PRENTICE HALL INTERNATIONAL PERSONAL COMPUTER BOOK

100 PROGRAMS FOR THE COMMODORE 16

IAN McLEAN and JOHN GORDON

100 PROGRAMS FOR THE COMMODORE 16

100 PROGRAMS FOR THE COMMODORE 16

Ian McLean
John Gordon
MEDC, Paisley College, Scotland

Prentice/Hall PHI International

Englewood Cliffs, NJ London Mexico New Dehli Rio de Janeiro
Singapore Sydney Tokyo Toronto Wellington

British Library Cataloging in Publication Data

McLean, Ian, 1946–
100 programs for the Commodore 16.
1. Commodore C16 (Computer)—Programming
I. Title II. Gordon, John, 1952–
001.64'25 QA76.8.C64

ISBN 0-13-634965-X

ISBN 0-13-634965-X

PRENTICE-HALL INC., Englewood Cliffs, New Jersey
PRENTICE-HALL INTERNATIONAL (UK) LTD., London
PRENTICE-HALL OF AUSTRALIA PTY., LTD., Sydney
PRENTICE-HALL CANADA, INC., Toronto
PRENTICE-HALL HISPANOAMERICANA, S.A., Mexico
PRENTICE-HALL OF INDIA PRIVATE LIMITED, New Dehli
PRENTICE-HALL OF JAPAN, INC., Tokyo
PRENTICE-HALL OF SOUTHEAST ASIA PTE., LTD., Singapore
PRENTICE-HALL DO BRASIL LTDA., Rio de Janeiro
WHITEHALL BOOKS LIMITED, Wellington, New Zealand

Printed in the United Kingdom by A. Wheaton & Co Ltd., Exeter

10 9 8 7 6 5 4 3 2 1

To Anne and Teresa

Contents

Section 4 – Point of Sale System (Disk Based Programs)

Section 5 – Programs for the Home

Section 6 – User Defined Characters

Section 7 – Graphics

Section 8 – Data Handling

Section 9 – Recreation

Section 10 – Programs for the School

Section 11 – Science Laboratory

Section 12 – Mathematics

Section 13 – Finale

Introduction

It is with a sense of satisfaction that we present to you this selection of computer programs for your Commodore 16.

We have, in our selection, attempted to answer the question:

"What do you use a microcomputer for?"

You will find routines in this book which cover the use of a micro at home, in business, at school and for pleasure.

The programs in the selection are laid out in the following format:

Program commentary.
Program listing.
On occasion a photograph of the screen or a printout.

The programs themselves exhibit many styles of construction and more than one style of presentation.

We have used two approaches when dealing with the Commodore editing characters. If we had used a Commodore printer then we could have used the normal Commodore symbols for them. We have instead, in some programs, used the CHR$(nn) code for these screen positioning characters. Other programs have been filtered through the "TIDY" program to replace the Commodore characters with the following:

All strings of the screen positioning characters start with the character [and finish with the character].

cursor-home is replaced by CH
clear-screen is replaced by CS
cursor-down is replaced by CD
cursor-up is replaced by CU
cursor-left is replaced by CL
cursor-right is replaced by CR
reverse-on is replaced by RON
reverse-off is relaced by ROF
the math sign pi is replaced by PI

Some of the programs are almost totally "user-friendly" and some are quite terse. Some of the programs are well littered with commentary lines and some are quite sparse.

We have, in our programs, shown off the machine as far as we could.

In the main we have used only the cassette as our backing storage device, but we have included one suite of programs to show the use of the disk system.

We must mention at this point the possibility of bugs. It is possible that we have left a few in the code. Hopefully, through the efforts of Prentice-Hall in carrying out field tests, these will be at a minimum. If there are any left, then we take this opportunity to apologise for them.

However, in a sense, none of these programs is complete. They could all be expanded in various ways. One of the joys of computing is to take a simple routine and give it a professional user-friendly appearance. We have not attempted to make our programs complete in this sense. This is left up to you. Consider this book to form part of your software library.

It might be interesting at this point do describe how this book was produced. The programs were developed, tested and debugged on our C-16s. We used Commodore 1541 Single Disk Drives for program storage at this point. We then created sequential files of the program listings using the C-16s. This was done by OPENing a sequential file on the disk and forcing the listing into it.

Once we had our listings on disk, we then moved over to our Commodore 64s in order to use the EASYSCRIPT word processing package to write the book. We moved over to the C-64 simply because we did not have access to word processing on the C-16s.

100 PROGRAMS FOR THE COMMODORE 16

P1 Dice

A gambler's delight, this program rolls three dice.
This could be extended into a game where you bet against the computer.

COMMANDS

Key in the program and RUN.
Press any key to roll the dice.
RUN/STOP key stops program.

```
10 REM DICE
20 REM ****
30 REM
40 SCNCLR
50 COLOR 0,1:REM BLACK SCREEN
60 COLOR 1,8:REM YELLOW INK
70 COLOR 4,1:REM BLACK BORDER
80 PRINT
90 PRINT TAB(20)"DICE"
100 PRINT:PRINT:PRINT
110 PRINT TAB(12);"THIS PROGRAM ROLLS"
120 PRINT TAB(15);"THREE DICE"
130 REM**********************
140 S$=CHR$(32)
150 T$=S$:E$=S$+S$+S$+S$
160 FOR N=0 TO 4
170 S$=S$+S$
180 NEXT
190 S$=S$+E$
200 REM S$ CONTAINS 36 SPACES
210 R$=CHR$(188)
220 A$=R$+T$+T$
230 B$=T$+R$+T$
240 C$=T$+T$+R$
250 D$=R$+T$+R$
260 REM DOT PATTERNS IN STRINGS
270 DIM DICE$(2,2)
280 REM********************
290 DO
300 :  CHAR ,11,20,"PRESS ANY KEY TO ROLL"
```

```
310 :  GETKEY W$
320 :  CHAR ,0,12,"":REM POSITION CURSOR
330 :  FOR N=0 TO 8
340 :     PRINT S$
350 :  NEXT
360 :  REM CLEAR PREVIOUS THROW
370 :  REM********************
380 :  FOR N=0 TO 2
390 :     VL%=1+6*RND(1)
400 :     ON VL% GOSUB 620,700,770,840,910,980
410 :  NEXT
420 :  REM GENERATE PATTERNS
430 :  REM********************
440 :  CHAR ,0,12,"":REM POSITION CURSOR
450 :  FOR N=0 TO 2
460 :     PRINT,DICE$(0,N),DICE$(1,N),DICE$(2,N)
470 :  NEXT
480 :  REM PRINT PATTERNS
490 :  REM********************
500 :  FOR D=0 TO 200:NEXT:REM DELAY
510 LOOP
520 REM
530 REM         ******************
540 REM         *                *
550 REM         *                *
560 REM         *  SUBROUTINES   *
570 REM         *                *
580 REM         *                *
590 REM         ******************
600 REM
610 REM ONE
620 DICE$(N,0)=E$
630 DICE$(N,1)=B$
640 DICE$(N,2)=E$
650 RETURN
660 REM**********************
670 REM**********************
680 REM TWO
690 REM TWO
700 DICE$(N,0)=A$
710 DICE$(N,1)=E$
720 DICE$(N,2)=C$
730 RETURN
740 REM**********************
750 REM**********************
760 REM THREE
770 DICE$(N,0)=A$
```

```
780 DICE$(N,1)=B$
790 DICE$(N,2)=C$
800 RETURN
810 REM**********************
820 REM**********************
830 REM FOUR
840 DICE$(N,0)=D$
850 DICE$(N,1)=E$
860 DICE$(N,2)=D$
870 RETURN
880 REM**********************
890 REM**********************
900 REM FIVE
910 DICE$(N,0)=D$
920 DICE$(N,1)=B$
930 DICE$(N,2)=D$
940 RETURN
950 REM**********************
960 REM**********************
970 REM SIX
980 DICE$(N,0)=D$
990 DICE$(N,1)=D$
1000 DICE$(N,2)=D$
1010 RETURN
1020 REM**********************
1030 REM**********************
```

P2 Sounds

This program creates some sounds which you might find useful in programs.

