rest in kleinen Buchstaben. Würdest du WASSER oder wasser eingeben, so würde der Spectrum diese Antwort nicht als richtig anerkennen, er ist da sehr penibel. Das große W erreichst du, indem du CAPS SHIFT niederdrückst und dann auf die Taste W tippst.)

Sie erfrischen sich mit einem Getränk. Nur SN7 nicht, der umherrollt und leise vor sich hin spricht, damit ihn ja keiner hört: „Flüssigkeit. Kann ich einfach nicht ausstehen."

„Wie dieses Programm wohl funktioniert?" fragt Mister Micro ein wenig neugierig. „Wir könnten

LIST

tippen und es uns einmal ansehen." Aleate schreibt LIST und drückt ENTER. Sofort erscheint das Programm auf dem Bildschirm:

```
10 PRINT"Wenn du die in der Ecke"
20 PRINT"stehende Getraenkemaschine"
30 PRINT"benutzen willst, musst du"
40 PRINT"mir das Codewort sagen"
50 INPUT C$
60 IF C$="Wasser" THEN GO TO 90
70 PRINT"FALSCH"
80 GO TO 10
90 PRINT"RICHTIG! ZUM WOHL!"
```

„Also, PRINT und INPUT kennen wir ja schon, aber GO TO in Zeile 80, das ist neu", sagt Aleate.

„Ja", erwidert Mister M, „GO TO 10; ich könnte mir denken, daß das GEHE NACH (GO TO) Zeile 10 des Programms bedeutet. Also, wenn das Programm zu Zeile 80 kommt, springt es auf Zeile 10 zurück und läuft von dort aus wieder weiter." Er kritzelt ein paar Notizen auf seinen Block.

„Da ist was Komisches in Zeile 50", Aleate weist auf die Zeile. „Wir kennen INPUT X, INPUT Y oder INPUT Q, zum Beispiel. Aber warum heißt es C$ und nicht C? C, das kann ich verstehen, aber C$? Was bedeutet das wohl?"

„Ich denke, das (Dollar-)Symbol $ soll dem Computer sagen, daß C$ nicht ein normaler Speicherplatz ist. Das muß irgendwas Besonderes sein." Mister Micro geht an die Tastatur und tippt

PRINT C$

Der Computer antwortet mit

Wasser

„Das ist das Codewort!" ruft Mister M aus. „C$ ist etwas Besonderes, weil es Wörter und nicht Zahlen speichert!" Er lehnt sich über die Tastatur und schreibt:

```
LET Q$="Hitze"
PRINT Q$
```

Da du nach jeder Zeile ENTER drücken mußt, sieht es auf deinem Bildschirm so aus:

LET Q$="Hitze"

Nach dem Drücken von ENTER erscheint 0 OK, 0 : 1. Du gibst nun ein:

PRINT Q$

und

ENTER.

Es erscheint das Wort „Hitze" und wieder die Meldung 0 OK, 0 : 1.

Um das zu vermeiden, kannst du auch die beiden Befehle in eine Reihe schreiben. Dann mußt du sie aber durch einen Doppelpunkt trennen:

```
LET Q$="Hitze" : PRINT Q$
```

Der Computer schreibt auf den Bildschirm:

```
Hitze
```

„Na, das scheint ja zu klappen", bemerkt Aleate.

„Ja", antwortet Mister M. „Da sind auch einige neue Wörter in Zeile 60:

```
60 IF C$="Wasser" THEN GO TO 90
```

GO TO 90 – das ist genau wie GO TO 10 in Zeile 80."
„Zeile 60 ist, glaube ich, gar nicht so schwer zu verstehen", wirft Aleate ein. „IF ist Englisch und heißt so viel wie WENN, und THEN ist DANN. Das ganze ist also praktisch eine Bedingung, z. B.

```
IF (WENN) ich hungrig bin, THEN (DANN) esse ich
IF (WENN) es kalt ist, THEN (DANN) ziehe ich den
Mantel an
IF (WENN) das, was in C$ steht, Wasser ist, THEN
(DANN) GO TO Zeile Nummer 90
```

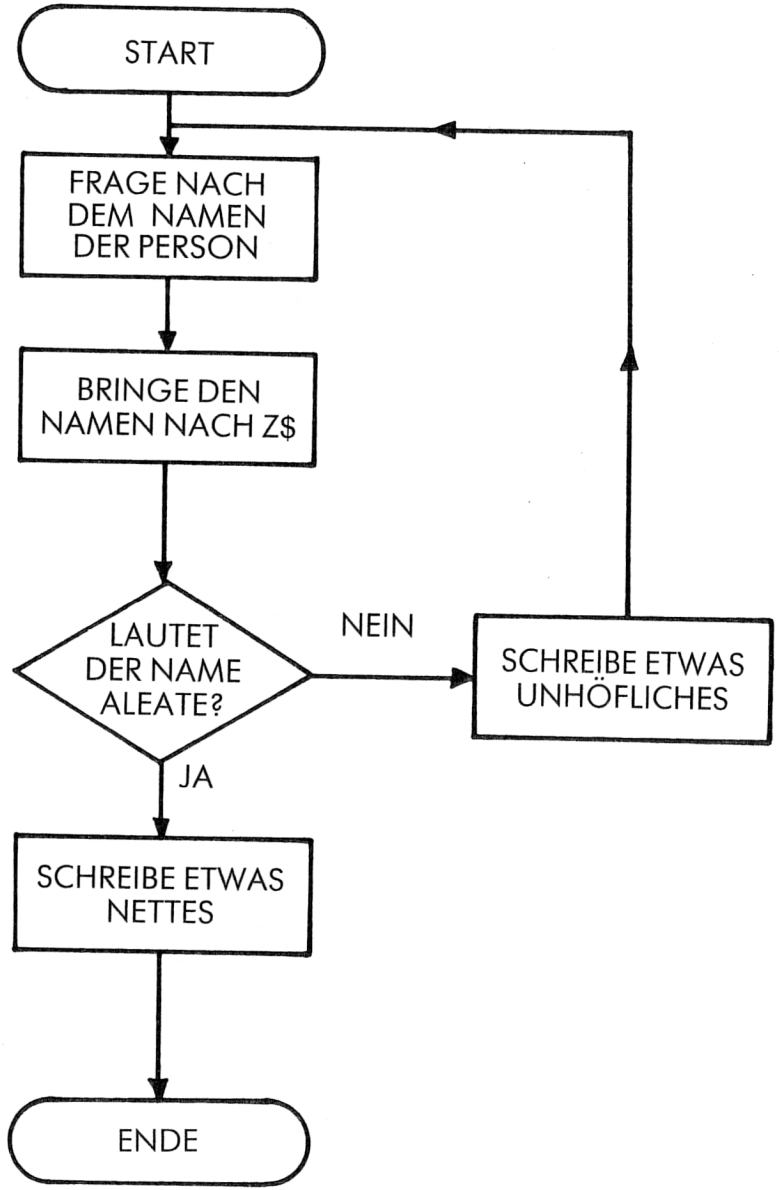
Wir können also sagen:

```
IF C$="FALSCH" THEN LET X=4
oder
IF Q=18+Y THEN PRINT"DAS HAT GEKLAPPT"
```

So etwas kann der Computer verstehen. Wir könnten das Programm zum Beispiel auch Namen von Leuten erkennen und GUTEN TAG oder HAU AB sagen lassen."

„Kannst du wohl so ein Programm schreiben?" fragt Mister M.

„Klar." Aleate macht sich eine Skizze, wie das Programm wohl aussehen würde. Und so sieht diese Skizze aus:



„So, und hier ist das Programm", sagt Aleate. Dies sind die Programmzeilen mit den Erklärungen von Aleate. Tippe jede Zeile ein, und drücke jedesmal ENTER.

(Falls du noch ein Programm geladen hast, gib zuerst NEW ein.)

„Zuerst müssen wir nach dem Namen der Person fragen."

```
10 PRINT"Wie heisst du?"
```

„Nun geben wir die Antwort in einen Speicherplatz, den wir Z$ nennen."

```
20 INPUT Z$
```

„Dann vergleichen wir den Namen mit ALEATE, also meinem Namen, und wenn ich es bin, gehen wir nach Zeile 100. Die schreibe ich später."

```
30 IF Z$="ALEATE" THEN GO TO 100
```

„Zu Zeile 40 kommen wir nur, wenn Z$ nicht ALEATE war, nun können wir also unsere unhöfliche Meldung schreiben."

```
40 PRINT"Hau ab"
50 PRINT Z$
```

„Nun springen wir zum Start unseres Programms zurück."

```
60 GO TO 10
```


„Wenn Z$ ALEATE lautete, springen wir zu Zeile 100, wo wir eine nette Meldung ausgeben."

```
100 PRINT"Guten Tag"
110 PRINT Z$
```

So, das ganz Programm sieht also wie folgt aus:

```
10  PRINT"Wie heisst du"
20  INPUT Z$
30  IF Z$="ALEATE" THEN GO TO 100
40  PRINT"Hau ab"
50  PRINT Z$
60  GO TO 10
100 PRINT"Guten Tag"
110 PRINT Z$
```

„Jetzt wollen wir das Programm ausführen." Aleate tippt

```
RUN
```

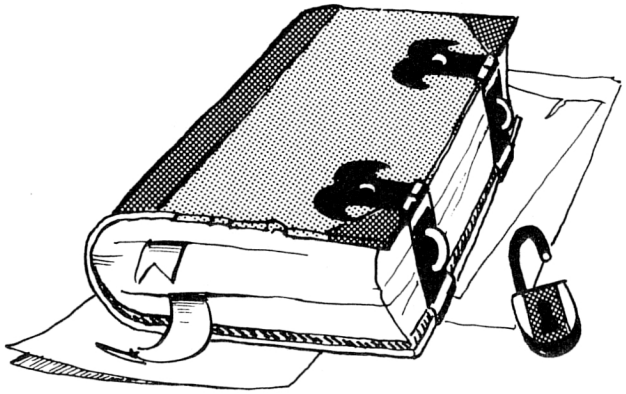
und drückt ENTER.

```
Wie heisst du
"L"
```

reagiert der Computer. „Fehlen da nicht die Zeilennummern 70, 80 und 90?" fragt Mister M. „Nein", Aleate weiß, daß nicht alle Nummern vorhanden sein müssen, „die Hauptsache ist, daß sie in aufsteigender Reihenfolge auftreten."

EPISODE 4

Eine aufregende Entdeckung



„Sieh! Sieh! Sieh!" SN7 dreht sich vor Aufregung im Kreise. „Ein Buch! Ein Buch!"

Sie gehen zu SN7, um zu sehen, was er entdeckt hat. Es sind die zerfledderten Überreste eines Buchs! Mister Micro hebt es auf und bläst den Staub ab. Er schlägt das Buch auf.

Drinnen steht in einer Handschrift, die mindestens 1000 Jahre alt sein muß, folgendes geschrieben: „Dieses Buch ist das Eigentum von Kapitän Barbara Winter."

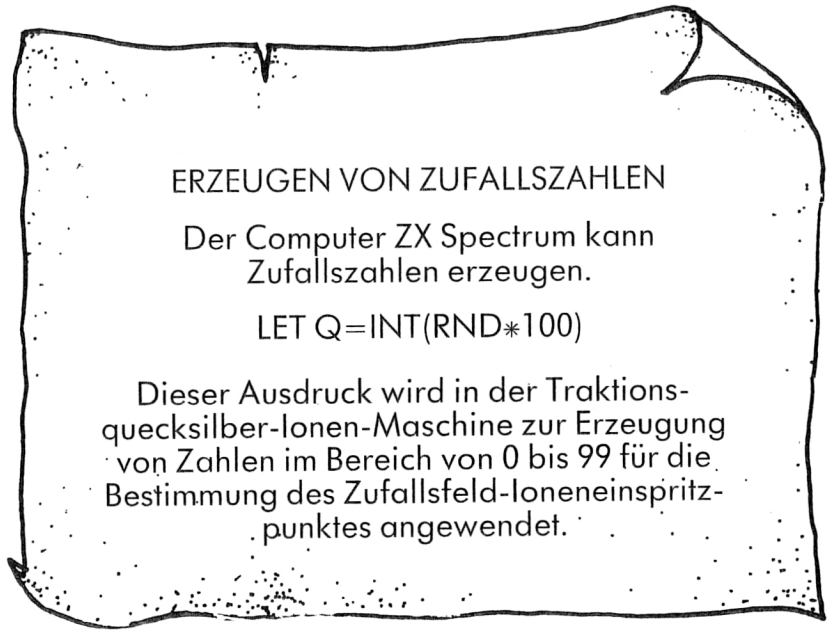
„Einfach faszinierend", murmelt Mister M. „Schau nur, da steht ein Datum: 2184 A.D. Du liebe Güte! Welches Jahr haben wir denn jetzt bloß?"

„Wir sind scheinbar in die Zukunft gereist", sagt Aleate, „mir scheint, Sie haben mehr erfunden als nur eine einfache Beam-Maschine!"

Mister M denkt ein Weilchen darüber nach. „Nun, ja", meint er, „auf jeden Fall muß dies hier ein altes Raumschiff von der Erde sein! Ich frage mich, warum es wohl verlassen wurde?"

„Vielleicht wegen der Großen Pest von 229 S.W.K. (Standard-Weltraumkalender), das heißt im Jahre 2186 A.D.", sagt Aleate erklärend.

Mister Micro blättert in dem Buch. „Dies sieht wie ein Handbuch für das Computersystem aus, das wir hier benutzen. Leider sind einige Seiten fast zerstört, und es fehlen auch welche. Na, wir können's ja benutzen, so gut es eben geht", sagt er. „Wollen wir uns doch mal dieses Stück hier ansehen mit der Überschrift ERZEUGEN VON ZUFALLSZAHLEN." Sie blicken auf die Seite. Und hier kannst du diese Seite sehen:



„Also, nach so einem Test ist der Ausdruck

```
LET Q=INT(RND*100)
```

ja richtig einfach!" bemerkt Mister M.

Aber wie bekommen wir die Befehle INT und RND auf den Bildschirm? Sie stehen ja in grüner Schrift über den Tasten „R" und „T". So etwas hatten wir bis jetzt noch nicht."

CAPS SHIFT und SYMBOL SHIFT gleichzeitig drücken, ganz einfach, ganz einfach", schnarrt SN7, der es mal wieder besser weiß.

„Es ist einen Versuch wert", meint Mr. M. Er drückt gleichzeitig auf CAPS SHIFT und SYMBOL SHIFT. Das blinkende „K" unten auf dem Bildschirm verwandelt sich in ein „E". Nun tippt er auf die Taste mit dem „R", über der in Grün INT steht. Und tatsächlich, das Wort INT erscheint auf dem Schirm.

„Siehst du, siehst du", freut sich SN7. „Ja", sagt Mr. Micro, „du hattest recht, wie immer".

Mr. M löscht das Wort INT auf dem Bildschirm, indem er auf CAPS SHIFT drückt und dann die Taste DELETE betätigt.

Jetzt wollen wir die Formel aber endlich ausprobieren!", sagt Aleate. Sie tippt

```
NEW
```

und drückt dann ENTER. Dann schreibt sie das folgende Programm:

```
10 LET Q=INT(RND*100)
20 PRINT Q
30 GO TO 10
```

(Schreibe das bitte auch einmal – vergiß nicht, NEW zu tippen und ENTER zu drücken!)