COMMANDS

Key in the program and RUN.
Keys 1 to 8 select the sound. Key 9 stops the program.

```
10 REM SOUNDS
20 REM ******
30 REM
40 COLOR 0,7,2:REM BLUE SCREEN
50 COLOR 1,2:REM WHITE INK
60 COLOR 4,4,4:REM CYAN BORDER
70 REM**************************
80 SCNCLR:PRINT
90 PRINT TAB(7)"SELECT SOUND BY PRESSING:"
100 PRINT:PRINT
110 PRINT TAB(9)CHR$(18)" KEY 1 ";
120 PRINT TAB(17)CHR$(146)"- MACHINE GUN"
130 PRINT
140 PRINT TAB(9)CHR$(18)" KEY 2 ";
150 PRINT TAB(17)CHR$(146)"- EXPLOSION"
160 PRINT
170 PRINT TAB(9)CHR$(18)" KEY 3 ";
180 PRINT TAB(17)CHR$(146)"- SHOT AND RICOCHET"
190 PRINT
200 PRINT TAB(9)CHR$(18)" KEY 4 ";
210 PRINT TAB(17)CHR$(146)"- ALARM BELL"
220 PRINT
230 PRINT TAB(9)CHR$(18)" KEY 5 ";
240 PRINT TAB(17)CHR$(146)"- BOUNCE"
250 PRINT
260 PRINT TAB(9)CHR$(18)" KEY 6 ";
270 PRINT TAB(17)CHR$(146)"- BREAKING WAVE"
280 PRINT
290 PRINT TAB(9)CHR$(18)" KEY 7 ";
300 PRINT TAB(17)CHR$(146)"- SURPRISE"
310 PRINT
320 PRINT TAB(9)CHR$(18)" KEY 8 ";
```

```
330 PRINT TAB(17)CHR$(146)"- ALIEN SPACESHIP"
340 PRINT
350 PRINT TAB(9)CHR$(18)" KEY 9 ";
360 PRINT TAB(17)CHR$(146)"- STOP PROGRAM"
370 REM**************************
380 DO
390 :  DO
400 :     GETKEY A$
410 :     SL=ASC(A$)-48
420 :  LOOP UNTIL SL>0 AND SL<10
430 :  ON SL GOSUB 490,580,700,810,950,1030,1120,1250
440 LOOP UNTIL SL=9
450 END
460 REM**************************
470 REM**************************
480 REM MACHINE GUN
490 VOL 8
500 FOR N=0 TO 10
510 :  SOUND 3,1000,2
520 :  FOR DE=0 TO 35:NEXT
530 NEXT
540 RETURN
550 REM**************************
560 REM**************************
570 REM EXPLOSION
580 SOUND 3,800,60
590 FOR A=6 TO 8 STEP .05
600 :  VOL A
610 NEXT
620 SOUND 3,600,40
630 FOR A=8 TO 3 STEP .1
640 :  VOL A
650 NEXT
660 RETURN
670 REM**************************
680 REM**************************
690 REM SHOT AND RICOCHET
700 VOL 8
710 SOUND 3,1000,3
720 FOR DE=0 TO 150:NEXT
730 SOUND 2,1000,15
740 FOR A=6 TO 1 STEP .2
750 :  VOL A
760 NEXT
770 RETURN
780 REM**************************
790 REM**************************
```

```
800 REM ALARM BELL
810 VOL 8
820 FOR K=0 TO 5
830 :  SOUND 1,917,8
840 :  FOR N=0 TO 1
850 :     SOUND 2,881,2
860 :     FOR DE=0 TO 45:NEXT
870 :     SOUND 2,953,2
880 :     FOR DE=0 TO 45:NEXT
890 :  NEXT
900 NEXT
910 RETURN
920 REM**************************
930 REM**************************
940 REM BOUNCE
950 VOL 8
960 FOR F=600 TO 900 STEP 50
970 :  SOUND 1,F,.01
980 NEXT
990 RETURN
1000 REM*************************
1010 REM*************************
1020 REM BREAKING WAVE
1030 VOL 8
1040 SOUND 3,500,60
1050 FOR DE=0 TO 200:NEXT
1060 VOL 6
1070 SOUND 3,800,100
1080 RETURN
1090 REM*************************
1100 REM*************************
1110 REM SURPRISE
1120 VOL 8
1130 FOR F=800 TO 900 STEP 20
1140 :  SOUND 1,F,.04
1150 :  SOUND 2,F+40,.02
1160 :  FOR DE=0 TO 10:NEXT
1170 :  SOUND 2,F-40,.02
1180 :  FOR DE=0 TO 10:NEXT
1190 NEXT
1200 SOUND 1,10,10
1210 RETURN
1220 REM*************************
1230 REM*************************
1240 REM ALIEN SPACESHIP
1250 VOL 4
1260 SOUND 1,800,134
```

```
1270 FOR N=0 TO 4
1280 :  A=6
1290 :  F=900
1300 :  DO UNTIL A<4
1310 :     VOL A
1320 :     SOUND 2,F,.1
1330 :     F=F-10
1340 :     A=A-.1
1350 :  LOOP
1360 NEXT
1370 RETURN
1380 REM**************************
1390 REM**************************
```

P3 Musak

This program turns your Commodore 16 into a musical instrument. The bottom three rows of keys select three octaves. Keys 1 to 9 select note length.

While using this program we found that there was a delay between pressing a key and hearing the corresponding note. This can be reduced by removing REM statements and putting as many instructions as possible in a single line. We could not eliminate it completely.

COMMANDS

Key in the program and RUN.
Stop the program by pressing the RETURN key.

```
10 REM MUSAK
20 REM *****
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,1:REM BLUE INK
60 COLOR 4,10,4:REM BROWN BORDER
70 SCNCLR
80 PRINT:PRINT
90 PRINT TAB(16)CHR$(18)"-------"
100 PRINT TAB(16)CHR$(18)" MUSAK "
110 PRINT TAB(16)CHR$(18)"-------"
120 PRINT:PRINT
130 PRINT TAB(6)"THIS PROGRAM LETS YOU PLAY"
140 PRINT TAB(6)"MUSIC ON YOUR COMPUTER."
150 PRINT
160 PRINT TAB(6)"1 TO 9 SELECT NOTE LENGTH;"
170 PRINT
180 PRINT TAB(6)"Q W E R T Y U I O P  +"
190 PRINT TAB(6)"A S D F G H J K L : ; *"
200 PRINT TAB(6)"Z X C V B N M , . / £ ="
210 PRINT TAB(6)"SELECT NOTE."
220 PRINT
230 PRINT TAB(6)"RETURN KEY ENDS PROGRAM"
240 REM**********************
250 DIM FR(35):REM HOLDS NOTE FREQUENCIES
```

```
260 FOR N=0 TO 35:READ FR(N):NEXT
270 KB$(0)="+POIUYTREWQ"
280 KB$(1)="*;:LKJHGFDSA"
290 KB$(2)="=\/.,MNBVCXZ"
300 REM************************
310 REM INITIAL CONDITIONS
320 VOL 8
330 DR=4:REM DEFAULT NOTE LENGTH
340 REM************************
350 REM PLAY
360 DO
370 :  DO:GET A$:LOOP UNTIL LEN(A$)>0:
380 :  REM ABOVE IS FASTER THAN GETKEY A$
390 :  A=ASC(A$)
400 :  F=0
410 :  IF A>48 AND A<58 THEN DR=A-48:F=1:REM LENGTH
420 :  IF F=0 THEN GOSUB 720:REM FREQUENCY
430 LOOP UNTIL ASC(A$)=13
440 VOL 0
450 END
460 REM************************
470 REM************************
480 REM
490 REM         ********
500 REM         *      *
510 REM         * DATA *
520 REM         *      *
530 REM         ********
540 REM
550 REM************************
560 REM************************
570 REM NOTE FREQUENCY DATA
580 DATA 917,911,904,897,889,881,873,864,854,844,834,822
590 DATA 810,798,784,770,755,739,722,704,685,664,643,620
600 DATA 596,571,544,516,485,453,419,383,345,305,262,217
610 REM************************
620 REM************************
630 REM
640 REM      ***************
650 REM      *             *
660 REM      * SUBROUTINES *
670 REM      *             *
680 REM      ***************
690 REM
700 REM************************
710 REM************************
```

```
720 REM FREQUENCY
730 FOR M=0 TO 2
740 :  FOR N=1 TO 12
750 :     IF MID$(KB$(M),N,1)=A$ THEN F=1:CT=M:NT=N-1
760 :     REM CT=OCTAVE, NT=NOTE
770 :     IF F=1 THEN M=2:N=12
780 :  NEXT
790 NEXT
800 E=CT*12+NT:NO=FR(E):REM NOTE FREQUENCY
810 IF CT=2 THEN V=2:ELSE V=1:REM VOICE
820 IF F=1 THEN SOUND V,NO,0:SOUND V,NO,6*DR
830 RETURN
```

P4 Bouncing Ball

This program shows a simple method of achieving animation using CHAR instructions. Animation is achieved by placing the ball on the screen and then placing a space on top. The ball is then placed one position on.

COMMANDS

Key in the program and RUN.
The RUN/STOP key stops the program.

```
10 REM BOUNCING BALL
20 REM *************
30 REM
40 COLOR 0,8,7:REM YELLOW PAPER
50 COLOR 1,1:REM BLACK PRINT
60 COLOR 4,15,4:REM BLUE BORDER
70 VOL 8
80 X=1:Y=INT(1+23*RND(1))
90 DY=1:DX=1:IF RND(1)<.5 THEN DY=-1
100 SCNCLR
110 DO
120 :  CHAR 1,X,Y,CHR$(113)
130 :  FOR N=0 TO 25:NEXT:REM REDUCES FLICKER
140 :  CHAR 0,X,Y," "
150 :  IF X=0 OR X=39 THEN DX=-DX:SOUND 3,1000,2
160 :  IF Y=0 OR Y=24 THEN DY=-DY:SOUND 3,1000,2
170 :  X=X+DX:Y=Y+DY
180 LOOP
```

P5 Ball and wall

This is an extension of the Bouncing Ball program, except that the POKE rather than the CHAR instruction is used. One of two alternative bounce angles may be chosen, depending on whether the ball hits the face or the corner of a brick. You could expand this program by introducing a bat to control the ball and having several layers of wall.