Dabei gibt Aleate folgende Erklärungen zu jeder Zeile des Programms:

„Zeile 10 ist einfach
vom Buch abgeschrieben.
Nur habe ich eine Zeilennummer
davor geschrieben, weil sie nun
Teil meines Programms ist."

```
10 LET Q=INT(RND*100)
```

„Diese Zeile schreibt den Wert
von Q, damit wir sehen
können, was die Anweisung in
Zeile 10 an Speicherplatz Q
gespeichert hat."

```
20 PRINT Q
```

„Dann springen wir von
Zeile 30 auf Zeile 10
zurück, um den Prozeß
noch einmal ablaufen
zu lassen."

```
30 GO TO 10
```

Aleate tippt RUN (und drückt ENTER). Auf dem Bildschirm erscheinen viele Zahlen. Nachdem der Bildschirm mit Zahlen vollgeschrieben ist, erscheint die Meldung – scroll? – am unteren Rand. Aleate drückt die Y-Taste, um die Frage mit „ja" (englisch „yes") zu beantworten. Die Zahlen werden jetzt nach oben geschoben, neue Zahlen erscheinen am unteren Bildschirmrand. „Wie aufmerksam," sagt Aleate. "Der Computer wartet, bis wir die Zahlen gelesen haben und macht dann erst weiter." Aleate drückt die CAPS SHIFT- zusammmen mit der BREAK-Taste, um das Programm anzuhalten.

„Diese Zahlen liegen alle im Bereich von 0 bis 99", sagt Mister M.

„Ja", antwortet Aleate, „der Computer scheint sie einfach zu erfinden, während er das Programm ausführt."

„Genau das ist mit ERZEUGEN VON ZUFALLSZAHLEN gemeint", erklärt Mister Micro. „Der Computer gibt ganz einfach beliebige Zahlen aus. Ach ja, sie liegen allerdings zwischen 0 und 99. Wir könnten ein bißchen mit dieser Anweisung experimentieren: Wir wollen einmal die 100 in 50 umändern; das gibt dann

```
LET Q=INT(RND*50)
```

anstatt

```
LET Q=INT(RND*100)
```

Ich vermute, wir erhalten nun Zahlen im Bereich von 0 bis 49."

„Probieren wir's." Aleate tippt Zeile 10 neu, so daß sie nun lautet:

```
10 LET Q=INT(RND*50)
```

Dann läßt sie das Programm ablaufen.

(Du weißt jetzt bestimmt schon, wie man das macht. Falls nicht: Hier ist es noch einmal erklärt. Du tippst

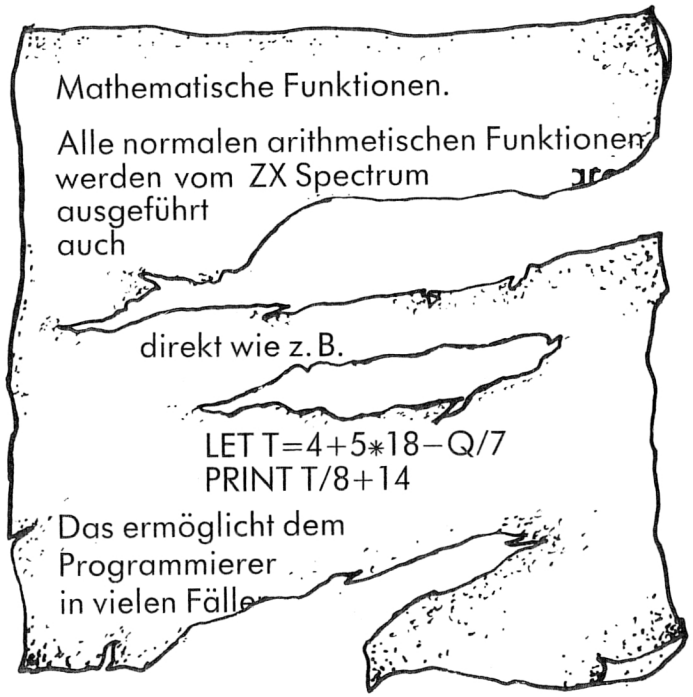
```
RUN
```

und dann drückst du die Taste ENTER.)

Schon erscheinen weitere Zahlen auf dem Bildschirm, die – wie Mister M vermutet hat – im Bereich von 0 bis 49 liegen. (Drücke CAPS SHIFT und BREAK, um das Programm anzuhalten.)

„O.K. Das klappt also. Wo ist denn das Buch?" fragt Mister Micro. Aleate reicht es ihm. „Danke", sagt Mister M. Er beginnt, die zerschlissenen Seiten durchzublättern.

„Hier! Genau das, was ich gesucht habe!" ruft er. „Hier sind alle mathematischen Funktionen: Addieren, Subtrahieren, Multiplizieren und Dividieren."



„Meine Güte, da fehlt aber eine Menge", bemerkt Aleate, die Mister Micro über die Schultern guckt.

„Nehmen wir einmal an, daß

T=4+5*18−Q/7

und

T/8+14

mathematische Ausdrücke sind. Wir wissen ja schon, was + bedeutet, und wir können davon ausgehen, daß − Minus ist." Um das mal schnell zu prüfen, tippt Mister M

PRINT 7−2

und auf dem Bildschirm erscheint

5

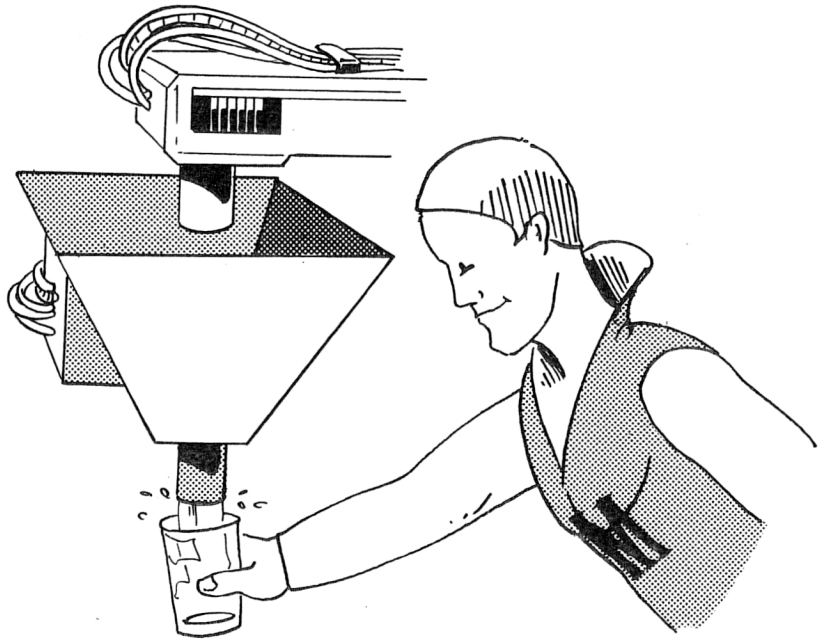
(Hast du auch ENTER gedrückt?)

„Gut. Das geht also. Aber was sollen / und * bedeuten?" Er versucht es mit

PRINT 8/2

und erhält auf dem Bildschirm

4



Dann tippt er

PRINT 10*5

und es erscheint

50

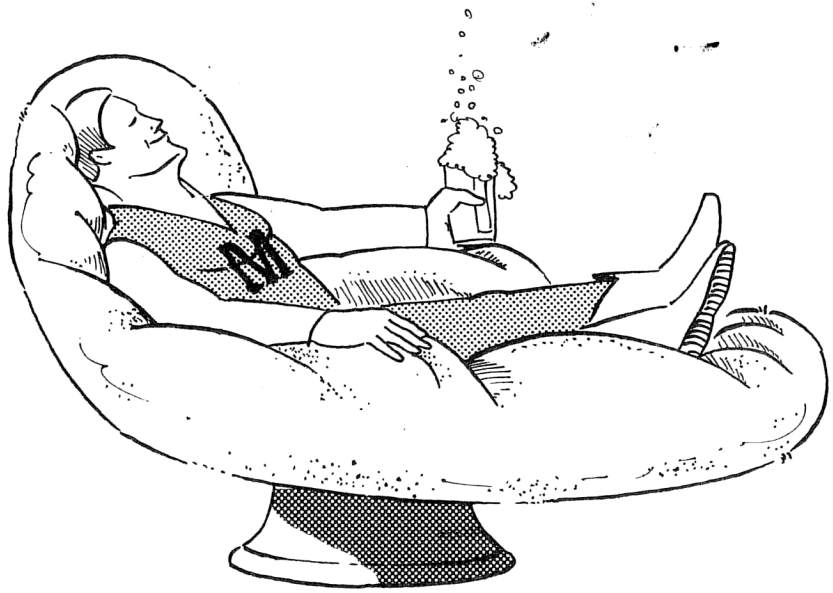
(Kannst du raten, was / und * bedeuten?)

„Wir sehen also, daß / Dividieren und * Multiplizieren ist." Mister Micro kritzelt sich ein paar Notizen auf den Block. Dann gönnen sich Aleate und Mister M ein Glas superdicken Stern-Shake und ruhen sich ein Weilchen aus.

Es ist vielleicht eine gute Idee, jetzt das Programm „STRINGY" ablaufen zu lassen (auf Seite B der Kassette). Damit kannst du schon einmal den Umgang mit dem Befehl TO üben.



Aleate erfindet ein Spiel

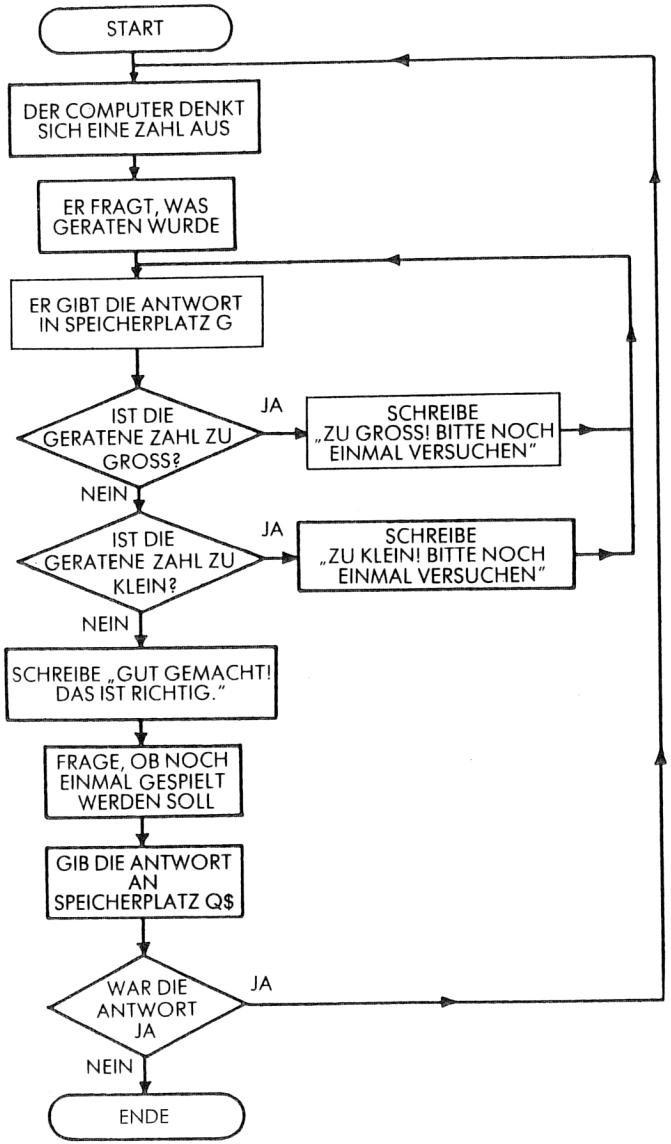


„Ich hab' eine Idee!" freut sich Aleate. „Nur so zum Spaß und auch zum Üben mit dieser Maschine könnten wir doch ein einfaches Ratespiel entwickeln."

„Prima", schmunzelt Mister M, „mir ist nach ein bißchen Unterhaltung."

Er holt sich ein Orange-Soda mit Eis aus der Getränkemaschine und läßt sich auf einem der superbequemen Bordsessel nieder.

„Wir wollen zuerst einen Ablaufplan zeichnen, damit wir sehen, wie das Programm ausschauen wird." Aleate leiht sich ein Blatt Papier von Mister Micros Block und zeichnet einen Plan, der so aussieht:



Sie fährt dann fort: „Jetzt können wir dies in ein Programm umwandeln."

Hier kannst du nun Aleates Programm und ihre Erklärungen dazu sehen. Tippe die Programmzeilen ein, und drücke ENTER am Ende jeder Zeile.

„Zuerst denkt sich der Computer eine Zahl aus und gibt sie an Speicherplatz N."

```
10 LET N=INT(RND*100)
```

„Nun bitten wir die mitspielende Person, eine Zahl zu raten."

```
20 PRINT"Auf welche Zahl tippst du?"
```

„Die von der ratenden Person eingegebene Zahl geht an einen Speicherplatz mit dem Namen G."

```
30 INPUT G
```

„Als nächstes testen wir G, um zu sehen, ob es zu groß ist. Wenn ja, springen wir nach Zeile 110. Diese Zeile habe ich noch nicht geschrieben."

```
40 IF G>N THEN GO TO 110
```

„Was ist das für ein Symbol zwischen G und N?" fragt Mister M.

„Auf meinem Heimatstern, Arfon, haben wir auch so ein Symbol. Das bedeutet größer als", erklärt Aleate.

„G>N bedeutet: G ist größer als N. Daher bedeutet:

```
IF G>N THEN GO TO 110
```

Wenn die Zahl am Speicherplatz G größer als die Zahl am Speicherplatz N ist, dann springe zur Zeile 110."

Aleate fährt fort:

„Wir haben nun geprüft, ob G zu groß ist. Jetzt wollen wir sehen, ob G zu klein ist. Das Symbol für kleiner als ist <. Daher lautet die Zeile:"

```
50 IF G<N THEN GO TO 130
```

„Wenn die geratene Zahl zu groß ist, springen wir von Zeile 40 nach Zeile 110. Ist die geratene Zahl zu klein, springen wir nach Zeile 130. So bleibt uns also ein G übrig, das weder zu groß noch zu klein ist. Das muß dann also genau richtig sein. Wir müssen dies dem Benutzer des Computers sagen:"

```
60 PRINT"Gut gemacht! Das ist richtig!"
```

„Wir könnten das Spiel hier beenden, oder wir können auch fragen, ob die Person noch einmal spielen möchte. Ich habe die zweite Möglichkeit gewählt."

```
70 PRINT"Noch ein Spiel?
```


„Wir nehmen die Antwort, die „ja" oder „nein" lautet, und geben sie an den Speicherplatz mit dem Namen Q$."

```
80 INPUT Q$
```

„Dann fragen wir, ob die Antwort „ja" war. Wenn dies der Fall war, gehen wir zur ersten Zeile des Programms, also Zeile 10, und fangen wieder von vorne an."

```
90 IF Q$="ja" THEN GO TO 10
```

„Wenn die Antwort nicht „ja" war, dann lautet sie „nein". Daher halten wir an."

```
100 STOP
```

„Nun haben wir in Zeile 40 gesagt, daß ein Sprung nach Zeile 110 gemacht werden soll, wenn die geratene Zahl zu groß ist. Wir müssen also zur ratenden Person sagen:"

```
110 PRINT"Zu gross! Bitte nochmal raten!"
```

„Dann gehen wir zur Zeile 20 zurück, damit die ratende Person noch einmal eine Zahl eingeben kann."

```
120 GO TO 20
```

„Wenn die Antwort statt dessen
eine zu kleine Zahl ist, würden
wir von Zeile 50 nach Zeile 130
springen und sagen:"

```
130 PRINT"Zu klein! Bitte nochmal raten!"
```

„Danach gehen wir zur Zeile 20
zurück, damit die ratende
Person noch einmal eine Zahl
eingeben kann."

```
140 GO TO 20
```

„So, und hier ist das ganze Programm:"

```
10  LET N=INT(RND*100)
20  PRINT"Auf welche Zahl tippst du?"
30  INPUT G
40  IF G>N THEN GO TO 110
50  IF G<N THEN GO TO 130
60  PRINT"Gut gemacht! Das ist richtig"
70  PRINT"Noch ein Spiel?"
80  INPUT Q$
90  IF Q$="ja" THEN GO TO 10
100 STOP
110 PRINT"Zu gross! Bitte nochmal raten!"
120 GO TO 20
130 PRINT"ZU KLEIN! Bitte nochmal raten!"
140 GO TO 20
```

„Gut", sagt Mister M von seinem superbequemen Bordsessel aus, „mal los! Probieren wir's aus." Er lehnt sich vor und tippt RUN. Dann drückt er ENTER.