COMMANDS

Key in the program and RUN.
The RUN/STOP key stops the program.

```
10 REM BALL AND WALL
20 REM *************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 4,7,2:REM BLUE BORDER
60 SC=3072:REM START OF SCREEN MEMORY
70 CL=1024:REM DIFFERENCE BETWEEN SCREEN AND COLOR MEMORY.
80 X=1:Y=INT(1+23*RND(1))
90 DY=1:DX=1:IF RND(1)<.5 THEN DY=-1
100 VOL7
110 GOSUB 300:REM BUILD WALL
120 DO
130 :   P=SC+40*Y+X:REM POSITION ON SCREEN
140 :   C=P-CL:REM COLOR MEMORY
150 :   POKE P,81:POKE C,3
160 :   GOSUB 390:REM TEST
170 :   IF X=0 OR X=39 OR F=1 THEN DX=-DX:SOUND 3,200*(4-F),2
180 :   IF Y=0 OR Y=24 THEN DY=-DY:SOUND 3,800,2
190 :   X=X+DX:Y=Y+DY
195 :   POKE P,32
200 LOOP
210 REM**************************
220 REM**************************
230 REM
240 REM       ***************
250 REM       *             *
260 REM       * SUBROUTINES *
270 REM       *             *
280 REM       ***************
```

```
290 REM
300 REM WALL
310 SCNCLR
320 FOR N=0 TO 24
330 :  POKE SC+30+N*40,102
340 :  POKE SC+30+N*40-CL,66
350 NEXT
360 RETURN
370 REM**************************
380 REM**************************
390 REM TEST
400 F=0
410 IF X<>29 AND X<>31 THEN RETURN:REM BALL NOT NEXT TO WALL
420 IF X=29 AND DX=-1 THEN RETURN:REM MOVING AWAY FROM WALL
430 IF X=31 AND DX=1 THEN RETURN:REM MOVING AWAY FROM WALL
440 IF PEEK(P+DX)=102 THEN F=1:POKE(P+DX),32:RETURN
450 REM BALL HITS FACE OF BRICK
460 CR=P+40*DY+DX
470  IF  PEEK(P+40*DY+DX)=102  THEN  F=1:POKE(P+40*DY+DX),32:
DY=-DY
480 REM BALL HITS CORNER OF BRICK
490 RETURN
500 REM**************************
510 REM**************************
```

P6 Digital clock

This program uses the micro's internal timer to run a 24 hour digital clock. It could be extended so that several zone times are displayed simultaneously.

COMMANDS

Key in the program and RUN.
Enter the current time.

```
10 REM DIGITAL CLOCK
20 REM *************
30 REM
40 UP$=CHR$(145):REM UP CURSOR
50 S$=CHR$(32)
60 FOR N=0 TO 2:S$=S$+S$+S$:NEXT:REM S$ CONTAINS 27 SPACES
70 COLOR 0,1:REM BLACK SCREEN
80 COLOR 1,6:REM GREEN INK
90 COLOR 4,1:REM BLACK BORDER
100 REM**************************
110 SCNCLR:PRINT:PRINT
120 PRINT TAB(10)"24 HOUR DIGITAL CLOCK"
130 PRINT TAB(10)"++++++++++++++++++++"
140 PRINT:PRINT
150 PRINT TAB(11)"ENTER INITIAL TIME"
160 PRINT
170 DO
180 :  PRINT TAB(11):INPUT"HOURS";H%
190 :  IF H%<0 OR H%>23 THEN PRINT UP$+S$+UP$:T=1:ELSE T=0
200 LOOP UNTIL T=0
210 A$=STR$(H%):IF LEN(A$)=2 THEN A$="0"+RIGHT$(A$,1)
220 IF LEN(STR$(H%))=3 THEN A$=RIGHT$(STR$(H%),2)
230 PRINT
240 DO
250 :  PRINT TAB(11):INPUT"MINUTES";M%
260 :  IF M%<0 OR M%>59 THEN PRINT UP$+S$+UP$:T=1:ELSE T=0
270 LOOP UNTIL T=0
280 B$=STR$(M%):IF LEN(B$)=2 THEN B$="0"+RIGHT$(B$,1)
290 IF LEN(STR$(M%))=3 THEN B$=RIGHT$(STR$(M%),2)
300 PRINT
```

```
310 DO
320 :  PRINT TAB(11):INPUT"SECONDS";S%
330 :  IF S%<0 OR S%>59 THEN PRINT UP$+S$+UP$:T=1:ELSE T=0
340 LOOP UNTIL T=0
350 C$=STR$(S%):IF LEN(C$)=2 THEN C$="0"+RIGHT$(C$,1)
360 IF LEN(STR$(S%))=3 THEN C$=RIGHT$(STR$(S%),2)
370 TI$=A$+B$+C$:REM SET CLOCK
380 SCNCLR
390 REM**************************
400 DO
410 :  IF LEFT$(TI$,2)="24" THEN TI$="000000"
420 :  D$=LEFT$(TI$,2)+":"+MID$(TI$,3,2)+":"+RIGHT$(TI$,2)
430 :  CHAR,15,10,D$
440 LOOP
```

P7 Blobs

This program uses the CIRCLE and PAINT commands to plot random regular shapes on the screen.

COMMANDS

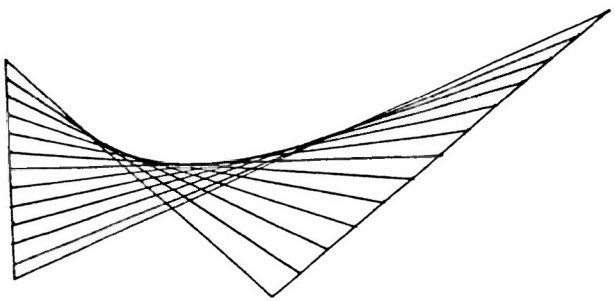
Key in the program and RUN.
Press space bar for new screen, Q to quit.

```
10 REM THIS PROGRAM USES THE CIRCLE
20 REM COMMAND TO CREATE A NICE EFFECT
25 A$="C"
30 DO WHILE A$<>"Q"
40 GRAPHIC 3,1
41 BG=INT(RND(1)*16+1)
42 FG=INT(RND(1)*16+1):IF BG=FG THEN 41
43 M1=INT(RND(1)*16+1):IF M1=FG OR M1=BG THEN 43
44 M2=INT(RND(1)*16+1):IF M2=FG OR M2=BG OR M2=M1 THEN 44
45 COLOR0,BG:COLOR 1,FG:COLOR 2,M1:COLOR 3,M2
50 FOR L=1 TO 100
60 X=INT(RND(1)*315+1):Y=INT(RND(1)*190+1)
70 SD=INT(RND(1)*5+3):C=INT(RND(1)*3+1)
80 HT=INT(RND(1)*20+1):WT=INT(RND(1)*20+1)
90 CIRCLE C,X,Y,HT,WT,,,,360/SD
95 PAINT C,X,Y
100 NEXT L
110 GET A$:IF A$="" THEN 110
120 LOOP
130 GRAPHIC 0
140 COLOR 0,7:COLOR 1,1
150 END
```

P8 String pattern

This program generates patterns of straight lines reminiscent of 'pin pictures'. It first obtains the end points of the reference lines and the number of points per line. The equations of the lines in the form y=m*x+c are calculated, followed by the step sizes.

The program then steps down each line, drawing straight lines to produce a pattern thus:



COMMANDS

Key in the program and RUN.
Enter information as requested.
Press any key to return to text mode.

```
10 REM STRING PATTERN
20 REM **************
30 REM
40 COLOR 0,1:REM BLACK SCREEN
50 COLOR 1,4,5:REM CYAN INK
60 COLOR 4,1:REM BLACK BORDER
70 REM****************************
80 FOR N=0 TO 3
90 :  DO
100 :      SCNCLR
110 :      IF N=1 OR N=3 THEN M$="END":ELSE M$="START"
120 :      PRINT M$" POINT OF LINE";1+INT(N/2)
130 :      INPUT"X-COORDINATE (0 TO 319)";X%(N)
```

```
140 :  LOOP UNTIL X%(N)>-1 AND X%(N)<320
150 :  REM************************
160 :  DO
170 :     SCNCLR
180 :     PRINT M$" POINT OF LINE";1+INT(N/2)
190 :     INPUT"Y-COORDINATE (0 TO 199)";Y%(N)
200 :  LOOP UNTIL Y%(N)>-1 AND Y%(N)<200
210 NEXT
220 DO
230 :  SCNCLR
240 :  INPUT"NUMBER OF POINTS PER LINE";K%
250 LOOP UNTIL K%>0
260 REM*****************************
270 REM PREVENT DIVISION BY ZERO
280 IF X%(0)=X%(1) THEN X%(1)=X%(1)+1
290 IF X%(2)=X%(3) THEN X%(3)=X%(3)+1
300 REM*****************************
310 REM CALCULATE PARAMETERS
320 D1=(X%(1)-X%(0))/K%
330 D2=(X%(3)-X%(2))/K%
340 M1=(Y%(1)-Y%(0))/(X%(1)-X%(0))
350 M2=(Y%(3)-Y%(2))/(X%(3)-X%(2))
360 B1=Y%(0)-M1*X%(0)
370 B2=Y%(2)-M2*X%(2)
380 REM*****************************
390 REM DRAW PATTERN
400 GRAPHIC 1,1
410 DRAW 1,X%(0),Y%(0) TO X%(1),Y%(1)
420 DRAW 1,X%(2),Y%(2) TO X%(3),Y%(3)
430 FOR N=1 TO K%
440 :  R1=X%(0)+N*D1
450 :  S1=M1*(X%(0)+N*D1)+B1
460 :  R2=X%(3)-N*D2
470 :  S2=M2*(X%(3)-N*D2)+B2
480 :  DRAW 1,R1,S1 TO R2,S2
490 NEXT
500 GETKEY A$:REM ANY KEY TO END
510 GRAPHIC CLR
520 SCNCLR
530 END
```

P9 Soft Keys

This program can be very usefully adapted to set up the soft keys at the right of the keyboard. This can be used in program development to build in your most useful command strings.

COMMANDS

Key in the program and RUN.

```
10 REM PROGRAMMING THE SOFT KEYS
20 REM
30 SCNCLR
40 PRINT:PRINT:PRINT
50 PRINT "THIS PROGRAM ALLOWS YOU TO RE-PROGRAM"
60 PRINT "THE FUNCTION KEYS AT THE RIGHT HAND"
70 PRINT "SIDE OF THE KEYBOARD"
80 PRINT:PRINT
90 PRINT "YOU CAN ENTER UP TO 128 CHARACTERS"
100 PRINT "OVER ALL 7 KEYS"
110 PRINT:PRINT
120 PRINT "ENTER STRING TO BE ENTERED TO EACH "
130 PRINT "KEY, TERMINATING EACH WITH THE ESC KEY"
140 :
150 PRINT:PRINT:PRINT "PRESS ANY KEY TO START"
160 GET B$:IF B$="" THEN 160
170 SCNCLR
180 CH=1:F$="":K=1
190 DO UNTIL CH=128 OR F$="FINISHED"
200 PRINT:PRINT
210 PRINT "PROGRAMMING KEY "K
220 A$="":K$=""
230 DO UNTIL ASC(A$)=27 OR CH=128
240 GET A$:IF A$="" THEN 240
250 IF ASC(A$)=27 THEN 280
260 IF ASC(A$)=13 THEN PRINT CHR$(95); :ELSE PRINT A$;
270 K$=K$+A$
280 CH=CH+1
290 IF CH=128 THEN PRINT:PRINT "YOU HAVE USED UP 128
CHARACTERS"
300 LOOP
```

```
310 KEY K,K$
320 K=K+1:IF K=8 THEN F$="FINISHED"
330 LOOP
```

P10 Guess the number

In this game the computer generates a random whole number between 1 and 100 and the player has to guess what it is.