„Ich wette mit dir, Aleate", sagt er, „daß ich die Zahl immer in weniger als acht Versuchen raten kann."

Warum war Mister Micro so sicher, daß er höchstens achtmal zu raten brauchte? Ob er wohl eine Geheimformel hat? Ja. Er hat tatsächlich eine, und hier ist eine sehr komplizierte Erklärung:

„Also", Mister M räkelt sich in seinem Sessel. „Wenn uns das Ratespiel nur sagen würde, ob die Antwort richtig oder falsch war, dann müßten wir 100mal raten, um sicher zu sein, daß wir die richtige Antwort finden. Bei jeder falsch geratenen Zahl wüßten wir, daß die von uns bereits probierten Zahlen falsch sind und nichts weiter.

In unserem Spiel wird uns aber gesagt, ob unsere Zahl zu groß oder zu klein war. Nehmen wir an, wir haben 50 als unsere erste Zahl geraten, und das war falsch. Der Computer hätte uns dann gemeldet: ZU GROSS! In diesem Fall wissen wir also, daß 50 nicht die Zahl war, und wir wissen auch, daß sicher 51 nicht die Zahl war, weil sie größer als 50 ist und 50 ja schon zu groß war. Wir wissen daher, daß 52, 53, 54 und alle anderen Zahlen bis 100 nicht die richtige Antwort sind. Und das ist eine Menge Information!

Wir brauchen jetzt also nur noch zwischen den Zahlen von 1 bis 49 zu raten. Die beste Antwort ist jetzt 25, weil diese Zahl etwa in der Mitte zwischen 1 und 49 liegt. Angenommen, der Computer sagt uns jetzt, daß 25 ZU KLEIN ist. Klar, daß wir jetzt eine Zahl in der Mitte zwischen 25 und 50 wählen müssen. Das ist ungefähr 38. Und so raten wir weiter, indem wir immer ein Zahl nehmen, die in der Mitte der Zahlen liegt, von denen wir wissen, daß sie falsch sind. Das machen wir, bis wir zur richtigen Zahl kommen!"

„Das hört sich furchtbar kompliziert an!" ruft Aleate, „ich glaube, ich rate lieber so, wie's mir einfällt."

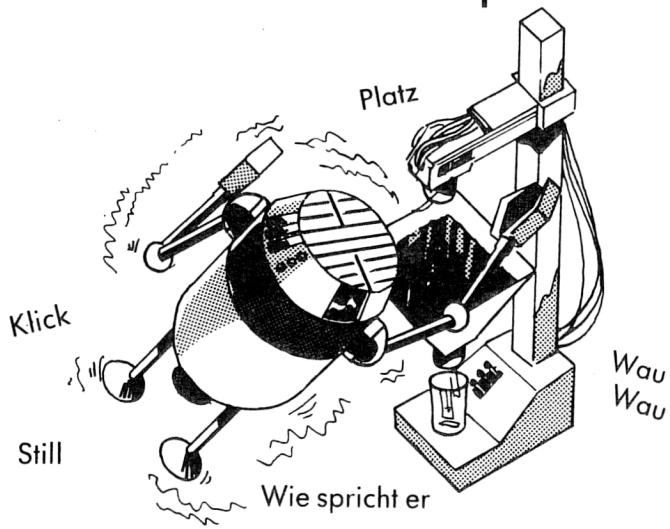
Deshalb gewinnt Mister Micro die meisten Spiele.



Vielleicht möchtest du nun einmal das Programm „RATEN" (auf Seite B der Kassette) ausprobieren. Es führt dir zuerst Aleates Spiel vor und erklärt dir dann, wie es funktioniert.

EPISODE 6

Schleifen knüpfen



Mister M beschäftigt sich noch im superbequemen Bordsessel mit der Theorie von Aleates Zahlenspiel. SN7 versucht so mit der Getränkemaschine zu reden wie ein Hund zum Beispiel einen Fremden verbellen würde, und Aleate blättert im Handbuch von Kapitän Barbara Winter. Sie murmelt vor sich hin: „Also, wenn ich jetzt alle Zahlen von 1 bis 10 schreiben lassen will, könnte ich ein Programm gebrauchen. Zum Beispiel eins wie dieses hier:

```
10 LET C=0
20 LET C=C+1
30 PRINT C
40 IF C<10 THEN GO TO 20
```

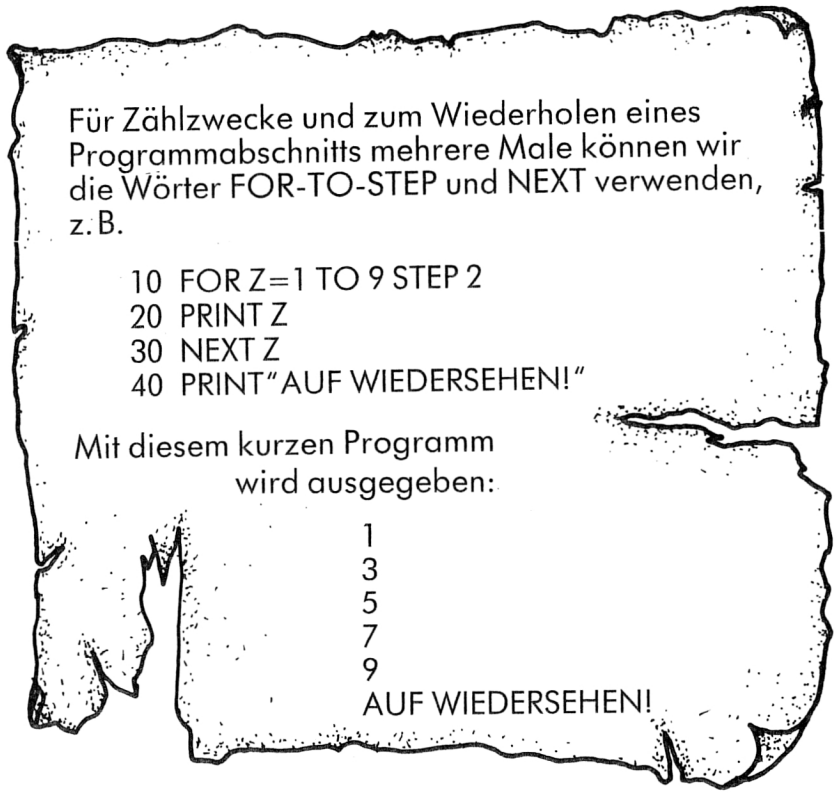
Zeile 20 ist interessant. Das muß bedeuten: Bringe die Zahl nach (Speicherplatz) C und addiere 1 hinzu, gib die Antwort dann zurück an denselben Platz (Speicherplatz C). Ich denke, das will ich mal ausprobieren." Sie schreibt (OHNE Zeilennummern):

```
LET C=24
LET C=C+1
PRINT C
```

Der Computer antwortet:

```
25
```

„Das scheint also zu gehen", murmelt sie. „Was ist das denn?" Sie blickt auf die zerschlissene Seite:



„Also", spricht Aleate vor sich hin, „anstatt bis 10 zu zählen unter Verwendung von LET, IF-THEN und GO TO, könnten wir auch FOR, STEP und NEXT benutzen."

Sie tastet NEW ein und drückt dann ENTER.

Dann:

```
10 FOR Z=1 TO 10 STEP 1
20 PRINT Z
30 NEXT Z
```

Als sie dann RUN schreibt und ENTER drückt, ergibt das Programm

```
1
2
3
4
5
6
7
8
9
10
```

„Gut. Das hat geklappt", bemerkt sie. „Nun wollen wir mal sehen, ob es bis 20 in Schritten von 2 zählen kann: Dazu brauche ich nur die Zeile 10 zu ändern. Ich schreibe einfach eine neue Zeile 10."

„Ja", fügt Mister M hinzu und blickt auf. „Wenn du eine neue Zeile mit der gleichen Zeilen-Nummer wie die alte Zeile schreibst, dann wird die alte Zeile durch die neue ersetzt."

Aleate tippt ein:

```
10 FOR Z=1 TO 20
```

Dann drückt Sie auf die Taste mit der Aufschrift LIST und dann ENTER. Sie erhält:

```
10 FOR Z=1 TO 20
20 PRINT Z
30 NEXT Z
```

Sie gibt wieder RUN ein und drückt ENTER.

Der Computer schreibt:

```
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
```

„Oh je! Ich wollte doch, daß er in Schritten von 2 zählt. Was habe ich nur falsch gemacht? Ach, wie dumm! Ich habe den Befehl STEP vergessen. Aber Moment mal!

Obwohl ich ihm die STEP-Größe nicht angegeben habe, hat er doch in Einern gezählt! Er muß also annehmen, daß man eine Schrittgröße (STEP) von Eins wünscht, wenn man

ihm nichts anderes sagt. Wie klug! Also", fährt sie fort, „um zum Zählen in Zweiern zu kommen, will ich jetzt die richtige Zeile 10 eintippen."

```
10 FOR Z=2 TO 20 STEP 2
```

Diesmal funktioniert es ausgezeichnet.

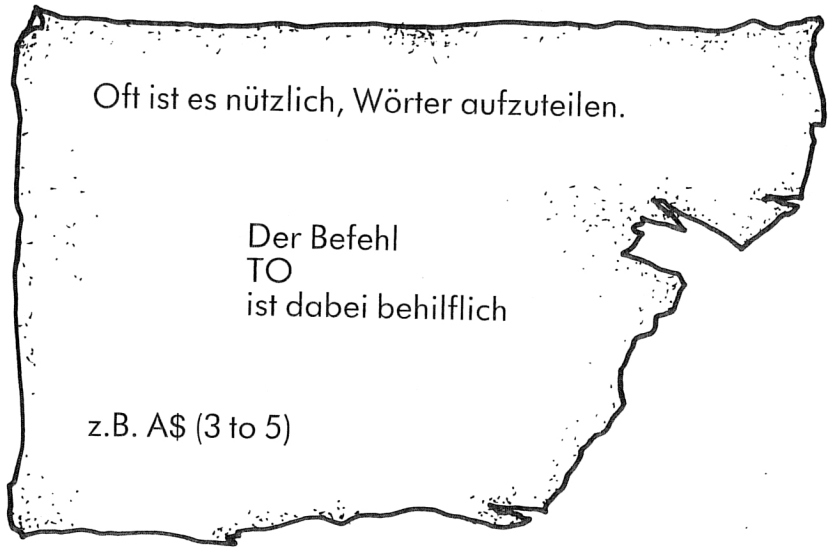
„Wir könnten FOR...NEXT benutzen, um etwas 100mal zu schreiben.

```
10 FOR Z=1 TO 100
20 PRINT"RAUMSCHIFF"
30 NEXT Z
```

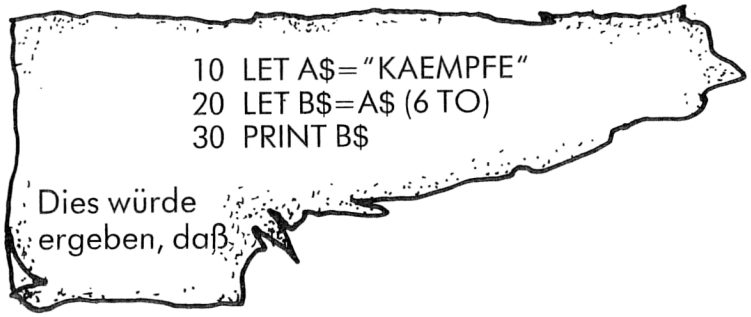
Hiermit würde

```
RAUMSCHIFF
```

einhundertmal geschrieben. Diese Anweisung wäre nützlich zum Zählen." Sie blättert weiter im Handbuch:



„Wie schade! Da fehlt ein Stück", stöhnt sie.



Die nächste Seite fehlt ganz. Aleate muß also experimentieren. Sie gibt das obige Programm ein und läßt es ablaufen. (Das solltest du jetzt auch machen.)

„Hmm", murmelt sie, „FE. Das Programm hat die beiden Buchstaben ganz rechts vom Wort KAEMPFE genommen und sie in B$ gebracht." Sie ändert dann Zeile 20 in:

```
20 LET B$=A$(TO 6)
```

Sie läßt das Programm wieder ablaufen und erhält:

```
KAEMPF
```

„Aha, wenn ich also die 6 durch eine 5 ersetzen würde, bekäme ich KAEMP." Das macht sie dann auch.

```
20 LET B$=A$(TO 5)
```

Und tatsächlich erhält sie, als sie das Programm wieder ablaufen läßt:

```
KAEMP
```

„Das bedeutet also, daß (TO 5) in diesem Falle die ersten fünf Zeichen ganz links von A$ nimmt. Um die ersten 5 Zeichen von links beispielsweise von Q$ herauszunehmen, würden wir daher Q$ (TO 5) schreiben. Hmm..."

Aleate wendet sich wieder dem Buch zu. Da ist nicht mehr

viel über die Anweisung lesbar geblieben. Doch folgendes kann Aleate noch entziffern:

```
40 LET Q$=C$(TO 14)
50 LET T$=C$(4 TO 6)
60 LET F$=Q$+T$
70 LET S$=""
80 PRINT"FERTIGMACHEN ZUM"
```

Sie schreibt ein kurzes Programm, um etwas auszuprobieren. Zuerst aber tippt sie NEW und drückt ENTER. Dann

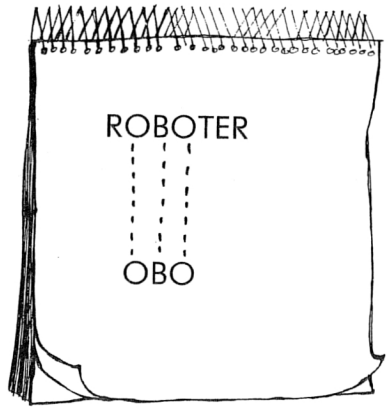
```
10 INPUT A$
20 LET B$=A$(2 TO 4)
30 PRINT B$
```

Als sie dies Programm ablaufen läßt, erscheint auf dem Bildschirm ein "[L]". Sie gibt das Wort ROBOTER ein, auf das der Computer mit

```
OBO
```

antwortet.

Mister M hat das Ratespiel schon eine Weile aufgegeben und mit Interesse zugeschaut. „Aha!" Er zieht seinen Notizblock aus der Tasche und schreibt



„Ja, und?" wundert sich Aleate.

„Na", erklärt Mister M, „OBO ist im Wort ROBOTER enthalten, nicht wahr? Wenn wir mit dem 2. Buchstaben von ROBOTER beginnen und dann bis zum vierten Buchstaben gehen, kriegen wir OBO. Und genau das bedeutet A$(2 TO 4)."

„Das verstehe ich", sagt Aleate, „wenn also ZEIT in A$ gespeichert wäre, lautet der zweite Buchstabe E, und drei Buchstaben von dieser Stelle an würden EIT ergeben. Das ist nützlich, denn damit können wir uns jeweils immer einen Buchstaben eines Worts ansehen." Um ihre Überlegungen zu beweisen, schreibt Aleate ein Programm.

```
10 LET Q$="HALLO SN7"
20 FOR L=1 TO 9 STEP 1
30 PRINT Q$(L TO L)
40 NEXT L
```

„Wie Sie sehen, habe ich die Zeit gut genutzt, während der sie sich mit dem Zahlenspiel befaßt haben, Mister M. Ich hab nämlich im Buch von Kapitän Barbara Winter weitergelesen."

Aleate fährt fort: „Ich verwende die Wörter FOR...NEXT, die ich gelernt habe, und nehme mir jedes Zeichen (Zwischenräume sind auch Zeichen!) von HALLO SN7 einzeln vor. Insgesamt haben wir 9 Zeichen, so daß wir mit Zeichen Nr. 1 beginnen und bis Zeichen Nr. 9 gehen."

„Richtig", stimmt Mister Micro zu, „die Hauptanweisung ist:

```
Q$(L TO L)
```

Damit wird der Text aus Q$, also HALLO SN7, genommen, und es wird jeweils immer der 1., 2., 3., ..., 9. Buchstabe (je nachdem, was L vorschreibt) genommen. Wenn also L eine

1 ist, bekommen wir H, und wenn L eine 6 ist, bekommen wir einen Zwischenraum, und wenn L 9 ist, bekommen wir 7." Aleate läßt das Programm ablaufen, und auf dem Bildschirm erscheint

```
H
A
L
L
O

S
N
7
```

„Na, das ist aber interessant", sagt Aleate, „das Programm hat die Wörter HALLO SN7 genommen und sie in senkrechter Reihenfolge angezeigt. Der Schlüssel liegt hier bei Zeile 30! Jedesmal, wenn das Programm an diese Zeile kommt, bewegt es den Cursor (die Schreibmarke) jeweils um eine Zeile nach unten. Wenn wir diese Zeile durch Hinzufügen eines Semikolons (;) hinter der letzten Klammer ändern, dann sollten die Buchstaben alle auf der gleichen Zeile am Bildschirm erscheinen." Aleate probiert das aus.

„Da. Es geht."

Nur mal so zum Spaß ändert Aleate Zeile 30 erneut, so daß sie lautet:

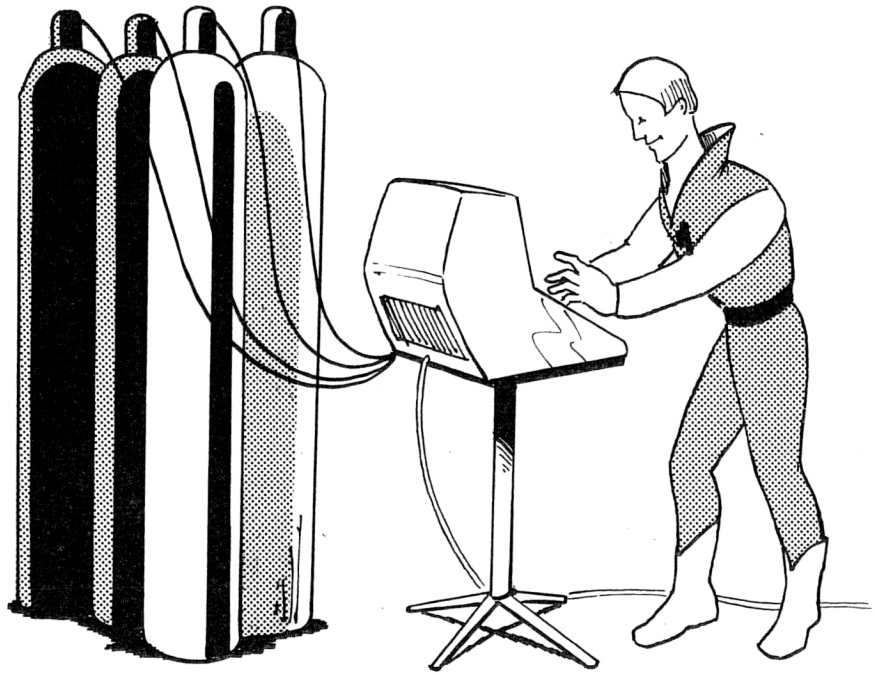
```
30 PRINT Q$(L TO)
```

Dann läßt sie das Programm wieder ablaufen.

„Was für eine sonderbare Wirkung", schmunzelt Mister M.

EPISODE 7

Start des Raumschiffs



Mr. M betätigt die Taste LOAD, dazu die Anführungszeichen "", und dann ENTER und schaltet den Recorder ein. Dann drückt er auf RUN und ENTER, um das Programm zu starten.

Wenn du dein Band zurückgespult hast, dann mußt du statt dessen schreiben:

LOAD"SAUERSTOFF"

und danach ENTER drücken. Wenn der Computer OK sagt, tippst du

RUN

und drückst ENTER.

Nach einem Weilchen erscheint die Meldung

Zum Aufdrehen der
Sauerstoffversorgung
brauche ich erst
das Codewort

„SN7, kannst du uns helfen?" fragt Aleate.

„Na klar! Na klar! Ich muß mich erst einmal an das Modul anschließen!" SN7 schließt sich wieder an und quietscht recht laut: „Es heißt SAUERSTOFF. Es heißt SAUERSTOFF!" Aleate tippt

sauerstoff

ein und drückt ENTER.

Das war richtig
der Sauerstoff kommt schon

(Achte darauf, daß du das Wort Sauerstoff genau so eingibst, wie es hier steht, also mit lauter kleinen Buchstaben. Der Spectrum würde deine Antwort sonst nicht als richtig ansehen!)

Aleate will sich das LISTing des Programms ansehen und tippt LIST. Dann drückt sie ENTER.

```
10  PRINT"Zum Aufdrehen der"
20  PRINT"Sauerstoffversorgung"
30  PRINT"brauche ich erst"
40  PRINT"das Codewort"
50  INPUT C$
60  LET D$=""
70  FOR Z=1 TO LEN C$
80  LET D$=D$+CHR$(CODE C$(Z)+1)
90  NEXT Z
100 IF D$="tbvfstupgg" THEN GO TO 130
110 PRINT"Falsch"
120 GO TO 40
130 PRINT"Das war richtig,"
140 PRINT"der Sauerstoff kommt schon"
```

„So. Das ist also das Programm." Mister M sieht es sich sorgfältig an. „Hmm, es sieht im Prinzip anders aus als das Programm für die Getränkemaschine. Wir haben INPUT C$ bei Zeile 50, machen aber was ganz Komisches auf Zeile 100, und dann vergleichen wir D$ mit tbvfstupgg, was die gleiche Anzahl an Buchstaben wie das Wort SAUERSTOFF hat. Vielleicht ist das nur ein Zufall, oder – warte mal – kann sein ... kann sein, es ist das Wort Sauerstoff in Code. Die Zeilen davor müssen etwas mit dem Codieren zu tun haben." Er denkt ein paar Augenblicke nach.

„Dieses Programm macht, denke ich, folgendes: Es nimmt das Wort, das wir an C$ gegeben haben, und setzt es in einen Code um, wobei es die codierte Version an D$ gibt. Dieses Codewort wird dann mit dem Wort SAUERSTOFF verglichen – im Code lautet das TBVFSTUPGG – und wenn sie übereinstimmen, geht's los. DAS WAR RICHTIG, DER SAUERSTOFF KOMMT SCHON sagt dann der Computer." Mister M ist fertig mit seiner langen Rede.

„Die Zeilen 60 bis 90 sind der Codier-Teil des Programms, denke ich", meint Aleate:

```
60 LET D$=""
70 FOR Z=1 TO LEN C$
80 LET D$=D$+CHR$(CODE C$(Z)+1)
90 NEXT Z
```

„Was bedeutet wohl Zeile 60?

```
60 LET D$=""
```

Wir haben ja schon Sachen gesehen wie

```
LET D$="HANS"
```

aber

```
LET D$=""
```

noch nicht.

Ich kann mir vorstellen, daß damit überhaupt keine Meldung nach D$ gegeben wird. Der Speicherplatz wird praktisch ausgefegt oder entleert. Damit wird dann D$ gleich Null."

Aleate fährt fort: „Und was bedeutet jetzt aber LEN C$ da in Zeile 70? Probieren wir's mal mit Schreiben von

```
PRINT C$
```

Ah, das ergibt auf dem Bildschirm

```
sauerstoff
```

und nun

```
PRINT LEN C$
```


Das ergibt

```
10
```

Das Wort SAUERSTOFF hat 10 Buchstaben. Probieren wir nun

```
PRINT LEN "HALLO"
```

ergibt das

```
5
```

Also, LEN gibt uns die Gesamtlänge der danach folgenden Meldung. Die zu prüfende Meldung kann an einem Speicherplatz gespeichert oder einfach in Anführungszeichen geschrieben werden, wie wir das bei HALLO gesehen haben.

Und jetzt zur ganzen Zeile 70:

```
70 FOR Z=1 TO LEN C$
```

ist der Anfang einer Schleife. Die Schleife läuft für jeden Buchstaben des Worts in C$ einmal durch.

Wenn das Wort z. B. MANN lautet, dann würde LEN C$ die Zahl 4 sein. Die Schleife würde umlaufen und Z zuerst auf 1, dann 2, dann 3 und schließlich 4 setzen."

„Gut", unterbricht Mister Micro, „was macht aber Zeile 80?" Sie sehen sich Zeile 80 an:

```
80 LET D$=D$+CHR$(CODE C$(Z)+1)
```

und müssen schlucken. War das eine lange Zeile!

„Du meine Güte!" ruft Aleate.

„Nun, fangen wir mal am Anfang an", Mister M kraust die Stirn und beginnt zu überlegen: „Erst einmal:

```
CHR$(CODE C$(Z)+1)
```

Arbeiten wir das in Schritten aus. Zuerst ersetzen wir mal C$ durch HALLO, und dann wollen wir Z gegen 3 auswechseln. Das ergibt

```
CHR$(CODE "HALLO" (3)+1)
```

Das wollen wir jetzt zerlegen!

```
LET D$=D$+CHR$(CODE "HALLO" (3)+1)
```

Betrachten wir zuerst CODE "HALLO" (3)+1

Das heißt doch, wir müssen den Code des dritten Buchstaben nehmen."

„Ach so", unterbricht Aleate, „der dritte Buchstabe von HALLO ist ein L. Wir können den Ausdruck

```
LET D$=D$+CHR$ (CODE "HALLO"(3)+1)
```

also auch kürzer schreiben:

```
LET D$=D$+CHR$ (CODE "L"+1)
```

Untersuchen wir nun CODE", sagt Mister Micro, „laßt uns folgendes versuchen:

```
PRINT CODE "B"
```

Er tippt das ein, und auf dem Bildschirm erscheint eine 66. „Probieren wir nun:

```
PRINT CODE "L"
```


Er gibt das ein, und der Computer schreibt eine 76.

```
76
```

„Wieso?"

„Klar!" ruft Aleate, „CODE heißt soviel wie zugehöriger Codewert! Jeder Buchstabe hat eine besondere Codezahl, mit der er dargestellt wird. Mit CODE wird uns lediglich der Codewert des danach folgenden Buchstabens angegeben. Wenn also Z die Zahl 3 war und C$ das Wort HALLO, dann hatten wir den Codewert für L, und der ist 76.

Wir wollen das einmal ansehen", schlägt Aleate vor. „Nehmen wir die umgeformte Zeile 80 und ersetzen wir CODE "L" durch 76."

```
LET D$=D$+CHR$(CODE"L"+1)
LET D$=D$+CHR$(     76     +1)
```

„Gut", meint Mister M, „das bringt uns jetzt zu CHR$, dem nächsten Rätsel. Wir wollen noch ein Experiment machen."

Er gibt

```
PRINT CHR$(76)
```

ein, und auf dem Bildschirm erscheint der Buchstabe L.

„Aha", sagt Aleate, „CHR$ scheint genau das Umgekehrte von CODE zu bewirken. In CHR$ setzen wir eine Zahl ein und erhalten einen Buchstaben bzw. ein Zeichen. Mit CODE haben wir ein Zeichen eingegeben und eine Zahl bekommen. Der Befehl CHR$ ist dann wohl eine Kurzform für Zeichen. In beiden Fällen ist die Zahl der Code für den Buchstaben. Schauen wir uns unsere Zeile noch einmal an:

```
LET D$=D$+CHR$(CODE "HALLO"(3)+1)
```

Sie setzen sich und blicken gedankenvoll auf den Bildschirm. Indessen rollt SN7 leise herum, und nur ein sanftes Rattern ist zu hören.

Ganz plötzlich steht Mister Micro auf, so daß Aleate einen Schreck kriegt und SN7 zum Stillstand kommt.

„Ich glaube...", murmelt er, tief in Gedanken, „ich glaube, ich kann da etwas sehen."

Aleate schaut hoffnungsvoll zu Mister M auf, als er weiterspricht: „Das Programm verwendet nicht CHR$(irgendetwas), sondern es nimmt CHR$(irgendetwas+1). Das ist's! Das ist's! Sieh nur, wenn CHR$(76)="L" ist, dann müssen wir CHR$(76+1), also CHR$(77) suchen: Und das ist M, ein Buchstabe weiter im Alphabet."

„Jetzt versteh' ich's auch!" ruft Aleate froh. „Das Programm fügt jedesmal einen Wert zum Code CHR$() hinzu, was bedeutet, daß es einfach nur den nächsten Buchstaben nimmt." Sie geht zur Tastatur hinüber und gibt ein:

```
PRINT CHR$(76)
```

Auf dem Bildschirm erscheint ein L. Dann tippt sie

```
PRINT CHR$(76+1)
```

Sie drückt ENTER, und der Buchstabe M erscheint. „Aha!" rufen sie im Chor, und SN7 rollt heran. „Tippe es ein! Tippe es ein! Tippe es ein!" quietscht er.

„Ja, Aleate, wir brauchen also nur jeden Buchstaben von TBVFSTUPGG um jeweils einen Buchstaben im Alphabet zurückzusetzen, und schon erhalten wir das Wort SAUERSTOFF." Mister Micro nimmt seinen Füller und schreibt auf den Notizblock:

```
TBVFSTUPGG
```


Dann rechnet er, vor sich hin murmelnd: „Der Buchstabe vor T ist S, der vor B ist A, vor V ist U, vor F ist E, vor S..."

„Klar, das ist's!" jubelt Aleate, „so ist das Wort SAUERSTOFF aufgebaut! Ganz klar! Sehen Sie nur!" Sie kritzelt auf den Notizblock:

```
TBVFSTUPGG
SAUERSTOFF
```

Trotz dieses Erfolgs zeigt Aleates Gesicht aber auch noch ein bißchen Verwunderung.

„Da fehlt aber immer noch etwas", murmelt sie. „Die letzte Zeile, die wir uns angesehen haben, Zeile 80. Die haben wir immer noch nicht völlig gelöst. Was bedeutet denn der erste Teil,

```
LET D$=D$+ usw.
```

Da bin ich mir gar nicht sicher."

Mister M übernimmt das Wort: „Ich denke, das ist wahrscheinlich ähnlich wie das, was wir mit Zahlen schon einmal hatten, das heißt also

```
LET X = X + 1
```

Machen wir mal einen Versuch:

```
LET D$ = ""
```

Das ist zum Entleeren von D$, damit der Platz keine Buchstaben mehr enthält.

Also:

```
LET D$ = D$ + "S"
PRINT D$
```

Wir erhalten:

```
S
```

Und wenn ich jetzt schreibe:

```
LET D$ = D$ + "N"
PRINT D$
```

dann bekomme ich:

```
SN
```

Dann noch:

```
LET D$ = D$ + "7"
PRINT D$
```

Das ergibt

```
SN7
```

Wie du siehst, fügen wir jedesmal einen weiteren Buchstaben hinzu. Nun zu

```
LET D$ = D$ + CHR$ (CODE C$(Z)+1)
```

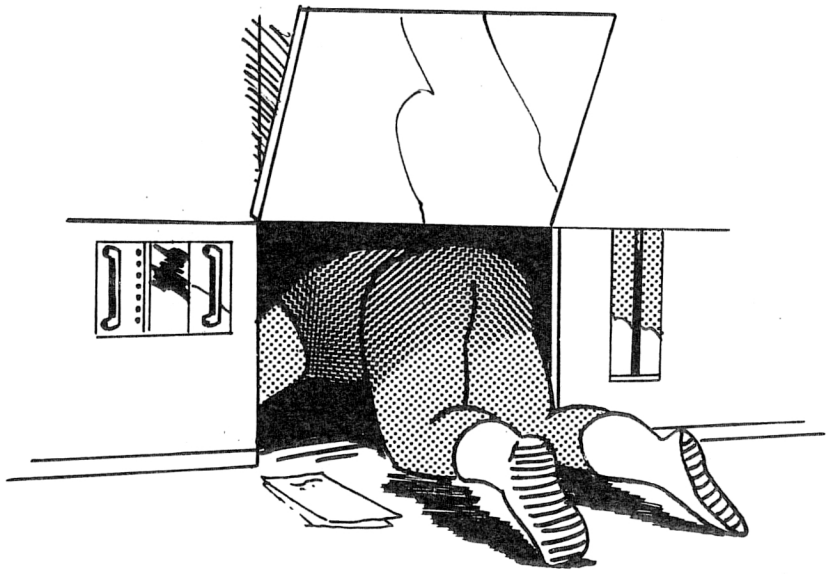
Wir fügen zu D$, das zunächst leer ist, die Buchstaben T,B, V, F,S, T, U, P, G und G jedesmal hinzu, wenn wir die FOR-NEXT-Schleife durchlaufen. So haben wir dann schließlich das Wort TBVFSTUPGG am Speicherplatz D$, sofern SAUERSTOFF das Wort an C$ war, das wir eingegeben haben. Nun", fährt er fort, „in Zeile 100 können wir sehen, ob D$="TBVFSTUPGG" ist. Wenn das stimmt, gehen wir zur Zeile 200. Der Rest des Programms ist dann das gleiche wie beim Programm für die Getränkemaschine."