The instuctions for the game are included in the code. Remember to press the RETURN key after typing in your guess.

COMMANDS

Key in the program and RUN.

```
10 REM GUESS THE NUMBER
20 REM ****************
30 REM
40 COLOR 0,10,1:REM BROWN SCREEN
50 COLOR 1,8:REM YELLOW INK
60 COLOR 4,14:REM BLUE BORDER
70 SCNCLR:PRINT
80 PRINT TAB(10)CHR$(18)"??????????????????????"
90 PRINT TAB(10)CHR$(18)"?                      ?"
100 PRINT TAB(10)CHR$(18)"? GUESS THE NUMBER ?"
110 PRINT TAB(10)CHR$(18)"?                      ?"
120 PRINT TAB(10)CHR$(18)"??????????????????????"
130 PRINT:PRINT
140 PRINT TAB(5)"IN THIS PROGRAM YOU ARE ASKED"
150 PRINT TAB(5)"TO GUESS A NUMBER BETWEEN 1"
160 PRINT TAB(5)"AND 100."
170 PRINT
180 PRINT TAB(5)"IF YOU GUESS WRONGLY THE"
190 PRINT TAB(5)"COMPUTER WILL TELL YOU WHETHER"
200 PRINT TAB(5)"YOU ARE TOO HIGH OR TOO LOW."
210 PRINT
220 PRINT TAB(5)"WHEN YOU ARE FINISHED THE"
230 PRINT TAB(5)"COMPUTER WILL TELL YOU THE"
240 PRINT TAB(5)"AVERAGE NUMBER OF ATTEMPTS"
250 PRINT TAB(5)"YOU TOOK."
260 PRINT:PRINT
270 PRINT TAB(7)CHR$(18)" PRESS ANY KEY TO PLAY "
280 GETKEY A$
290 REM*****************************
300 DO
```

```
310 : AT=0:PL=0
320 : DO
330 :  SCNCLR:PRINT:PRINT
340 :  N%=1+100*RND(1)
350 :  FOR K=1 TO 100
360 :     AT=AT+1
370 :      PRINT TAB(6):INPUT"WHAT IS YOUR GUESS";G
380 :      IF G>N% THEN PRINT TAB(6)"TOO HIGH"
390 :      IF G<N% THEN PRINT TAB(6)"TOO LOW"
400 :      IF G=N% THEN PRINT TAB(6)"CORRECT":K=100:PL=PL+1
410 :      PRINT
420 :  NEXT
430 :  PRINT
440 :  PRINT TAB(6):INPUT"WANT TO TRY AGAIN (Y/N)";Y$
450 : LOOP UNTIL ASC(Y$)<>89
460 : REM**************************
470 : SCNCLR
480 : AV=AT/PL
490 : AV=(INT(100*AV+.5))/100
500 : CHAR,6,5,"YOU TOOK AN AVERAGE OF:"
510 : PRINT
520 : PRINT TAB(5)AV;"PER SHOT"
530 : PRINT:PRINT
540 : IF AV<7 THEN PRINT TAB(6) CHR$(18)" PRETTY GOOD! "
550 : PRINT:PRINT
560 : PRINT TAB(6):INPUT"WANT TO PLAY AGAIN (Y/N)";Y$
570 LOOP UNTIL ASC(Y$)<>89
580 END
590 REM*****************************
600 REM*****************************
```

P11 Reaction test

This program could help develop your keyboard skills. The computer places a random character on the screen and starts to time your response.

The object of the game is to press the required key as quickly as possible.

When the program is complete an average reaction time is displayed on the screen.

COMMANDS

Key in the program and RUN.

```
10 REM REACTION TEST
20 REM *************
30 REM
40 COLOR 0,1:REM BLACK BORDER
50 COLOR 1,8:REM YELLOW INK
60 COLOR 4,1:REM BLACK SCREEN
70 DO
80 :  SCNCLR
90 :  CHAR,13,2,"REACTION TEST"
100 :  CHAR,13,3,"-------------"
110 :  CHAR,4,8,""
120 :  INPUT"HOW MANY TRIES DO YOU WANT";TR%
130 LOOP UNTIL TR%>0
140 TI$="000000":REM INITIAL TIME VALUE
150 REM*****************************
160 FOR N=1 TO TR%
170 :  SCNCLR
180 :  A%=65+26*RND(1)
190 :  X%=1+38*RND(1)
200 :  Y%=1+22*RND(1)
205 :  CHAR,X%,Y%,CHR$(A%)
210 :  DO
220 :     GETKEY A$
230 :  LOOP UNTIL ASC(A$)=A%
240 NEXT
250 REM*****************************
```

```
260 T%=TI/60:REM TOTAL TIME IN SECONDS
270 SCNCLR:PRINT:PRINT:PRINT
280 PRINT TAB(8)"NUMBER OF ATTEMPTS"TR%
290 PRINT:PRINT
300 PRINT TAB(8)"TOTAL TIME"T%"SECONDS"
310 PRINT:PRINT
320 AT=INT((T%/TR%)*100+.5)/100
330 PRINT TAB(8)"AVERAGE REACTION TIME":PRINT
340 PRINT TAB(7)AT"SECONDS"
350 PRINT:PRINT:PRINT
360 PRINT TAB(8):INPUT"WANT ANOTHER GO (Y/N)";Y$
370 IF ASC(Y$)=89 THEN RUN
380 END
390 REM*****************************
400 REM*****************************
```

P12 Mastermind

This program implements the first version of the popular game by Invicta Ltd.

The object of the game is to determine the color of four rectangles on the screen. The player has up to twenty attempts to work out the code.

When an attempt has been entered, the computer responds by indicating whether you have a correctly colored rectangle in the correct position, or a correctly colored rectangle in the wrong position.

For each correct color in the correct position, the computer places a purple dash to the right of the guess.

For each correct color in the wrong position the computer places a cyan dash to the right of the guess.

COMMANDS

Key in the program and RUN.
Enter your guess as e.g. RGYB. If you enter an invalid color a purple rectangle will result.

```
10 REM MASTERMIND
20 REM **********
30 REM
40 REM****************************
50 H$=CHR$(19):REM HOME CURSOR
60 S$=CHR$(32):F$=S$:REM SPACE
70 FOR N=0 TO 18:S$=S$+F$+F$:NEXT
80 REM S$ CONTAINS 64 SPACES
90 COLOR 0,1:REM BLACK SCREEN
100 COLOR 4,1:REM BLACK BORDER
110 REM****************************
120 REM INITIAL LINE
130 SCNCLR:PRINT:PRINT:PRINT:PRINT
140 P(0)=156:P(1)=156:P(2)=156:P(3)=156
150 GOSUB 730:REM PRINT LINE
160 REM****************************
```

```
170 REM SET UP TARGET PATTERN
180 FOR N=0 TO 3
190 T(N)=1+INT(4*RND(1)):C(N)=T(N)
200 NEXT:REM T(N) HOLDS CORRECT ANSWER
210 REM*****************************
220 REM PLAY
230 DO
240 :  L=L+1
250 :  FOR N=0 TO 3:C(N)=T(N):NEXT
260 :  REM*****************************
270 :  REM GET GUESS
280 :  DO
290 :     PRINT CHR$(158)H$;
300 :     INPUT "ENTER COLOR PATTERN (E.G. RGYB)";G$
310 :     IF LEN(G$)<>4 THEN PRINT H$;S$
320 :  LOOP UNTIL LEN(G$)=4
330 :  REM*****************************
340 :  GOSUB 820:REM SET UP ARRAYS
350 :  PRINT H$;S$
360 :  CHAR ,0,L+4,"":REM POSITION CURSOR
370 :  GOSUB 730:REM PRINT LINE
380 :  REM*****************************
390 :  REM TEST FOR CORRECT COLOR IN CORRECT POSITION
400 :  PC=0
410 :  FOR N=0 TO 3
420 :     IF G(N)=C(N) THEN PC=PC+1:G(N)=8:C(N)=9
430 :     REM CHANGE G(N) AND C(N) ON MATCH
440 :  NEXT
450 :  REM*****************************
460 :  REM TEST FOR CORRECT COLOR IN WRONG POSITION
470 :  CC=0
480 :  FOR N=0 TO 3
490 :     FOR K=0 TO 3
500 :        IF G(N)=C(K) THEN CC=CC+1:C(K)=9:K=3
510 :     NEXT
520 :  NEXT
530 :  REM*****************************
540 :  REM ACT ON RESULTS
550 :  IF PC>0 AND PC<4 THEN GOSUB 1170
560 :  IF CC>0 THEN GOSUB 1240
570 LOOP UNTIL L=20 OR PC=4
580 IF PC=4 THEN GOSUB 1080:ELSE GOSUB 930
590 REM*****************************
600 REM FINISH
610 PRINT:PRINT:PRINT CHR$(158)
620 PRINT TAB(9):INPUT"ANOTHER GAME (Y/N)";Y$
630 IF ASC(Y$)=89 THEN RUN
```

```
640 END
650 REM
660 REM         ***************
670 REM         *             *
680 REM         * SUBROUTINES *
690 REM         *             *
700 REM         ***************
710 REM
720 REM GENERATE AND PRINT LINE
730 A$="":REM NO SPACE
740 FOR N=0 TO 3
750 A$=A$+CHR$(P(N))+CHR$(32)+CHR$(184)
760 NEXT
770 PRINT TAB(14)A$;
780 RETURN
790 REM****************************
800 REM****************************
810 REM SET UP ARRAYS
820 FOR N=0 TO 3
830 G(N)=0:P(N)=156
840 IF MID$(G$,N+1,1)="R" THEN G(N)=1:P(N)=28
850 IF MID$(G$,N+1,1)="G" THEN G(N)=2:P(N)=30
860 IF MID$(G$,N+1,1)="B" THEN G(N)=3:P(N)=31
870 IF MID$(G$,N+1,1)="Y" THEN G(N)=4:P(N)=158
880 NEXT
890 RETURN
900 REM****************************
910 REM****************************
920 REM PATTERN NOT FOUND
930 SCNCLR:PRINT:PRINT:PRINT
940 PRINT CHR$(158):REM YELLOW INK
950 PRINT TAB(8)"THE CORRECT PATTERN WAS"
960 PRINT:PRINT
970 FOR N=0 TO 3
980 :    IF T(N)=1 THEN P(N)=28
990 :    IF T(N)=2 THEN P(N)=30
1000 :   IF T(N)=3 THEN P(N)=31
1010 :   IF T(N)=4 THEN P(N)=158
1020 NEXT
1030 GOSUB 720
1040 RETURN
1050 REM****************************
1060 REM****************************
1070 REM CORRECT GUESS
1080 SCNCLR:PRINT:PRINT:PRINT
1090 PRINT CHR$(158):REM YELLOW INK
1100 PRINT TAB(14)"CORRECT"
```

```
1110 PRINT:PRINT
1120 PRINT TAB(10)"YOU TOOK";L;"GOES."
1130 RETURN
1140 REM**************************
1150 REM**************************
1160 REM COLOR AND POSITION
1170 FOR N=1 TO PC
1180 PRINT CHR$(156)CHR$(32)CHR$(188);
1190 NEXT
1200 RETURN
1210 REM**************************
1220 REM**************************
1230 REM POSITION
1240 FOR N=1 TO CC
1250 PRINT CHR$(159)CHR$(32)CHR$(188);
1260 NEXT
1270 RETURN
1280 REM**************************
1290 REM**************************
```