Mit dem frischen Sauerstoff und einem guten Getränkevorrat scheint nun alles in bester Ordnung zu sein. Doch

haben sie immer noch nicht herausgefunden, wie man mit dem Raumschiff fliegen kann. Mister M sorgt sich. „Wie können wir denn nur dieses Schiff dazu bringen, daß es fliegt?" fragt er.

Aleate hat eine gute Idee. „Kann denn ein Mann Ihres Kalibers, der doch Beam-Maschinen bauen kann, nicht die elektrischen Drähte in diesem Schiff neu verlegen?" fragt Aleate. „Ich kann mit diesem Computer jetzt navigieren, und die Flugbahn-Berechnungen sind auch kein Problem: Die sind sicher genau wie bei meinem Schiff."

„Ich besorge das Programmieren, wenn Sie sich um das Anlassen der Maschine kümmern", sagt Aleate. Aber Mister M hat sich schon an die Arbeit begeben. Man sieht seine untere Hälfte aus der Service-Luke ragen. Dieser Anblick wird von gedämpften Worten begleitet:



„Dieser grüne Draht geht an die Zentraleinheit, und der blaue Schalter steuert den Explosionsmodulator."

„Dieser grüne Draht geht an die Zentraleinheit, und der blaue Schalter steuert den Explosionsmodulator."

Mister M arbeitet stundenlang. Aleate ist inzwischen mit dem Programmieren fertig. Sie hat sich hingesetzt und ist eingeschlafen.

Bis...

„Du lieber Himmel! Es ist ein Dynodetonations-Antiquantum-Annihilationsantrieb mit Umkehrpolarität!" ruft Mister Micro. „Äußerst eindrucksvoll!" Er kriecht aus der Inspektionsluke und steht auf. Er streckt die Hände aus und betätigt einige Drucktasten an der Schalttafel.

„Fertigmachen zum Beschleunigen", ertönt eine künstliche Stimme. Mister M setzt sich auf einen der Sessel. Aleate wacht auf. Plötzlich ist ein Rumoren in der Ferne zu hören.

„10...9...8..." ertönt die Synthesestimme, „7...6...5...4 ...3", das Rumoren wird zu einem Donnern –

„2"

Aus dem Donnern wird rasendes Kreischen –

„1"

Das Kreischen wird zum Getöse wie von tausend Wirbelstürmen –

„0"

Mister M und Aleate werden hart in die Sessel gedrückt, und schon ist das Raumschiff weit weit von dem Platz entfernt, an dem es sich gerade noch befunden hat.

Zagonenthon blickt auf seinen Radarschirm und flucht, als seine Raketen vom Standardtyp Fünf an dem Ort explodieren, an dem Aleate und Mister M einst gewesen sind. Gerade noch einmal davongekommen.

Mister M erklärt

...wobei es sich um eine logische, kurzgefaßte und rationale Erklärung der Kunst des Programmierens des Computers handelt, der im allgemeinen Sprachgebrauch als ZX Spectrum bezeichnet wird.

Für Rolf, eine Zusammenfassung der Weisheit und Intelligenz, gesammelt von Deinem lieben Onkel anläßlich meiner unfreiwilligen Reise in eine fremde Welt.

Auf diesen mannigfaltigen Seiten wird Du meine bescheidenen Sammlungen über die folgenden geheimen Themen finden:

BENUTZUNG DER TASTATUR
CHR$()
CODE
DOPPELPUNKT (:)
EDITIEREN (BEARBEITEN)
FOR...NEXT
GO TO
INPUT
INT
LEN
LET und SPEICHERPLÄTZE
LIST
LOAD
MATHEMATISCHE SYMBOLE
MEHRANWEISUNGS-ZEILEN
NEW
PRINT
REM
RND
SAVE (SICHERN)
SEMIKOLON
(STRICHPUNKT (;)
STEP
STOP
STRING
?SYNTAX ERROR
(SYNTAX-FEHLER)
TO
VARIABLE (siehe LET)
VERIFY (PRÜFEN)
und auch
LÖSUNGEN für
einige Übungen!

BENUTZUNG DER TASTATUR

Wenn du dir die Tastatur ansiehst, wirst du bemerken, daß jede Taste mit mehreren Wörtern und Symbolen bedruckt ist. Alle Befehle, die du in einem Programm benötigst, sind auf der Tastatur vorhanden und können – je nach dem, in welchem Modus sich der Computer befindet – direkt eingegeben werden. Welcher Modus gerade eingeschaltet ist, verrät dir der Cursor (Schreibmarke), der in Form eines blinkenden Buchstabens in der linken, unteren Ecke steht. Es gibt fünf verschiedene Betriebsarten, die folgendes bedeuten:

K Damit ist der Befehlswort (keyword) – Modus gemeint. Immer, wenn das blinkende K auf dem Bildschirm steht, werden die in Weiß auf der Taste stehenden Befehle oder aber die Zahlen 1 bis 0 nach Tastendruck aufgerufen. Wenn auf dem Bildschirm **kein** Cursor zu sehen ist, der Computer aber auch kein Programm abarbeitet, ist der K-Modus eingeschaltet.

L Das ist der normale Schreibmaschinen- oder Buchstaben-Modus. Dabei wird immer der Buchstabe (oder die Zahl) aufgerufen, der (die) ebenfalls in weißer Schrift auf der Taste steht. Normalerweise werden die Buchstaben als Klein-Buchstaben wiedergegeben. Erst, wenn du CAPS SHIFT gedrückt hältst und dabei einen Buchstaben tippst, erscheint dieser als Groß-Buchstabe.

C Großschreib-Modus. Dieser ist mit dem L-Modus identisch, wobei aber alle Buchstaben direkt als Groß-Buchstaben erscheinen. Um den Cursor auf C umzuschalten, kannst du CAPS SHIFT und die Taste 2 drücken (gleichzeitig).

Wenn sich der Computer im K-, L- oder C-Modus befindet, erreichst du die in Rot geschriebenen Befehle durch das Drücken der Taste SYMBOL SHIFT und das gleichzeitige Betätigen der gewünschten Befehlstaste.

E Damit ist der erweiterte Modus (extended mode) gemeint. Um diesen Modus einzuschalten, werden die Tasten CAPS SHIFT und SYMBOL SHIFT gleichzeitig gedrückt. Der Cursor wird dann zu einem blinkenden E. In diesem Modus können die grünen Befehle über den Tasten angewendet werden. Wenn du im E-Modus noch die Taste SYMBOL SHIFT gedrückt hältst, kannst du auch noch die Befehle **unter** den Tasten (rot) einsetzen.

G Grafik-Modus. Durch gleichzeitiges Betätigen der Tasten CAPS SHIFT und 9 wird der Cursor zu einem blinkenden G. Dadurch werden die auf den Tasten 1 bis 8 abgebildeten grafischen Symbole aufgerufen.

CHR$

Dein Computer ist mit einem Sondercode versehen, den er in seiner elektronischen Ausstattung verwendet. Jedes Zeichen, das heißt jeder Buchstabe, jede Zahl und jedes Symbol (z.B. *) auf der Tastatur hat einen eigenen Codewert (oder Codezahl), den der Computer immer dann benutzt, wenn er mit einem dieser Zeichen arbeitet. Der Befehl CHR$ bringt den Computer dazu, dir das Zeichen für den jeweiligen Codewert mitzuteilen.

Zum Beispiel führt

```
PRINT CHR$66
```

zur Anzeige des Buchstabens B auf dem Bildschirm. In dem Sondercode deines Computers ist die Zahl 66 das Codewort für den Buchstaben B. Alle Symbole auf der Tastatur des Computers haben ihre eigene Codezahl. Beobachte einmal, was folgender Befehl bewirkt!

```
PRINT CHR$13
```

Also, 13 ist die Codezahl für die Taste ENTER, und da dein Computer nicht ENTER schreiben kann, ohne daß ihm dies

mit PRINT „ENTER" gesagt wird, hat er den Befehl ENTER *ausgeführt*, wodurch der Computer eine neue Zeile beginnt: Die Meldung OK erscheint jetzt auf dem Bildschirm.

Du kannst z. B. mit der Anweisung

```
LET X$ = CHR$67
```

den Buchstaben C in den Speicherplatz X$ bringen.

Das Gegenteil von CHR$ ist CODE, wodurch ein Zeichen in seine Codezahl umgewandelt wird (siehe CODE). CHR$ und CODE werden im Codierprogramm in Kapitel 7 angewendet.

CODE

Dein Computer ist mit einem Sondercode versehen, den er in seiner elektronischen Ausstattung verwendet. Jedes Zeichen, das heißt jeder Buchstabe, jede Zahl und jedes Symbol (z. B. *) auf der Tastatur hat einen eigenen Codewert (oder Codezahl), den der Computer immer dann benutzt, wenn er mit einem dieser Zeichen arbeitet. Der Befehl CODE bringt den Computer dazu, dir den Codewert für das jeweilige Zeichen mitzuteilen.

Zum Beispiel, wenn du

```
PRINT CODE "A"
```

tippst und die Taste ENTER drückst, erhältst du die Zahl 65. Sie ist der Codewert des Computers für den Buchstaben A.

Das von dir verwendete Zeichen muß immer von Anführungszeichen oder Gänsefüßchen (") umschlossen sein. Andernfalls kann es der Computer nicht verstehen.

Probiere einmal,

```
PRINT CODE "1"
```


einzugeben. Nachdem du dann ENTER gedrückt hast, sollte dir der Codewert deines Computers für die Zahl 1 angezeigt werden, der 49 ist. Du kannst aber auch Befehle schreiben, wie z. B.

```
10 LET B = CODE "7"
20 PRINT B
```

Wenn du dieses Programm ablaufen läßt (mit dem Kommando RUN), sollte der Codewert 55 auf dem Bildschirm erscheinen, weil 55 die Codezahl der Ziffer 7 ist.

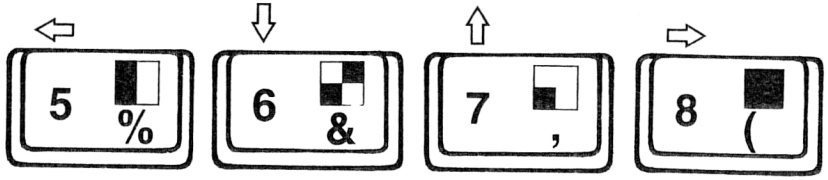
Der Befehl CODE wird vom Codierprogramm in Kapitel 7 verwendet. Das Gegenteil von CODE ist CHR$, mit dem ein Codewert in das tatsächliche Zeichen umgewandelt wird (siehe CHR$).

DOPPELPUNKT

Der Doppelpunkt ist ein Zeichen (:), mit dem einzelne Befehle in einer Programmzeile voneinander getrennt werden (siehe Mehranweisungs-Zeilen).

EDITIEREN (BEARBEITEN)

Was tust du, wenn du einen Fehler gemacht hast und ihn berichtigen willst? Du kannst die ganze Zeile neu eintippen, aber das ist viel zu langwierig. Zum Glück kann dir dein Computer hier helfen. Er hat sogenannte „Cursor-Steuertasten" – die vier Tasten 5 bis 8.



Der Cursor (auch Schreibmarke genannt) ist das blinkende Viereck auf dem Bildschirm, das dir anzeigt, wo du

als nächstes schreiben wirst. Mit den Cursor-Steuertasten kannst du eben diesen Cursor steuern!

So – und wie kannst du nun einen Fehler berichtigen? Nun, das hängt zum Teil davon ab, was falsch ist. Nehmen wir einmal an, du hast INPUT C$ getippt, wolltest aber PRINT C$ schreiben. (Schreibe dies *genau* wie hier angegeben, und drücke ENTER, wenn du mit der Zeile fertig bist). Am besten, du benutzt die Zeilennummer 1, damit kein Programm gestört wird, das du möglicherweise in deiner Maschine hast.

Jetzt steht die Zeile am oberen Bildschirmrand. Wenn du genau hinsiehst, wirst du rechts neben der Zeilen-Nummer einen kleinen Pfeil (>) sehen. Damit wird die Stelle (Zeile) markiert, die du bearbeiten kannst. Um diese Markierung zu bewegen, kannst du CAPS SHIFT und die Tasten 6 und 7 drücken (je nach Richtung). Die Markierung springt dann eine Zeile höher oder tiefer.

```
1 INPUT C$
```

Um nun das Wort INPUT in PRINT zu ändern, machen wir folgendes:

1. Drücke CAPS SHIFT nieder und dann die Taste 1 (EDIT). Die Zeile steht jetzt auch am unteren Bildschirmrand. Neben der Zeilen-Nummer steht der blinkende K-Cursor.
2. Halte CAPS SHIFT weiter gedrückt und betätige dann die Taste 8, um den Cursor in die auf der Taste angegebene Richtung zu bewegen. Der Cursor soll vor C$ stehenbleiben.
3. Betätige jetzt wieder zusammen mit CAPS SHIFT die Taste 0, über der auch das Wort DELETE steht. Was immer links neben dem Cursor steht (in diesem Fall INPUT), wird gelöscht, der Rest der Zeile schrumpft dadurch zusammen.

4. Jetzt kannst du den Befehl PRINT eingeben (durch Drükken der Taste P).
5. Drücke ENTER
6. Die bearbeitete Zeile 1 verschwindet wieder vom unteren Bildschirmrand und wird in der korrigierten Form in das Programm eingefügt.

Schon fertig!

Tippe jetzt eine neue Zeile 1 ein, wieder *genau* wie hier beschrieben, und drücke ENTER am Ende der Zeile.

```
1 IF X>5 THEN GO TO 40
```

wobei das Symbol > soviel wie „größer als" heißt – falls du es noch nicht gesehen hast (im Abschnitt „Mathematische Symbole" findest du mehr darüber). Nehmen wir nun an, daß diese Zeile falsch in deinem Programm ist, und du in Wirklichkeit folgendes haben wolltest:

```
1 IF X<5 THEN GO TO 4
```

Das Symbol < bedeutet „kleiner als". Also, wie ändern wir nun unsere alte Zeile 1 in die neue Zeile 1?

1. Versichere dich, daß die Zeilenmarkierung noch auf Zeile 1 steht. Ist das nicht der Fall, benutze zusammen mit CAPS SHIFT die Tasten 6 und 7, um dorthin zu gelangen.
2. Drücke jetzt CAPS SHIFT und die Taste 1 (EDIT), damit die Zeile wieder an den unteren Rand des Bildschirms gelangt.
3. Halte CAPS SHIFT gedrückt und betätige dann 3mal die Taste 8, damit der Cursor direkt rechts neben dem Zeichen > steht.