P13 Tic tac toe

This program implements the old game of tic tac toe or noughts and crosses.

COMMANDS

Key in the program and RUN.
Follow instructions and prompts as required.

```
10 REM TIC-TAC-TOE
20 REM THIS PROGRAM ALLOWS TWO PLAYERS
30 REM TO PLAY THE GAME OF TIC-TAC-TOE
40 REM OTHERWISE KNOWN AS NOUGHTS AND
50 REM CROSSES
60 PRINT CHR$(147)
70 DIM B$(3,3),T(3,3):REM STRING AND NUMERIC REPRESENTATION
OF BOARD
80 DIM C$(2):C$(1)="X":C$(2)="O"
90 FOR I=1 TO 3:FOR J=1 TO 3:B$(I,J)=" ":NEXT J,I
100 :
110 REM PRINT OUT BOARD BACKGROUND
120 Y=6:X=15:GOSUB 540
130 G1$=CHR$(32)+CHR$(98)+CHR$(32)+CHR$(98)+CHR$(32)
140 G2$=CHR$(99)+CHR$(123)+CHR$(99)+CHR$(123)+CHR$(99)
150 PRINT G1$
160 PRINT SPC(X);G2$:PRINT SPC(X);G1$
170 PRINT SPC(X);G2$:PRINT SPC(X);G1$
180 :
190 :
200 Y=20:X=1:GOSUB 540
210 PRINT "PLAYER 1 IS 'X' PLAYER 2 IS 'O'"
220 FL$="":P=1
230 G=1
240 DO UNTIL FL$="FINISHED"
250 Y=17:X=1:GOSUB 540
260 INPUT "ROW NUMBER ";R%
270 IF R%<1 OR R%>3 THEN 250
280 Y=18:X=1:GOSUB 540
290 INPUT "COLUMN NUMBER";C%
300 IF C%<1 OR C%>3 THEN 280
```

```
310 IF B$(R%,C%)<>" " THEN PRINT "   ALREADY USED, PRESS ANY
KEY" :ELSE GOTO 340
320 GET A$:IF A$="" THEN 320
330 Y=19:X=0:GOSUB 540
335 PRINT "                                          ";:GOTO 250
340 B$(R%,C%)=C$(P)
350 Y=4+2*R%:X=13+2*C%:GOSUB 540:PRINT B$(R%,C%)
360 GOSUB 640:REM TEST FOR END
370 IF FL$<>"FINISHED" THEN P=3-P
380 G=G+1:IF G=10 THEN FL$="FINISHED"
390 LOOP
400 X=1:Y=1:GOSUB 540
410 IF G=10 THEN PRINT "GAME DRAWN":END
420 IF WC=0 THEN 450
430 IF B$(1,WC)="X" THEN PRINT "PLAYER 1 IS WINNER" :ELSE
PRINT "PLAYER 2 IS WINNER"
440 END
450 IF WR=0 THEN 480
460 IF B$(WR,1)="X" THEN PRINT "PLAYER 1 IS WINNER" :ELSE
PRINT "PLAYER 2 IS WINNER"
470 END
480 IF B$(2,2)="X" THEN PRINT "PLAYER 1 IS WINNER" :ELSE
PRINT "PLAYER 2 IS WINNER"
490 END
500 :
510 :
520 :
530 :
540 REM SUBROUTINE TO POSITION CURSOR
550 REM FOR PRINTING
560 REM CURSOR POSITIONED AT POINT X,Y
570 PRINT CHR$(19)
580 FOR I=1 TO Y:PRINT:NEXT I
590 PRINT SPC(X);
600 RETURN
610 :
620 :
630 :
640 REM TEST FOR END OF GAME
650 :
660 :
670 REM TEST FOR ROW WINNER
680 FOR RW=1 TO 3
690 WN=(B$(RW,1)=B$(RW,2)) AND (B$(RW,2)=B$(RW,3))
700 IF WN AND B$(RW,1)<>" " THEN WR=RW:FL$="FINISHED":RW=3
710 NEXT RW
720 :
```

```
730 :
740 REM TEST FOR COLUMN WIN
750 FOR CL=1 TO 3
760 WN=(B$(1,CL)=B$(2,CL)) AND (B$(2,CL)=B$(3,CL))
770 IF WN AND B$(1,CL)<>" " THEN WC=CL:FL$="FINISHED":CL=3
780 NEXT CL
790 :
800 :
810 REM TEST FOR DIAGONAL
820 WN=B$(1,1)=B$(2,2) AND B$(2,2)=B$(3,3)
830 IF WN AND B$(1,1)<>" " THEN FL$="FINISHED"
840 :
850 :
860 REM TEST FOR OTHER DIAGONAL
870 WN=B$(3,1)=B$(2,2) AND B$(2,2)=B$(1,3)
880 IF WN AND B$(2,2)<>" " THEN FL$="FINISHED"
890 :
900 :
910 RETURN
```

P14 Loan repayment period

This program uses the formula

$$T = -\frac{1}{N} * \frac{\log(1-(P.R)/(N.A))}{\log(1+R/N)}$$

where T=Period in years
P=Principal
R=Rate of interest
N=Number of payments each year
A=Amount of each payment.

This could be calculated by using a calculator, but it is far quicker to allow the computer to do the work for you.

This program could be improved by designing a more robust input routine, to check for bad keyboard input.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM  - LOAN REPAYMENT PERIOD
20 PRINT "[CS ]":PRINT:PRINT:PRINT
30 PRINT "IF YOU ARE ABOUT TO TAKE OUT A LOAN"
40 PRINT "IT COULD BE USEFUL TO CONSIDER HOW LONG"
50 PRINT "IT WILL BE BEFORE THE LOAN IS REPAID."
60 PRINT :PRINT :PRINT
70 PRINT "TO USE THIS PROGRAM YOU MUST INPUT"
80 PRINT:PRINT
90 PRINT TAB(5);"AMOUNT BORROWED"
100 PRINT TAB(5);"NUMBER OF PAYMENTS PER YEAR"
110 PRINT TAB(5);"AMOUNT OF REPAYMENTS"
120 PRINT:PRINT:PRINT
```

```
130 PRINT "PRESS ANY KEY TO CONTINUE"
140 GET A$:IF A$="" THEN GOTO 140
150 PRINT "[CS ]"
160 FOR Z=1 TO 9:PRINT:NEXT Z
170 PRINT TAB(5);"AMOUNT BORROWED  $"
180 PRINT TAB(5);"ANNUAL INTEREST RATE(%)"
190 PRINT TAB(5);"NUMBER OF PAYMENTS PER YEAR"
200 PRINT TAB(5);"AMOUNT OF PAYMENTS $"
210 Y=10:X=24:GOSUB 5000
220 INPUT P
230 Y=11:X=28:GOSUB 5000
240 INPUT R:R=R/100
250 Y=12:X=32:GOSUB 5000
260 INPUT N
270 Y=13:X=25:GOSUB 5000
280 INPUT A
290 TM=-LOG(1-P*R/N/A)/LOG(1+R/N)/N
300 YR=INT(TM)
310 MT=INT(12*(TM-YR))+1
315 IF MT=12 THEN MT=0:YR=YR+1
320 PRINT:PRINT:PRINT
330 PRINT "LOAN WILL BE PAID OFF IN "
340 PRINT YR;" YEARS AND "
350 PRINT MT;" MONTHS"
360 END
4990 REM SUBROUTINE TO PLACE CURSOR AT X,Y
5000 PRINT "[CH ]";:REM CURSOR HOME
5010 FOR I=1 TO Y
5020 PRINT "[CD ]";
5030 NEXT I
5040 FOR I=1 TO X
5050 PRINT "[CR ]";
5060 NEXT I
5070 RETURN
```

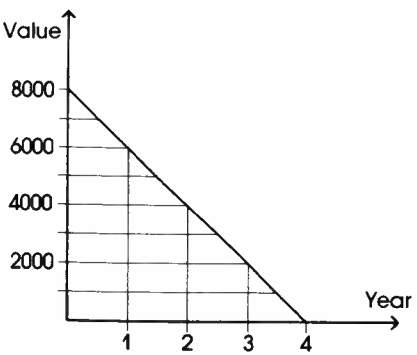
P15 Depreciation

This program may be used to calculate the depreciation in the value of an article arising from normal use through time.

The program shows the effect of two common methods of calculating this depreciation.

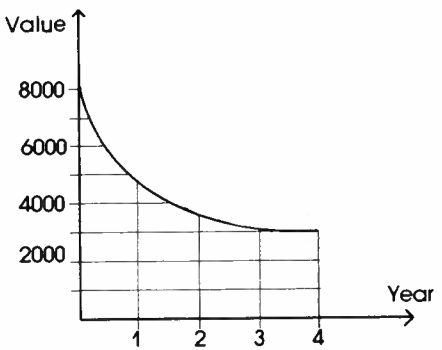
1. The straight line method

   Under this method a fixed amount (a percentage of the initial value) is written off annually. The resultant graph shows a linear relationship between value and year. If, for example, the initial value was $8000, and 25% of this original value was written off each year, then we have:



2. The diminishing balance method

Under this method a percentage of the residual value at the beginning of each year is written down at the end of that year. This gives a curve like:



The program presents the two depreciation methods in the form of a table showing the amount to be written off over a period of years.

COMMANDS

Key in the program and RUN.
Follow the instructions.
Enter amounts as numbers only. This program works with any currency.

```
10 REM DEPRECIATION
20 REM ************
30 REM
40 SCNCLR
50 COLOR 0,15,1:REM DARK BLUE SCREEN
60 COLOR 1,8:REM YELLOW INK
70 COLOR 4,15,1:REM DARK BLUE BORDER
```

```
80 PRINT TAB(7)"DEPRECIATION CALCULATIONS"
90 PRINT TAB(7)"************************"
100 PRINT:PRINT
110 PRINT TAB(5)"THERE ARE TWO COMMON METHODS OF"
120 PRINT TAB(5)"CALCULATING THE DEPRECIATION OF"
130 PRINT TAB(5)"THE VALUE OF AN ASSET OVER A"
140 PRINT TAB(5)"GIVEN PERIOD. THESE ARE:"
150 PRINT
160 PRINT TAB(5)"1. THE STRAIGHT LINE METHOD."
170 PRINT
180 PRINT TAB(5)"2. THE DIMINISHING BALANCE"
190 PRINT TAB(8)"METHOD."
200 PRINT:PRINT
210 PRINT TAB(5)"THIS PROGRAM SHOWS DEPRECIATION"
220 PRINT TAB(5)"OVER A FIXED PERIOD USING BOTH"
230 PRINT TAB(5)"METHODS."
240 PRINT:PRINT
250 PRINT TAB(5)"PRESS ANY KEY TO CONTINUE"
260 GETKEY A$
270 REM************************
280 REM************************
290 SCNCLR
300 PRINT:PRINT
310 INPUT"VALUE OF ASSET ON ACQUISITION";IV
320 PRINT:PRINT
330 PRINT"IF YOU WROTE OFF A FIXED AMOUNT"
340 INPUT"EACH YEAR, HOW MUCH WOULD IT BE";A
350 PRINT
360 PRINT:PRINT"IF YOU WROTE OFF A PERCENTAGE"
370 INPUT"EACH YEAR, WHAT WOULD IT BE";PR
380 PRINT:PRINT
390 PRINT"NUMBER OF YEARS TO BE PRESENTED"
400 INPUT"(1 TO 20)";YR%
410 REM NO TRAPS ON ENTRIES
420 REM SILLY ENTRIES WILL GIVE SILLY RESULTS
430 PRINT:PRINT
440 PRINT "PRESS ANY KEY TO CONTINUE"
450 GETKEY A$
460 REM************************
470 REM************************
480 SCNCLR
490 PRINT"YEAR","STR. LINE",,"DIM. BAL."
500 PRINT,"VALUE",,"VALUE"
510 PRINT
520 VA=IV
530 FOR N=1 TO YR%
540 VL=IV-A*N
```

```
550 REM VALUE=INITIAL VALUE - (FIXED AMT, * NO. OF YEARS)
560 IF VL<0 THEN VL=0
570 VA=VA*(100-PR)/100
580 IF VA<0 THEN VA=0
590 REM REDUCE VALUE BY A PERCENTAGE EACH YEAR
600 DEF FNR(X)=(INT(X*100+.5))/100
610 REM FIGURES CORRECT TO 2 DECIMAL PLACES AFTER POINT
620 PRINTN,FNR(VL),,FNR(VA)
630 NEXT
640 GETKEY A$:REM ANY KEY ENDS PROGRAM
650 END
```

STOCK CONTROL SYSTEM

The next four programs form a rudimentary stock control system. As they are written they use the tape cassette for information storage, but they could easily be amended to use floppy disks instead. If you intend using your computer extensively for file handling of this type you would be well advised to consider floppy disks.

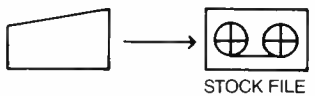
The first program lets you set up the stock file initially. In its present form it allows only ten different types of stock item. We are sure the reader will be able to amend the program to increase this number if necessary.

The next program is used to record all transactions, both additions to and withdrawals from stock. At present the program does not verify the data as it is entered. This would be a useful extension.

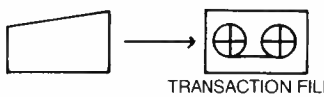
The third program updates the stock file with the information held in the transaction file. This is a fairly complicated program, which we have tried to make self explanatory by the use of PRINT and REM statements. This program also produces a list of items to be reordered.

The final program allows the full stock file to be printed out. This could be done after the stock file has been created or after it has been updated.

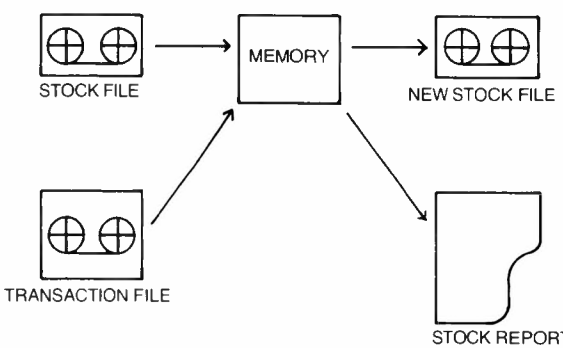
The full stock control system is:



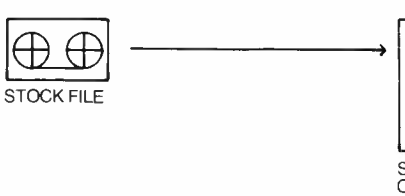
PROGRAM P16 STOCK FILE CREATION
USER KEYS DATA
INTO MEMORY OR PROGRAM
SAVES TO TAPE



PROGRAM P17 TRANSACTION FILE CREATION



PROGRAM P18



PROGRAM P19

P16 Stock file creation

This program creates the stock file.

COMMANDS

Key in the program and RUN.
Have some blank tapes ready.

```
10 REM STOCK FILE CREATION
20 REM ******************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,3:REM BLUE INK
60 COLOR 4,10,5:REM BROWN BORDER
70 SCNCLR
80 CHAR,10,2,"STOCK FILE CREATION"
90 CHAR,10,3,"==================="
100 CHAR,4,6,"THIS PROGRAM SETS UP A STOCK FILE"
110 CHAR,4,7,"ON TAPE. THE STRUCTURE OF EACH"
120 CHAR,4,8,"STOCK LINE IS:"
130 CHAR,9,10,"STOCK NUMBER (1 TO 10)"
140 CHAR,9,11,"DESCRIPTION (UP TO 30 CHRS)"
150 CHAR,9,12,"NUMBER IN STOCK"
160 CHAR,9,13,"REORDER LEVEL"
170 CHAR,9,14,"REORDER QUANTITY"
180 CHAR,4,16,"PLEASE ENTER DATA WHEN PROMPTED"
190 CHAR,5,19,CHR$(18)+" PRESS ANY KEY TO CONTINUE "
200 PRINTCHR$ (146)
210 GETKEY A$
220 REM**************************
230 DIM N%(9,2):DIM J%(9,2)
240 FOR K=0 TO 9
250 :  SCNCLR
260 :  CHAR,12,2,"STOCK NUMBER =":PRINT K+1
270 :  CHAR,9,5,"DESCRIPTION"
280 :  DR$="NONE"
290 :  REM EMPTY STRING ELEMENTS PREVENT VERIFICATION
300 :  CHAR,9,6,"":INPUT DR$
310 :  IF LEN(DR$)>30 THEN DR$=LEFT$(DR$,30)
320 :  D$(K)=DR$
330 PRINT
```