4. Drücke gleichzeitig CAPS SHIFT und 0 (DELETE), und das Zeichen > verschwindet.
5. Drücke jetzt die Taste mit dem neuen Zeichen (SYMBOL SHIFT und die Taste R gleichzeitig).
6. Bewege danach den Cursor durch CAPS SHIFT und die Taste 8 rechts neben die Zahl 0, die wir ja auch noch löschen wollen.
7. Drücke gleichzeitig wieder CAPS SHIFT und DELETE, damit die 0 verschwindet.
8. Drücke jetzt ENTER, damit die Zeile wieder in das Programm aufgenommen wird.

Wir wollen jetzt noch einmal zusammenfassen, wie eine Programmzeile bearbeitet wird:

- Bewege die Zeilenmarkierung (durch CAPS SHIFT und die Tasten 6 oder 7) so, daß sie auf die zu bearbeitende Zeile zeigt.
- Verwende EDIT (CAPS SHIFT und 1), um die Zeile aus dem Programm an den unteren Bildschirmrand zu bekommen.
- Bewege den Cursor mit den entsprechenden Tasten in die gewünschte Richtung.
- Lösche die Buchstaben oder Befehle, die ausgetauscht werden sollen, durch DELETE (CAPS SHIFT und 0) und setze die neuen Zeichen einfach ein.
- Betätige dann ENTER, damit die bearbeitete Zeile wieder an die richtige Stelle im Programm kommt.

Hier sind ein paar Übungen für dich zum Ausprobieren. Die möglichen Antworten hierzu findest du im Kapitel *Lösungen.*

Übung 1

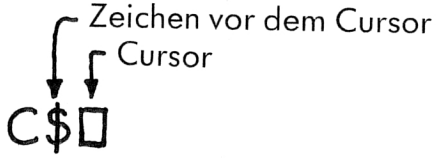
Ändere

```
10 PRINT C$
```

in

```
10 PRINT X
```

Ein Tip: Denke daran, daß du das Zeichen $ löschen mußt durch Drücken der Tasten CAPS SHIFT/DELETE. Denke aber daran, daß DELETE immer das unmittelbar *links* neben dem Cursor stehende Zeichen löscht.



Übung 2

Ändere

```
1 LET X=20
```

in

```
1 LET D5=20
```

FOR...NEXT

Mit dieser Anweisung bringst du deinen Computer dazu, einen Teil eines Programms mit einer bestimmten Häufigkeit zu wiederholen.

Nehmen wir einmal an, du willst, daß dein Computer etwas 10mal wiederholt – z. B. das Schreiben von „SN7". Eine Möglichkeit hierfür ergibt sich aus einem Programm,

wie es hier aufgeschrieben ist. Versuch's einmal, indem du es eingibst. Vergiß bitte nicht, am Ende jeder Zeile ENTER zu drücken.

```
2 FOR X = 1 TO 10
3 PRINT "SN7"
4 NEXT X
```

Schreibe jetzt RUN und drücke ENTER. Dein Computer gibt nun zehnmal SN7 auf den Bildschirm. Hierbei sagt ihm Zeile 2, wie oft er „SN7" schreiben soll (PRINT). Versuche einmal, Zeile 2 wie folgt zu ändern:

```
2 FOR X = 1 TO 15
```

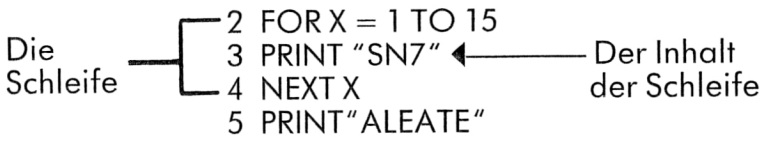
Wie oft schreibt das Programm es nun wohl? Richtig, es würde die PRINT-Anweisung 15mal ausführen und dir 15 „SN7" geben – welch' schrecklicher Gedanke! Also, mit der Zahl hinter dem TO bestimmt man, wie oft etwas ausgeschrieben werden soll. Von einer Programmzeile wie dieser sagt man, sie bildet eine Schleife zwischen den Zeilen 2 und 4.

Versuche nun, das Programm durch Hinzufügen von

```
5 PRINT"ALEATE"
```

zu verändern.

Wenn du das Programm jetzt ablaufen läßt, wird es dennoch nur einmal „ALEATE" schreiben. Das liegt daran, daß der Inhalt der Schleife zwischen den Zeilen FOR und NEXT liegt.



Übung 3

Als Übung für dich selbst versuche einmal, die Schleife so

zu gestalten, daß sie nach jedem „SN7" den Namen „ALEATE" schreibt. Die Lösung findest du im Abschnitt *Lösungen.*

Außer dem Zählen bis zu einer bestimmten Zahl kann die Schleife auch so gesetzt werden, daß sie eine veränderliche Zahl zählt, wenn du die feste Zahl durch eine Veränderliche (oder auch Variable) ersetzt, z.B. L. Schreibe NEW und drücke dann ENTER. Versuche nun folgendes:

```
1 LET L = 20
2 FOR X = 1 TO L
3 PRINT "SN7"
4 NEXT X
```

Wenn dieses Programm ausgeführt wird, durchläuft es die Schleife 20mal.

Übung 4

Ändere das obige Programm so, daß es „SN7" 12mal schreibt. Eine Lösungsmöglichkeit findest du im Abschnitt *Lösungen.*

Übung 5

Wenn du INPUT bereits kennst, kannst du das obige Programm so abändern, daß es fragt, wie oft es „SN7" schreiben soll, und dann ausführt, was ihm gesagt wird. Die Lösung findest du ebenfalls hinten.

Außer zum Schreiben von Meldungen kann eine Schleife auch zum Zählen verwendet werden. Versuche doch einmal:

```
1 LET L = 20
2 FOR X = 1 TO L
3 PRINT X
4 NEXT X
```

In diesem Falle schreibt das Programm den Wert von X (der als *Schleifenvariable* bezeichnet wird), während es abläuft. Dies kann zum Beispiel nützlich sein, wenn du – sagen wir mal – die 1x8-Tabelle schreiben lassen willst. Hierfür braucht man nur das X mit 8 malzunehmen (wenn du das Multiplizieren noch nicht gemacht hast, kannst du dies im Abschnitt mit der Überschrift „Mathematische Symbole" nachsehen).

```
1 LET L = 10
2 FOR X = 1 TO L
3 PRINT X*8
4 NEXT X
```

Übung 6

Ändere das obige Programm so, daß es die Multiplikationstabelle von 9 ausschreibt. Eine Lösung findest du im Abschnitt *Lösungen*.

Ein anderes Kommando, das man in Verbindung mit der Schleife FOR...NEXT verwenden kann, ist STEP. Weitere Einzelheiten hierzu findest du im Abschnitt mit der Überschrift STEP.

GO TO

Mit dieser Anweisung wird dem Computer gesagt, daß er auf eine bestimmte Programmzeile springen und das Programm von dort weiter ausführen soll.

So können wir zum Beispiel sagen:

```
100 LET X = 25
110 GOTO 130
120 STOP
130 PRINT X
140 GOTO 120
```


Wenn dein Computer dieses Programm ausführt, macht er es in dieser Reihenfolge: 100, 110, 130, 140, 120 und nicht 100, 110, 120, 130, 140. Der Grund hierfür ist, daß deinem Computer in dem obigen Programm in Zeile 110 gesagt wird, daß er nach Zeile 130 gehen soll (go to), was er dann sofort ausführt, indem er Zeile 120 überspringt. Wenn er dann zu Zeile 140 kommt, wird ihm gesagt, daß er auf Zeile 120 springen soll, was er auch ausführt. Bei dieser Zeile stoppt er dann das Programm (STOP) und gibt die Meldung 9 STOP statement, 120 : 1 (Unterbrechung bei Zeile 120) aus, was lediglich bedeutet, daß ihm gesagt worden ist, bei Zeile 120 anzuhalten.

Der Befehl GO TO ist nützlich, wenn du ihn mit den Anweisungen IF...THEN verwendest, um dem Computer zu sagen, was er tun soll, nachdem er eine Entscheidung über etwas getroffen hat (siehe IF...THEN). Ein gutes Beispiel für die Verwendung von GO TO findest du im Kapitel 5, in dem Aleate ein Zahlen-Ratespiel erstellt.

INPUT

Mit dieser Anweisung bringst du deinen Computer dazu, ein Programm zu stoppen und Informationen vom Benutzer anzunehmen, wie z. B. eine Zahl oder eine Meldung. Nachdem du ihm geantwortet und dann RETURN gedrückt hast, führt er den Rest des Programms aus. Versuche es einmal mit:

```
1 INPUT A$
2 PRINT A$
```

Wenn du jetzt RUN schreibst und ENTER drückst, geschieht folgendes:

1. Der Computer sieht die Anweisung INPUT und schreibt einen blinkenden [L]-Cursor auf den Bildschirm – er wartet auf eine Mitteilung.

2. Gib etwas ein, sagen wir mal: ICH BEGRUESSE DICH, ERDLING.
3. Drücke ENTER.
4. Der Computer speichert die Meldung am Speicherplatz A$ (weitere Informationen hierüber findest du unter LET).
5. Der Computer erhält die Anweisung PRINT A$,
   - schaut am Speicherplatz A$ nach und zieht die Meldung ICH BEGRUESSE DICH, ERDLING hervor, die er dann
   - als Meldung auf den Bildschirm schreibt.
6. Der Computer sucht nach der nächsten Programmzeile, und wenn er nichts findet, hört er auf.

Wenn Speicherplätze wie A$ mit dem Symbol $ (Dollar) aufhören, dienen sie zum Speichern von Zeichenketten (Meldungen). Wenn du eine Zahl speichern willst, würdest du einen Speicherplatz wie A oder AA oder X oder KW anstelle von A$ verwenden. Du kannst dir deine eigenen Speicherplatznamen ausdenken – weitere Erläuterungen findest du unter LET. (In Kapitel 1 geben Mister M und Aleate Informationen an die Speicherplätze R und H im Programm CREWDATEN.)

INT

Diese Anweisung ist eine Abkürzung von INTEGER (ganze Zahl). Mit ihr wird der Computer aufgefordert, eine Dezimalzahl zu nehmen und in eine ganze Zahl umzuwandeln. Bei positiven Zahlen wie z.B. 8.1 oder 9.64 werden ganz einfach die Zahlen hinter dem Dezimalpunkt (beim Computer meistens anstelle von Dezimalkomma) fallengelassen, so daß man in unserem Beispiel dann 8 bzw. 9 erhalten würde. Bei negativen Zahlen wie z.B. −10.2 und

−12.812 entfallen ebenfalls die Zahlen hinter dem Dezimalpunkt, jedoch wird die negative Zahl in die folgende negative Zahl umgewandelt: Bei unserem Beispiel würde daher −10.2 zu einer −11 und −12.812 zu einer −13 werden.

Hier sind einige Beispiele für die Verwendung von INT: Tippe jedes Beispiel ein, und drücke dann ENTER. Denke daran, daß INT nicht alleine stehen kann; es muß immer mit einem anderen Kommando verwendet werden, wie zum Beispiel PRINT oder LET. Wenn du nur INT 2.47 schreiben würdest, dann würde das nicht funktionieren, weil du deinem Computer nicht gesagt hast, was er mit der Antwort anfangen soll!

```
PRINT INT 891.2148
```

ergibt dann 891.

```
1 LET X=INT −16.4
2 PRINT X
```

Wenn du nun RUN tippst und ENTER drückst, wird dir eine −17 auf den Bildschirm gegeben.

```
1 LET TP=6+INT 3.1
2 PRINT TP
```

Nach Tippen von RUN und Drücken von ENTER bekommst du eine 9.

Verstehst du, warum?

LEN

LEN ist eine Abkürzung des englischen Worts für Länge (length), und es berechnet dir die Länge eines Textes. Zum Beispiel:

```
LEN "ROBOTER" ergibt 7
```

denn das Wort ROBOTER hat sieben Buchstaben.

```
A$="SN7"
LEN A$ ist also 3
```

LET und SPEICHERPLÄTZE

Schaue dir dieses kurze Programm an:

```
1 LET X=9
2 PRINT X
```

Die Zeilennummern sind nicht wichtig, doch muß die zweite größer als die erste Zeilennummer sein.

Mister M und Aleate haben herausgefunden, daß mit LET etwas an einen Speicherplatz gebracht wird. Du kannst dir auch vorstellen, es sei ein kleiner Kasten, in den du Zahlen hineinlegst.

```
LET X = 9
```

Wenn du jetzt sagst:

```
LET X = 8
```

dann würde der Computer eine 8 in den Kasten mit dem Namen X legen.

Am besten wir betrachten den ganzen Vorgang einmal von Anfang an! Zuerst das Stück „LET X". Dieses sagt dem Computer:

„Suche nach einem Kasten mit dem Namen X, und wenn du ihn gefunden hast, bringe den nachstehend geschriebenen Wert in diesen Kasten."

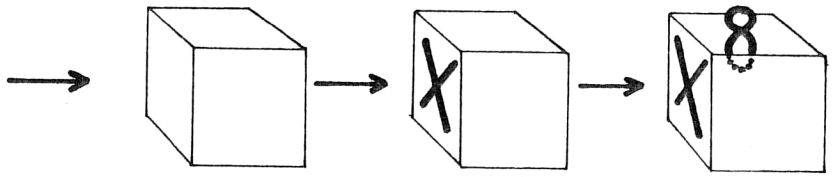
Was geschieht aber, wenn kein Kasten mit dem Namen X vorhanden ist? Nun, dann bedeutet die Anweisung:

„Finde einen Kasten ohne Namen, laß (LET) ihn X heißen, und gib dann die Zahl in diesen Kasten."

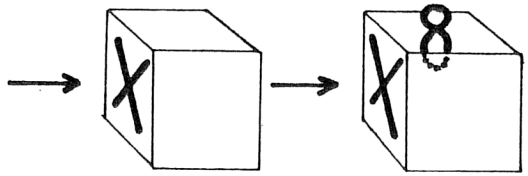
Die verwendete Zahl ist natürlich die Zahl, die hinter dem Gleichheitszeichen steht.

Wie sieht das nun in der Praxis aus?

1. Wenn X noch nicht vorhanden ist.



2. Wenn X bereits vorhanden ist.



Ist die Zahl erst einmal in einem Kasten gespeichert, dann ist es kein Problem, sie auf den Bildschirm zu bekommen: Du brauchst nur etwas einzutippen wie:

```
PRINT X
```

Das bedeutet:

„Suche nach einem Kasten mit dem Namen X, sieh nach, was in dem Kasten drin ist, und zeige diesen Wert auf dem Bildschirm."

Bis jetzt haben wir alle Kästen nur X genannt. Das ist aber ein bißchen langweilig. Leider sind die meisten Namen für

Speicherplätze in BASIC nun mal so. Sehen wir uns mal ein paar an:

X – langweilig
A – vielleicht nicht ganz so langweilig
B5 – fast schon gut
ET – das ist schon besser!

Was kannst du bei allen diesen Namen merken? Richtig! Sie fangen alle mit einem Buchstaben an. Was noch? Also, wenn eine Variable für die Speicherung von Zahlen benutzt wird, kann ihr Name beliebig lang sein, darf jedoch nur Buchstaben oder Ziffern enthalten. Beispiele:

```
LET SN7 = 5
LET EINELANGEVARIABLE = 999
```

Es ist egal, ob du für Variablen-Namen große oder kleine Buchstaben verwendest. Der Spectrum macht keinen Unterschied. Es ist z.B. variable für ihn das gleiche wie VARIABLE, beides bezeichnet denselben Speicherplatz.

Soll eine Variable statt eines Zahlenwertes eine Zeichenkette (d.h. eine Aneinanderreihung von beliebigen Zeichen, wie z.B. Namen oder Meldungen) aufnehmen, dann gelten andere Regeln. Der Name einer solchen Variablen darf nur einen einzigen Buchstaben lang sein. Diesem Buchstaben muß das Dollarzeichen ($) folgen. Die Zeichenkette auf der rechten Seite des Gleichheitszeichens in einer LET-Anweisung muß in Gänsefüßchen eingefaßt sein.