```
340 :   PRINT TAB(9):INPUT"NUMBER IN STOCK";N%(K,0)
350 PRINT
360 :   PRINT TAB(9):INPUT"REORDER LEVEL";N%(K,1)
370 PRINT
380 :   PRINT TAB(9):INPUT"REORDER QUANTITY";N%(K,2)
390 NEXT
400 REM*************************
410 SCNCLR
420 CHAR,4,2,"ENSURE THERE IS A TAPE IN THE"
430 CHAR,4,3,"CASSETTE RECORDER READY FOR SAVING"
440 CHAR,4,6,"WHAT IS THE NAME OF THE FILE"
450 CHAR,4,7,"":INPUT N$
460 REM*************************
470 DO:REM FILE SAVE STARTS HERE
480 :   SCNCLR:CHAR,4,4,""
490 :   OPEN 1,1,1,N$
500 :   FOR K=0 TO 9
510 :      PRINT# 1,D$(K)
520 :      FOR J=0 TO 2:PRINT# 1,N%(K,J):NEXT
530 :   NEXT
540 :   CLOSE 1
550 :   REM*************************
560 :   SCNCLR
570 :   CHAR,4,4,"REWIND CASSETTE RECORDER TO START"
580 :   CHAR,4,5,"OF FILE"
590 :   CHAR,4,8,"PRESS ANY KEY WHEN READY"
600 :   GETKEY A$
610 :   CHAR,4,10,""
620 :   OPEN 1,1,0,N$
630 :   K=0
640 :   DO WHILE ST=0 AND K<10
650 :      INPUT# 1,E$(K)
660 :      FOR N=0 TO 2:INPUT# 1,J%(K,N):NEXT
670 :      K=K+1
680 :   LOOP
690 :   CLOSE 1
700 :   REM*************************
710 :   SCNCLR
720 :   F=1
730 :   FOR K=0 TO 9
740 :      IF D$(K)<>E$(K) THEN K=9:F=2
750 :      R=0
760 :      DO WHILE F=1 AND R<3
770 :         IF N%(K,R)<>J%(K,R) THEN K=9:R=3:F=2
780 :         R=R+1
790 :      LOOP
800 :   NEXT
```

```
810 :  ON F GOSUB 920,990
820 LOOP UNTIL F=1
830 END
840 REM*************************
850 REM*************************
860 REM
870 REM       SUBROUTINES
880 REM
890 REM*************************
900 REM*************************
910 REM VERIFICATION CHECK PASSED
920 CHAR,7,7,CHR$(18)+" VERIFICATION CHECK PASSED "
930 PRINT CHR$(146)
940 CHAR,9,9,"STOP CASSETTE RECORDER"
950 RETURN
960 REM*****************************
970 REM*****************************
980 REM VERIFICATION CHECK FAILED
990 CHAR,7,7,CHR$(18)+" VERIFICATION CHECK FAILED "
1000 PRINT CHR$(146)
1010 CHAR,5,10,"REWIND TAPE TO INITIAL POSITION"
1020 CHAR,8,12,"PRESS ANY KEY WHEN READY"
1030 GETKEY A$
1040 RETURN
1050 REM*****************************
1060 REM*****************************
```

P17 Transaction file creation

This program allows you to record up to 100 transactions.

COMMANDS

Key in the program and RUN.
Have a blank tape ready.

```
10 REM TRANSACTION FILE CREATION
20 REM ************************
30 REM
40 COLOR 0,2:REM WHITE PAPER
50 COLOR 1,7,3:REM BLUE INK
60 COLOR 4,10,4:REM BROWN BORDER
70 SCNCLR
80 CHAR,8,2,"TRANSACTION FILE CREATION"
90 CHAR,8,3,"========================="
100 CHAR,4,5,"THIS PROGRAM ALLOWS THE USER TO"
110 CHAR,4,6,"RECORD A SET OF TRANSACTIONS"
120 CHAR,4,7,"AGAINST A STOCK FILE."
130 CHAR,4,9,"THE TRANSACTION FILE STRUCTURE IS"
140 CHAR,4,10,"AS FOLLOWS:"
150 CHAR,9,12,"STOCK NUMBER"
160 CHAR,9,13,"CODE [1 - WITHDRAWAL"
170 CHAR,9,14,"     [2 - ADDITION"
180 CHAR,9,15,"QUANTITY"
190 CHAR,4,17,"UP TO 100 ITEMS CAN BE HANDLED."
200 CHAR,4,19,"ENTER DETAILS WHEN PROMPTED."
210 CHAR,4,20,"ENTER A NEGATIVE STOCK NUMBER TO
220 CHAR,4,21,"FINISH."
230 CHAR,5,23,CHR$(18)+" PRESS ANY KEY TO CONTINUE "
240 PRINT CHR$(146)
250 GETKEY A$
260 REM*************************
270 DIM T%(99,2):DIM W%(99,2)
280 N=0:F=0
290 DO WHILE N<100 AND F=0
300 :  SCNCLR:PRINT
310 :  DO
320 :     PRINT
330 :     PRINT TAB(9);
```

```
340 :      INPUT"STOCK NUMBER(1 TO 10)";T%(N,0)
350 :   LOOP WHILE T%(N,0)=0 OR T%(N,0)>10
360 :   IF T%(N,0)<0 THEN F=1
370 :   R=0
380 :   DO UNTIL F=1 OR R=1
390 :      REM****************
400 :      DO
410 :         PRINT
420 :         PRINT TAB(9);
430 :         INPUT"CODE (1 OR 2)";T%(N,1)
440 :      LOOP UNTIL T%(N,1)=1 OR T%(N,1)=2
450 :      REM****************
460 :      DO
470 :         PRINT
480 :         PRINT TAB(9);
490 :         INPUT"QUANTITY";T%(N,2)
500 :      LOOP UNTIL T%(N,2)>0
510 :      REM****************
520 :      R=1
530 :   LOOP
540 :   N=N+1
550 LOOP
560 REM*************************
570 SCNCLR
580 CHAR,4,2,"ENSURE TAPE IS READY THEN PRESS"
590 CHAR,4,3,"ANY KEY."
600 GETKEY A$
610 CHAR,4,5,"WHAT IS THE NAME OF THE FILE?"
620 CHAR,4,6,"":INPUT N$
630 R$=CHR$(13):REM RETURN CHARACTER
640 REM*************************
650 DO:REM FILE SAVE STARTS HERE
660 :   SCNCLR
670 :   CHAR,4,4,""
680 :   OPEN 1,1,1,N$
690 :   FOR K=0 TO 99
700 :      PRINT# 1,T%(K,0) R$ T%(K,1) R$ T%(K,2)
710 :   IF T%(K,0)<1 THEN K=99
720 :   NEXT
730 :   CLOSE 1
740 :   REM*************************
750 :   SCNCLR
760 :   CHAR,4,3,"REWIND CASSETTE RECORDER TO START"
770 :   CHAR,4,4,"OF FILE."
780 :   CHAR,4,7,"PRESS ANY KEY WHEN READY."
790 :   GETKEY A$
800 :   REM*************************
```

```
810 :  CHAR,4,10,""
820 :  OPEN 1,1,0,N$
830 :  K=0
840 :  DO UNTIL K>99 OR ST<>0
850 :     N=0
860 :     DO UNTIL N>2 OR ST<>0
870 :        INPUT# 1,W%(K,N)
880 :        N=N+1
890 :     LOOP
900 :     IF W%(K,0)<1 THEN K=99
910 :     K=K+1
920 :  LOOP
930 :  CLOSE 1
940 :  REM*************************
950 :  SCNCLR
960 :  CHAR,15,10,CHR$(18)+"VERIFYING"+CHR$(146)
970 :  F=1
980 :  FOR K=0 TO 99
990 :     FOR N=0 TO 2
1000 :        IF T%(K,N)<>W%(K,N) THEN K=99:F=2:N=2
1010 :     NEXT
1020 :     IF T%(K,0)<0 THEN K=99
1030 :  NEXT
1040 :  SCNCLR
1050 :  ON F GOSUB 1160,1220
1060 LOOP UNTIL F=1
1070 END
1080 REM*************************
1090 REM*************************
1100 REM
1110 REM      SUBROUTINES
1120 REM
1130 REM*************************
1140 REM*************************
1150 REM VERIFICATION CHECK PASSED
1160 CHAR,7,8,CHR$(18)+" VERIFICATION CHECK PASSED "
1170 PRINT CHR$(146)
1180 CHAR,9,11,"STOP CASSETTE RECORDER"
1190 RETURN
1200 REM*************************
1210 REM*************************
1220 CHAR,7,8,CHR$(18)+" VERIFICATION CHECK FAILED "
1230 PRINT CHR$(146)
1240 CHAR,4,11,"REWIND TAPE TO INITIAL POSITION"
1250 CHAR,8,13,"PRESS ANY KEY WHEN READY"
1260 GETKEY A$
1270 RETURN
```

P18 Stock file update and report

This program updates the stock file and produces a list of items to be ordered.

The program can output to the screen or to a printer. It will work both with printers which have an automatic line feed on carriage return (such as the Commodore MPS801) and those which do not.

COMMANDS

Key in the program and RUN.
Have your stock file tape, your transaction file tape and a blank tape handy.