Beispiele:

```
LET A$ = "SN7"
LET X$ = "Wie geht es Dir?"
```

Überzeuge dich selbst, und probiere einige aus – hier sind ein paar neue Ideen von Mister M, aber du kannst es natürlich selbst versuchen.

Übung 7

a) Speichere 3 in D5, lasse dann auf dem Bildschirm ausschreiben, was in D5 ist.

b) Speichere 11 in NN, lasse dann auf dem Bildschirm ausschreiben, was in NN ist.

c) Speichere 5 in AZ, lasse dann auf dem Bildschirm ausschreiben, was in AZ ist.

d) Speichere 21 in Z1, lasse dann auf dem Bildschirm ausschreiben, was in Z1 ist.

Falls du steckenbleibst, kannst du hinten die *Lösungen* zu den Fragen finden.

Einfache Zahlen sind überhaupt kein Problem für deinen Spectrum, und wenn du dich erinnerst, fand Mister M in dem Programm CREWDATEN die Zeile:

```
LET T=R+H
```

Klar, daß hier drei Kästen in Frage kommen: T, R und H. Kannst du in Worten ausdrücken, was dieses bedeutet? Denke mal nach. Na, denke mal! Hast du nachgedacht? Was es eigentlich bedeutet, ist dies:

„Nimm die Zahl im Kasten R, addiere die Zahl im Kasten H dazu, und gib die Antwort in den Kasten T."

Hattest du dir das auch so gedacht?

Die Computerprogrammierer nennen Kastennamen wie R, H und T *Variable* oder auch *Veränderliche*.

LIST

Das Kommando LIST bringt den Computer dazu, das Programm darzustellen, das gerade im Speicher abgespeichert ist. Wenn du nur einen Teil des Programms sehen

willst, kannst du dem Computer genau sagen, welche Zeilen du aufgelistet haben willst.

LIST	bedeutet LISTE des ganzen Programms
List 30	bedeutet LISTE von Zeile 30 an aufwärts bis zum Ende.

LOAD

Hiermit wird dein Computer dazu gebracht, ein Programm von einem Magnetband abzulesen und im Speicher abzuspeichern.

Zuerst mußt du darauf achten, daß dein Computer und der Kassettenrecorder richtig angeschlossen sind. Dein Kassettenrecorder wird an den EAR und MIC-Buchsen auf der Rückseite deines Computers angeschlossen.

Wenn du jetzt LOAD "" und dann ENTER eingibst, wird der Bildschirm weiß. Du kannst jetzt den Kassettenrekorder starten. Wenn die Lautstärke am Rekorder richtig eingestellt ist (ungefähr 3/4 des Reglers aufdrehen), erscheinen auf dem Schirm farbige Streifen, außerdem ertönen schrille Geräusche. Dann wird der Programm-Name angezeigt. Die Streifen bleiben so lange sichtbar, bis das Programm fertig geladen ist. Danach startet das Programm entweder automatisch oder es erscheint die Meldung Ø OK, Ø:1. Es kann dann mit RUN gestartet werden (achte darauf, den Rekorder auszuschalten). Wenn du ein bestimmtes Programm laden willst, mußt du den vollen Namen des Programms eintippen, z. B. schreibst du dann

```
LOAD "CREWDATEN"
```

Von nun an sind die Anleitungen ähnlich wie oben beschrieben, nur daß dein Computer lediglich das von dir genannte Programm lädt, auch wenn es nicht das erste auf dem Band ist. Bei unserem Beispiel lädt er dann nur CREWDATEN, selbst wenn er andere Programme zuerst findet.

Mach' dir keine Sorgen, wenn er tatsächlich erst andere Programme findet; er sucht so lange weiter, bis er das Programm findet, das du abgerufen hast.

Zum Speichern der von dir geschriebenen Programme siehe bitte unter SAVE nach.

MATHEMATISCHE SYMBOLE

\+ Das Pluszeichen (Addieren oder Zusammenzählen)
– Das Minuszeichen (Subtrahieren oder Abziehen)
* Das Multiplikations- oder Mal-Zeichen (Multiplizieren oder Malnehmen)
/ Das Divisions- oder Teilungszeichen (Dividieren oder Teilen)
> Das Zeichen „größer als"
< Das Zeichen „kleiner als"

Unter IF...THEN und in Kapitel 5 kannst du mehr über die beiden letzten Zeichen erfahren.

Probiere die folgenden Operationen einmal auf deinem Computer aus:

a) Addiere 3 zu 4
b) Subtrahiere 7 von 10
c) Multipliziere 4 mit 3
d) Dividiere 12 durch 3
e) Wenn 10 größer als 3 ist, schreibe „JA"
f) Wenn 2 kleiner als 21 ist, schreibe „NEIN"

Nachstehend findest du einige mögliche Lösungen zu diesen Problemen, aber probiere es bitte erst einmal selbst!

a) PRINT 3+4
b) PRINT 10–7
c) PRINT 4*3
d) PRINT 12/3
e) IF 10>3 THEN PRINT "JA"
f) IF 2<21 THEN PRINT "NEIN"

MEHRANWEISUNGS-ZEILEN

Du brauchst nicht jede neue Anweisung an deinen Computer auf eine neue Zeile zu schreiben. Dein Spectrum erlaubt es dir, mehr als eine Anweisung in einer Zeile zu schreiben. Dazu setzt du einen Doppelpunkt (:) zwischen die einzelnen Anweisungen. Du hast z. B. schon Programme wie dieses hier gesehen, bei denen immer nur eine Anweisung in einer Zeile steht:

```
1 INPUT A$
2 FOR X=1 TO 10
3 PRINT A$
4 NEXT X
```

Es könnte aber auch so geschrieben sein:

```
1 INPUT A$
2 FOR X=1 TO 10:PRINT A$
3 NEXT X
```

Oder sogar:

```
1 INPUT A$:FOR X = 1 TO 10:PRINT A$:NEXT X
```

Es ist aber übersichtlicher, in einem Programm immer nur eine Anweisung je Zeile zu schreiben.

NEW

Dieses Kommando sagt dem Computer, daß er alle Speicherplätze entleeren und alle Programme vergessen soll, die er bereits im Speicher hat. Dieses Kommando wird gewöhnlich gegeben, bevor du mit dem Schreiben oder Laden eines *neuen* Programms beginnst.

PRINT

Mister M und Aleate fanden heraus, daß das Wort PRINT den Computer anweist, etwas auf den Bildschirm zu schreiben. Als Aleate zum Beispiel tippte:

```
PRINT "ICH MAG EIS"
```

und dann ENTER drückte, gab der Computer „ICH MAG EIS" auf dem Bildschirm aus. Daraus schlossen sie, daß der Computer bei Verwendung von PRINT alles auf den Bildschirm schreibt, was zwischen Anführungszeichen (Gänsefüßchen "") steht, und es dann genau so wiedergibt, wie es geschrieben wurde. Probiere selbst, wobei du schreiben kannst, was du willst. Zum Beispiel:

```
PRINT"SCHREIBE, WAS DU WILLST"
```

oder

```
PRINT"SPASS SPASS SPASS SPASS SPASS!"
```

Denke daran, daß du immer die Taste ENTER drücken mußt, wenn du mit einer Zeile fertig bist, damit der Computer weiß, daß er nun ausführen soll, was du ihm aufgetragen hast.

Mister M und Aleate entdeckten auch, daß man mit dem Wort PRINT auch andere Dinge außer Texte schreiben kann. Als sie (ohne Anführungszeichen)

```
PRINT R
```

schrieben, gab der Computer eine Zahl aus. Das war die Zahl, die am Speicherplatz mit dem Namen R gespeichert war. (Wenn du mehr über Speicherplätze wissen willst, sieh bitte unter LET nach.) (In dem Programm CREWDATEN wurde Speicherplatz R zum Speichern der Anzahl an Robotern benutzt, obwohl er natürlich auch zum Speichern anderer Zahlen verwendbar ist.)

Mit einem Semikolon (Strichpunkt ;) wird dem Computer mitgeteilt, daß er das nächste, das er auf dem Bildschirm ausgeben soll, direkt neben das letzte dort Geschriebene setzen soll. Das bedeutet also, daß diese beiden Dinge in *derselben* Zeile stehen. Wenn wir z. B. schreiben:

```
100 PRINT "GRUESS"
110 PRINT "DICH"
```

Und wenn wir dann RUN schreiben und ENTER drücken, bringt der Computer folgendes auf den Bildschirm:

```
GRUESS
DICH
```

Wenn wir jetzt aber schreiben (vergiß bitte nicht das Semikolon):

```
100 PRINT "DIES IST EINE";
110 PRINT "MELDUNG"
```

und dann RUN tippen sowie ENTER drücken, schreibt dein Computer:

```
DIES IST EINEMELDUNG
```

Das ist aber immer noch nicht ganz richtig, nicht wahr? Weißt du, weshalb? Weil das Semikolon bewirkt, daß etwas *direkt rechts neben* dem letzten Wort geschrieben wird. Wenn wir einen Zwischenraum zwischen den Wörtern haben wollen, müssen wir das auch in unserer PRINT-Anweisung vorsehen:

```
100 PRINT "DIES IST EINE ";
                         ↑
                  (Beachte den Zwischenraum)

110 PRINT "MELDUNG"
```


Wenn wir nun RUN tippen und ENTER drücken, erhalten wir:

```
DIES IST EINE MELDUNG
```

Probiere nun doch einige Meldungen selbst aus. Wenn du eine Meldung versuchst, die zu lang für eine Zeile ist, schreibt dein Computer automatisch auf der nächsten Zeile weiter.

Es ist wichtig, daß du beachtest, daß das Semikolon *nur* die Stellung des als nächstes auf dem Bildschirm Geschriebenen beeinflußt. Es hat auch keine Bedeutung, wie viele andere Zeilen des Programms zwischen der Zeile mit dem Semikolon und der Zeile liegen, die als nächste etwas auf den Bildschirm schreibt.

REM

Mit dieser Anweisung kannst du Kommentare oder Bemerkungen zu deinem Programm hinzufügen. Dies ist nützlich, wenn du ein kompliziertes Programm hast und du vergessen könntest, was es eigentlich tun soll.

Zum Beispiel

```
10 REM DIESES PROGRAMM SCHREIBT
20 REM EINE ZAHL AUF DEN BILDSCHIRM
30 INPUT X
40 PRINT X
```

Dein Computer ignoriert alle mit einem REM anfangenden Zeilen völlig. Die REMs sind Gedächtnisstützen für dich – deinem Computer wird auf den anderen Zeilen gesagt, was er zu tun hat. Es ist wichtig, daß du daran denkst, daß dein Computer absolut überhaupt nichts mit REM-Anweisungen macht. Nach dem Wort REM schreibt er nicht einmal eine Meldung auf den Bildschirm. Wenn du sehen willst, woran du dich mit Hilfe von REM erinnern willst, muß du LIST schreiben (siehe LIST). Dann werden die von dir in

das Programm gegebenen Bemerkungen zusammen mit dem übrigen Programm aufgelistet.

Wenn du das englische Wort REMEMBER schreibst, das auf deutsch „erinnere dich" heißt, nimmt sich der Computer nur den Teil REM und ignoriert den Rest, weil er REM als Anweisung erkennen kann.

RND

Hiermit sagst du dem Computer, er soll sich eine Zahl ausdenken, das heißt, er soll sie erfinden. Normalerweise erfindet der Computer Dezimalzahlen zwischen 0 und 1.0 (oder ganz, ganz dicht an 1.0, aber niemals ganz da). Wenn wir zum Beispiel

```
PRINT RND
```

schreiben, erhalten wir eine *Zufallszahl* – eine Zahl, die sich der Computer gerade zufällig ausgedacht hat und die er dann auf den Bildschirm schreibt.

Einige Beispiele für die Benutzung von RND:

Mit dem folgenden Programm erhält man eine Zufallsdezimalzahl zwischen 0 und 100.0. Mit der Zahl 100 hinter dem Multiplikationszeichen sagst du deinem Computer, daß er mit seiner Zufallszahl bis 100 gehen kann.

```
PRINT RND*100
```

Mit dem nächsten Programm geben wir eine Zufallsdezimalzahl zwischen 0 und 143.0 in Speicherplatz X:

```
LET X = RND*143
```

Damit wir eine ganze Zufallszahl zwischen 1 und 30 erhalten (mehr über Ganzzahlen findest du unter INT), verwendest du:

```
PRINT INT(RND*30+1)
```


Zum Schluß kannst du eine ganze Zufallszahl zwischen 10 und 39 mit

```
PRINT INT(RND)*30)+10
```

erhalten.

Dein Computer macht folgendes mit dieser Anweisung: Er denkt sich zuerst eine ganze Zahl zwischen 0 und 29 aus, dann fügt er 10 zu dieser Zahl hinzu und gibt die Antwort zur Anzeige an den Bildschirm.

SAVE

Mit diesem Kommando befiehlst du deinem Computer, ein Programm zur späteren Anwendung zu speichern. Ein Programm, das dein Computer gerade geladen hat, brauchst du nicht zu speichern (SAVE), weil das Originalprogramm beim Laden nicht vom Computer gelöscht wird, sondern auf dem Band erhalten bleibt. Du brauchst also nur ein neues Programm zu speichern oder – wie man auch sagen kann – zu sichern (SAVE).

Um ein Programm zu speichern, mußt du natürlich erst einmal ein Programm in deinem Computer haben!

Wenn du einen Kassettenrecorder verwendest, mußt du aufpassen, daß du dein neuestes Programm nicht über eines deiner älteren schreibst! Deshalb mußt du auf einen leeren Teil des Bandes vorspulen, auf dem nichts aufgezeichnet ist. Denke dir einen Namen für dein Programm aus, sagen wir mal „NAME", und schreibe:

```
SAVE "NAME"
```

Drücke nun ENTER, und auf dem Bildschirm erscheint die Meldung:

```
Start tape, then press any key.
(Starte den Rekorder,
dann drücke irgendeine Taste).
```

Befolge diese Aufforderung, und drücke die Aufnahme- und die Wiedergabe-Taste deines Kassettenrecorders gleichzeitig. Der Bildschirm ist leer, wenn das Programm aufgezeichnet wird, und dann zeigt er wieder die gleichen farbigen Streifen wie bei LOAD.

Nach einem Weilchen zeigt er Ø OK, Ø : 1 an. Das bedeutet, daß der Computer fertig ist.

Du kannst das Programm auch weiterhin ablaufen lassen (RUN), weil es im Speicher des Computers bleibt, nachdem du es aufgezeichnet hast. Dein Computer hat das Programm nur kopiert; er hat es nicht aus seinem Speicher entfernt. Wenn du prüfen willst, ob das Programm auch richtig auf Band aufgezeichnet ist, kannst du dies mit dem Kommando VERIFY ausführen.

SEMIKOLON (oder STRICHPUNKT)

Das Semikolon (;) wird zum Trennen von Dingen verwendet, so daß das nächste, das auf den Bildschirm geschrieben wird, direkt neben das letzte auf dem Bildschirm Geschriebene kommt.

Weitere Einzelheiten hierüber findest du im Abschnitt PRINT.

STEP

Diese Anweisung wird in Verbindung mit dem Kommando FOR...NEXT verwendet, und sie bewirkt, daß die Zählvariable jeweils in *Stufen* (d.h. STEPs) und nicht in Einern geändert wird. Die Größe der Stufe oder auch des Schritts wird durch die Zahl bestimmt, die nach dem Kommando STEP folgt. In den Beispielen für FOR...NEXT wurde eine 1x8-Tabelle durch Multiplizieren (Malnehmen) der Zählvariable aufgestellt. Das gleiche läßt sich aber auch mit STEP erreichen. So kannst du zum Beispiel das 1 x 2 mit

Hilfe des folgenden Programms ausschreiben lassen:

```
2 FOR X = 2 TO 24 STEP 2
4 PRINT X
6 NEXT X
```

Durch Ändern der Zahl hinter dem Kommando STEP im obigen Programm kannst du untersuchen, welche Wirkung STEP hat. Achte mal darauf, was passiert.

STEP kann auch rückwärts zählen, wenn die Zahl hinter STEP negativ ist. Wir wollen nun das Programm zum Rückwärtszählen von 10 bis 1 aufschreiben:

```
2 FOR X = 10 TO 1 STEP -1
4 PRINT X
6 NEXT X
```

Übung 8

Ändere das obige Programm so, daß es von 20 bis Null in Schritten von 2 rückwärts zählt. Eine Lösung findest du im Abschnitt *Lösungen*.

STOP

Dieses Wort darf nur innerhalb eines Programms verwendet werden, und es sagt deinem Computer, daß er mit dem Ausführen des Programms aufhören soll, wenn er das Kommando STOP antrifft.

Ein Beispiel:

```
100 LET X = 5
110 LET Y = 100
120 PRINT X+Y
130 STOP
140 PRINT Y-X
```

Wenn dein Computer die Zeile 130 liest, hört er mit dem Ausführen des Programms auf und gibt die Meldung aus:

```
9 STOP statement, 130 : 10        (Unterbrechung in
                                   Zeile 130)
```

Dies bedeutet, daß er angehalten hat, weil ihm gesagt worden ist, daß er bei Zeile 130 STOPpen und alles ignorieren soll, was nach dieser Zeile kommt.

Das Wort STOP ist sehr nützlich, wenn du willst, daß der Computer den Ablauf eines Programms an einer anderen Zeile als der letzten unterbricht. Ein Beispiel hierfür ist in der Erklärung von IF...THEN gegeben und ein weiteres im Abschnitt über GO TO.

STRING

Dieses ist ein Begriff, mit dem die Computerprogrammierer eine Anzahl von Zeichen beschreiben, die zusammen verwendet werden. Im Text haben wir dies *Meldung* oder *Zeichenkette* genannt.

So ist in der Anweisung

```
PRINT "COMPUTER"
```

das Wort COMPUTER ein *String* (oder eine Zeichenkette). Diese Kette könnte auch als Variable – z.B. als A$ – gespeichert werden, und in diesem Falle würde man von der Zeichenkette A$ (oder String A$) sprechen.

Im allgemeinen wird der Begriff String nur dann verwendet, wenn in der Kette Buchstaben als Zeichen vorhanden sind. Doch kann solch ein String aus Buchstaben, Zahlen oder einer Mischung aus beiden bestehen. Bei einer Mischung würde man von einer *alphanumerischen* Zeichenkette sprechen.

?SYNTAX ERROR (Syntax-Fehler)

Wenn Dein Computer dieses blinkende Fragezeichen ausgibt, meint er damit, es wurde ein Fehler gemacht, weil er nicht verstehen kann, was da eingegeben wurde. Du solltest dann prüfen, ob die gerade von dir eingetippte Zeile oder die angegebene Programmzeile (falls eine Zeile angegeben wird) richtig ist.

TO

Dieser Befehl ermöglicht das Zerteilen von Strings (Zeichenketten, d.h. Aneinanderreihungen von Buchstaben oder Ziffern oder Sonderzeichen). Die Anwendung ist etwas komplizierter, so daß es dir vielleicht eine Hilfe sein wird, erst einmal das Kapitel 7 durchzulesen, in dem Aleate den Umgang mit TO erlernt hat.

Unter Verwendung von TO können Teile aus einem String herausgenommen werden. Um das zu machen, mußt du erst entscheiden, welcher Buchstabe des Strings der erste und welcher der letzte sein soll.

Hier ein Beispiel:

```
PRINT "COMPUTER" (4 TO 6)
```

Auf dem Bildschirm erscheint:

```
PUT
```

eben der vierte bis sechste Buchstabe des Wortes COMPUTER. Ebenso kannst du auch String-Variablen einsetzen:

```
LET A$="COMPUTER"
PRINT A$ (4 TO 6)
```

Auch jetzt wird

PUT

auf dem Bildschirm erscheinen.

Die Regel oder Syntax von TO ist

String (erster Buchstabe TO letzter Buchstabe)

Nachdem ein Stück aus einer Zeichenkette (String) herausgeschnitten worden ist, wird es zu einer neuen, selbständigen Kette, d.h.

LET B$="COMPUTER" (4 TO 6)

würde eine Variable mit dem Namen B$ erzeugen, die die Zeichenkette PUT enthält.

TO kann jeden Buchstaben von jeder Stelle einer Zeichenkette aus herausnehmen, z.B.

LET B$="COMPUTER" (1 TO 4)

würde eine Variable B$ mit dem Inhalt COMP erzeugen.

TO kann viele Formen haben. Fehlt z.B. die erste Zahl vor TO, nimmt der Computer an, daß du am Anfang des Strings starten willst, fehlt die letzte Zahl, wird der Ausschnitt bis zum Ende des Strings genommen.

Wenn in den Klammern nur eine Zahl ohne den Befehl TO steht, wird auch nur der an dieser Stelle stehende Buchstabe genommen. Hier wieder einige Beispiele:

"COMPUTER"(1 TO 4) ="COMP"
"COMPUTER"(TO 4) ="COMP"
"COMPUTER"(6 TO 8) ="TER"
"COMPUTER"(6 TO)="TER"
"COMPUTER"(4 TO 4)="P"

"COMPUTER"(4) ="P"
"COMPUTER"(TO) ="COMPUTER"
"COMPUTER"="COMPUTER"

Wahrscheinlich wirst du einige Zeit brauchen, um TO immer richtig einzusetzen. Um dir dabei zu helfen, versuche doch einige der folgenden Übungen:

Die Auflösung steht in dem Kapitel Lösungen.

Übung 9

a) Mache B$="UTER" aus "COMPUTER"
b) Mache B$="MAN" aus "KOMMANDO"
c) Mache B$="ET" aus "RAKETE"
d) Mache B$="PECT" aus "SPECTRUM"
e) Mache B$="SEHEN" aus "FERNSEHEN"
f) Mache B$="BRIEF" aus "BRIEFKASTEN"

Wenn du erst einmal mit TO umgehen kannst, ist es dir möglich, Spiele zu erfinden, bei denen die Spieler raten müssen, welche Buchstaben in einem Wort sind. Warum probierst du es nicht?

VARIABLE

Siehe LET.

VERIFY (Prüfen)

Um sicher zu sein, daß ein Programm richtig auf dem Band aufgezeichnet wurde, benutzen wir das Wort VERIFY (das heißt soviel wie prüfen). Das Programm im Computer muß das gleiche wie auf dem Band sein, und daher mußt du zuerst VERIFY unmittelbar nach dem Speichern eines Programms benutzen. Führe dieses aus, bevor du irgend etwas anderes tust. Achte darauf, daß du das Band an eine

Stelle kurz vor dem Anfang des Programms oder ganz an den Anfang des Bandes zurückspulst. Schreibe dann (wenn du dein Programm beispielsweise NAME genannt hast)

VERIFY "NAME"

und drücke dann ENTER. Der Bildschirm wird jetzt leer.

Starte dann den Recorder.

Nach einiger Zeit tauchen wieder die blauen und roten Streifen auf.

Nach einem kurzen Moment erscheint dann die Meldung:

Program:
NAME

Während die Aufzeichnung geprüft wird, ob sie in Ordnung ist, bleiben die Streifen auf dem Bildschirm. Nachdem dies erfolgt ist, kommt die Meldung

0 OK, 0:1

Das bedeutet, daß die Aufzeichnung in Ordnung ist. Wenn sie nicht in Ordnung ist, zeigt der Bildschirm an:

R Tape loading error, 0:1

In diesem Fall solltest du versuchen, das Programm erneut zu speichern.

Lösungen

Übung 1

1. Zuerst bringst du die Zeilenmarkierung in die Zeile 1. Dafür benutzt du CAPS SHIFT zusammen mit der Aufwärtspfeil-Cursor-Taste. Drücke dann CAPS SHIFT und EDIT.
2. Steuere den Cursor mit CAPS SHIFT und der Rechtspfeiltaste, bis er gerade rechts neben den $-Zeichen steht.

   1 PRINT C$ [L]

3. Halte CAPS SHIFT gedrückt, und drücke dann zweimal auf DELETE.
4. Tippe X.
5. Drücke ENTER.

Übung 2

Du kannst dies wie folgt ausführen:

1. Steuere den Cursor mit CAPS SHIFT und den Cursor-Steuertasten in die Zeile 1. Drücke dann CAPS SHIFT und EDIT.
2. Bewege den Cursor, bis er rechts neben dem X steht.
3. Halte CAPS SHIFT nieder, und drücke DELETE einmal.
4. Tippe D5, und drücke ENTER.

Einfach, nicht wahr? Probiere dieses Beispiel ein paarmal aus, bis du es richtig verstehst. Wenn es dir nicht klar ist, macht nichts! Du kannst immer eine neue Zeile vollständig neu tippen, wenn du dich verschrieben hast – das Editieren ist im Grunde ganz einfach, wenn du es erst einmal in den Griff gekriegt hast!

Übung 3

Eine Möglichkeit, „ALEATE" nach jedem „SN7" zu schreiben, ist wie folgt:

```
2 FOR X=1 TO 15
3 PRINT "SN7"
4 PRINT "ALEATE"
5 NEXT X
```

Nun wird „ALEATE" jedesmal geschrieben, wenn die Schleife durchlaufen wird, weil das Kommando PRINT „ALEATE" innerhalb der Schleife liegt.

Übung 4

Wahrscheinlich liegt die beste Möglichkeit, den Computer dazu zu bringen, die Schleife nur 12mal anstatt 20mal auszuführen, in der Änderung von Zeile 1:

```
1 LET L=12
```

Dann kann das übrige Programm unverändert bleiben:

```
2 FOR X=1 TO L
3 PRINT "SN7"
4 NEXT X
```


Übung 5

Auch hier brauchst du nur Zeile 1 zu ändern. Der Rest des Programms bleibt gleich.

```
1 INPUT L
```

Übung 6

Für die 1 x 9-Tabelle brauchst du nur Zeile 3 zu ändern in:

```
3 PRINT X*9
```

Übung 7

Die Aufgabe läßt sich mit diesen kurzen Programmen erfüllen: Schreibe jedes Programm einzeln, drücke ENTER nach jeder Zeile, schreibe RUN und drücke ENTER erneut, wenn du mit jedem Programm fertig bist.

a)
```
10 LET D5 = 3
20 PRINT D5
```

b)
```
10 LET NN = 11
20 PRINT NN
```

c)
```
10 LET AZ=5
20 PRINT AZ
```

d)
```
10 LET Z1=21
20 PRINT Z1
```

Übung 8

```
2 FOR X=20 TO 0 STEP-2
4 PRINT X
6 NEXT X
```

Übung 9

a) Gut : LET B$ ="COMPUTER"(5TO8)
   Besser:LET B$="COMPUTER"(5TO)
b) LET B$="KOMMANDO"(4TO6)
c) LET B$="RAKETE"(4TO5)
d) LETB$="SPECTRUM"(2TO5)
e) Gut: LET B$="FERNSEHEN"(5TO9)
   Besser:LET B$="FERNSEHEN"(5TO)
f) Gut: LET B$="BRIEFKASTEN"(1TO5)
   Besser:LET B$="BRIEFKASTEN"(TO5)

Stichwortverzeichnis

Addieren (+) 56
Addieren von zwei Zahlen, Programm 35–39
Benutzung der Tastatur 90

CAPS SHIFT 18
CHR$() 83, 91
Code 79
Codieren 79

Dividieren (/) 57, 58
Dollar($) 45
Doppelpunkt 93

Editieren 93
Einschalten 14
ENTER 17

FOR-TO-NEXT 68, 97

Ganze Zufallszahlen 55
Ganzzahl 53
Gerät einschalten 14
Getränkecode 42, 44
GOTO 46, 100
Größer als (>) 61

IF-THEN 46
INPUT 23, 101
INT() 52, 102

Kapitän Barbara Winters Handbuch 51
Kleiner als (<) 62

Laden von Kassetten 18
LEN() 103
LET 26, 104
LIST 30, 107
LOAD 108
Löschen des Bildschirms 27

Mathematische Symbole 56, 109
Mehranweisungs-Zeilen 110
Meldungen 42
Multiplizieren (*) 56, 57

Namen für Speicherplatz 24, 25
Namen-Erkennungsprogramm 47–49
NEW 34, 110
NEXT 69–71

Obere Symbole auf den Tasten 17

PRINT 22, 23, 111
PRINT mit " 25, 111

Ratespiel 60–64
Ratespiel-Strategie 65
Reihenfolge der Zeilennummern 31
REM 113
RND() 55, 114
RUN 15, 33

Sauerstoff 78
SAVE 115
Schalttafel 14
Semikolon 116
Speicherplatz 24, 25, 104
STEP 68–71, 116
String 118
Strom einschalten 14
Subtrahieren (–) 57
SYMBOL SHIFT 17
SYNTAXERROR 119

Tastatur 14
Tasten 15
THEN 46
TO 71–75, 119

Variablennamen 25, 121
VERIFY 121

Zahlen-Ratespiel 60–64
Zeichenketten 25
Zeilennummern 31
Zufallszahlen 52

Auszug aus der SYBEX-Bibliothek

PROGRAMMIERUNG DES Z80
von Rodnay Zaks – ein kompletter Lehrgang in der Programmierung des Z80 Mikroprozessors und eine gründliche Einführung in die Maschinensprache. 608 Seiten, 176 Abbildungen, Format DIN A5, Ref.Nr.: **3006** (1982)

BASIC COMPUTER SPIELE/Band 1
herausgegeben von David H. Ahl – die besten Mikrocomputerspiele aus der Zeitschrift „Creative Computing" in deutscher Fassung mit Probelauf und Programmlisting. 208 Seiten, 56 Abbildungen, Ref.Nr. **3009**

BASIC COMPUTER SPIELE/Band 2
herausgegeben von David H. Ahl – 84 weitere Mikrocomputerspiele aus „Creative Computing". Alle in Microsoft-BASIC geschrieben mit Listing und Probelauf. 224 Seiten, 61 Abbildungen, Ref.Nr.: **3010**

SINCLAIR ZX SPECTRUM Programme zum Lernen und Spielen
von T. Hartnell – ein Buch zur praktischen Anwendung. Grundzüge des Programmierens aus dem kaufmännischen Bereich sowie Spiele, Lehr- und Lernprogramme in BASIC. 232 Seiten, 140 Abbildungen, Ref. Nr. **3022** (1983)

SINCLAIR ZX SPECTRUM BASIC HANDBUCH
von D. Hergert – eine wichtige Hilfe für jeden SPECTRUM-Anwender. Gibt eine Übersicht aller BASIC-Begriffe, die auf diesem Rechner verwendet werden können, und erläutert sie ausführlich anhand von Beispielen. 288 Seiten, 188 Abbildungen, Ref.-Nr.: **3027** (1983)

SYBEX MIKROCOMPUTER LEXIKON
– die schnelle Informationsbörse! Über 1500 Definitionen, Kurzformeln, Begriffsschema der Mikroprozessor-Technik, englisch/deutsches und französisch/deutsches Wörterbuch, Bezugsquellen. 192 Seiten, Format 12,5 x 18 cm, Ref.Nr.: **3005**

Fordern Sie ein Gesamtverzeichnis unserer Verlagsproduktion an:

SYBEX

SYBEX-VERLAG GmbH
Vogelsanger Weg 111
4000 Düsseldorf 30
Tel.: (02 11) 62 64 41
Telex: 8 588 163

SYBEX
6–8, Impasse du Curé
75018 Paris
Tel.: 1/203-95-95
Telex: 211.801 f

SYBEX INC.
2344 Sixth Street
Berkeley, CA 94710, USA
Tel.: (415) 848-8233
Telex: 336311