```
10 REM STOCK FILE UPDATE AND REPORT
20 REM ****************************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,2:REM BLUE INK
60 COLOR 4,10,4:REM BROWN BORDER
70 SCNCLR
80 PRINT:PRINT
90 PRINT TAB(6)"STOCK FILE UPDATE AND REPORT"
100 PRINT TAB(6)"============================"
110 PRINT:PRINT
120 PRINT TAB(4)"THIS PROGRAM UPDATES A STOCK FILE"
130 PRINT TAB(4)"OF TEN ITEMS AND THEN SAVES IT"
140 PRINT TAB(4)"BACK TO TAPE. THE PROGRAM ALSO"
150 PRINT TAB(4)"PRODUCES A REPORT OF ALL ITEMS TO"
160 PRINT TAB(4)"BE REORDERED."
170 PRINT:PRINT:PRINT
180 PRINT TAB(6)CHR$(18)" PRESS ANY KEY TO CONTINUE "
190 GETKEY A$
200 REM****************************
210 SCNCLR:PRINT:PRINT
220 DIM N%(9,2)
230 DIM J%(9,2)
240 DIM T%(99,2)
250 PRINT TAB(4)"WHAT IS THE NAME OF THE STOCK"
260 PRINT TAB(4):INPUT"FILE";N$
```

```
270 PRINT:PRINT
280 PRINT TAB(4)"POSITION TAPE AT START OF STOCK"
290 PRINT TAB(4)"FILE, THEN PRESS ANY KEY."
300 GETKEY A$
310 REM*****************************
320 REM LOAD STOCK FILE
330 SCNCLR:PRINT:PRINT
340 OPEN 1,1,0,N$
350 K=0
360 DO UNTIL K>9 OR ST<>0
370 :  INPUT# 1,D$(K)
380 :  J=0
390 :  DO UNTIL J>2 OR ST<>0
400 :     INPUT# 1,N%(K,J)
410 :     J=J+1
420 :  LOOP
430 :  K=K+1
440 LOOP
450 CLOSE 1
460 REM*****************************
470 SCNCLR:PRINT:PRINT
480 PRINT TAB(8)CHR$(18)" STOP CASSETTE RECORDER "
490 PRINT:PRINT
500 PRINT TAB(6)"WHAT IS THE NAME OF THE"
510 PRINT TAB(6):INPUT"TRANSACTION FILE";N$
520 PRINT:PRINT
530 PRINT TAB(6)"POSITION TAPE AT START OF"
540 PRINT TAB(6)"TRANSACTION FILE, THEN PRESS"
550 PRINT TAB(6)"ANY KEY."
560 GETKEY A$
570 REM*****************************
580 REM LOAD TRANSACTION FILE
590 SCNCLR:PRINT:PRINT
600 OPEN 1,1,0,N$
610 K=0
620 DO UNTIL K>99 OR ST<>0
630 :  J=0
640 :  DO UNTIL J>2 OR ST<>0
650 :     INPUT# 1,T%(K,J)
660 :     J=J+1
670 :  LOOP
680 :  K=K+1
690 LOOP
700 CLOSE 1
710 REM*****************************
720 SCNCLR:PRINT:PRINT
730 PRINT TAB(8)CHR$(18)" STOP CASSETTE RECORDER "
```

```
740 PRINT:PRINT
750 PRINT TAB(11)"UPDATING COMMENCING"
760 J=0
770 DO UNTIL J>99
780 :  K=T%(J,0)-1
790 :  LP=0
800 :  DO WHILE K>=0 AND LP=0
810 :     IF T%(J,1)=1 THEN N%(K,0)=N%(K,0)-T%(J,2)
820 :     IF T%(J,1)=2 THEN N%(K,0)=N%(K,0)+T%(J,2)
830 :     PRINT".";
840 :     LP=1
850 :  LOOP
860 :  IF K<0 THEN J=99
870 :  J=J+1
880 LOOP
890 PRINT:PRINT
900 PRINT TAB(12)"UPDATING COMPLETE"
910 REM*****************************
920 PRINT:PRINT
930 PRINT TAB(4)"WHAT IS THE NAME OF THE NEW STOCK"
940 PRINT TAB(4):INPUT"FILE";N$
950 SCNCLR:PRINT:PRINT
960 PRINT TAB(4)"WIND TAPE TO WHERE YOU WISH TO"
970 PRINT TAB(4)"SAVE THE NEW STOCK FILE, THEN"
980 PRINT TAB(4)"PRESS ANY KEY."
990 GETKEY A$
1000 REM*****************************
1010 REM SAVE NEW STOCK FILE
1020 DO
1030 :  SCNCLR:PRINT:PRINT TAB(4)
1040 :  R$=CHR$(13):REM RETURN KEY
1050 :  OPEN 1,1,1,N$
1060 :  FOR K=0 TO 9
1070 :     PRINT# 1,D$(K) R$ N%(K,0) R$ N%(K,1) R$ N%(K,2)
1080 :  NEXT
1090 :  CLOSE 1
1100 :  REM*************************
1110 :  SCNCLR:PRINT:PRINT
1120 :  PRINT TAB(8)CHR$(18)" STOP CASSETTE RECORDER "
1130 :  PRINT:PRINT
1140 :  PRINT TAB(4)"REWIND THE TAPE TO THE START OF"
1150 :  PRINT TAB(4)"THE NEW STOCK FILE THEN PRESS"
1160 :  PRINT TAB(4)"ANY KEY."
1170 :  GETKEY A$
1180 :  REM***********************
1190 :  REM LOAD NEW STOCK FILE BACK IN
1200 :  SCNCLR:PRINT:PRINT
```

```
1210 :  OPEN 1,1,0,N$
1220 :  K=0
1230 :  DO UNTIL K>9 OR ST<>0
1240 :  INPUT# 1,E$(K)
1250 :     N=0
1260 :     DO UNTIL N>2 OR ST<>0
1270 :        INPUT# 1,J%(K,N)
1280 :        N=N+1
1290 :     LOOP
1300 :     K=K+1
1310 :  LOOP
1320 :  CLOSE 1
1330 :  REM************************
1340 :  REM VERIFY NEW STOCK FILE
1350 :  SCNCLR:PRINT:PRINT
1360 :  PRINT TAB(8)CHR$(18)" STOP CASSETTE RECORDER "
1370 :  F=2
1380 :  FOR K=0 TO 9
1390 :     IF D$(K)<>E$(K) THEN F=1:K=9
1400 :     FOR R=0 TO 2
1410 :        IF N%(K,R)<>J%(K,R) THEN F=1:K=9:R=2
1420 :     NEXT
1430 :  NEXT
1440 :  PRINT:PRINT
1450 :  REM************************
1460 :  ON F GOSUB 2640,2740
1470 :  IF F=2 THEN GOSUB 1640
1480 LOOP UNTIL F=2
1490 END
1500 REM*****************************
1510 REM*****************************
1520 REM
1530 REM         ***************
1540 REM         *             *
1550 REM         *             *
1560 REM         * SUBROUTINES *
1570 REM         *             *
1580 REM         *             *
1590 REM         ***************
1600 REM
1610 REM*****************************
1620 REM*****************************
1630 REM SELECT OUTPUT DEVICE
1640 SCNCLR
1650 PRINT:PRINT
1660 PRINT TAB(4)"SELECT THE OUTPUT DEVICE FOR"
1670 PRINT TAB(4)"THE REORDER REPORT BY PRESSING:"
```

```
1680 PRINT
1690 PRINT TAB(10)CHR$(18)" KEY 1 "CHR$(146)" - SCREEN"
1700 PRINT
1710 PRINT TAB(10)CHR$(18)" KEY 2 "CHR$(146)" - PRINTER"
1720 DO:GET A$:LOOP UNTIL A$="1" OR A$="2"
1730 A=VAL(A$)
1740 ON A GOSUB 1790,2200
1750 RETURN
1760 REM*****************************
1770 REM*****************************
1780 REM SCREEN OUTPUT
1790 K=0
1800 DO UNTIL K>9
1810 :  R=0
1820 :  LP=0
1830 :  IF N%(K,0)>N%(K,1) THEN R=1:ELSE R=0
1840 :  DO WHILE R=0 AND LP=0
1850 :     SCNCLR:PRINT:PRINT
1860 :     PRINT TAB(8)CHR$(18);
1870 :     PRINT" ITEMS TO BE REORDERED "
1880 :     PRINT:PRINT
1890 :     PRINT TAB(14)"STOCK ITEM";K+1
1900 :     PRINT:PRINT
1910 :     R%=(40-LEN(D$(K)))/2
1920 :     PRINT TAB(R%)D$(K)
1930 :     PRINT:PRINT
1940 :     PRINT TAB(10)"NUMBER IN STOCK ";N%(K,0)
1950 :     PRINT
1960 :     PRINT TAB(10)"REORDER LEVEL   ";N%(K,1)
1970 :     PRINT
1980 :     PRINT TAB(10)"REORDER QUANTITY";N%(K,2)
1990 :     PRINT:PRINT
2000 :     PRINT TAB(4)"NEGATIVE NUMBERS IN STOCK MEAN"
2010 :     PRINT TAB(4)"CUSTOMERS WAITING FOR GOODS."
2020 :     PRINT:PRINT
2030 :     PRINT TAB(4)CHR$(18);
2040 :     PRINT" PRESS ANY KEY FOR NEXT ITEM "
2050 :     GETKEY A$
2060 :     LP=1
2070 :  LOOP
2080 :  K=K+1
2090 LOOP
2100 SCNCLR:PRINT:PRINT
2110 PRINT TAB(7)"NO FURTHER ITEMS ARE TO BE"
2120 PRINT TAB(7)"REORDERED."
2130 PRINT:PRINT
2140 PRINT TAB(12)"HAVE A NICE DAY."
```

```
2150 PRINT:PRINT
2160 PRINT TAB(17)CHR$(18)" BYE! "
2170 RETURN
2180 REM*****************************
2190 REM*****************************
2200 SCNCLR:PRINT:PRINT
2210 PRINT TAB(4)"DOES YOUR PRINTER HAVE AUTOMATIC"
2220 PRINT TAB(4)"LINE FEED ON CARRIAGE RETURN";
2230 Y$="N"
2240 INPUT Y$
2250 IF ASC(Y$)=89 THEN FLN=3:ELSE FLN=131
2260 REM*****************************
2270 OPEN FLN,4
2280 PRINT# FLN,SPC(8)"ITEMS TO BE REORDERED"
2290 K=0
2300 DO UNTIL K>9
2310 :  LP=0
2320 :  IF N%(K,0)>N%(K,1) THEN R=1:ELSE R=0
2330 :  DO WHILE R=0 AND LP=0
2340 :     PRINT# FLN,:PRINT# FLN,
2350 :     PRINT# FLN,SPC(14)"STOCK ITEM";K+1
2360 :     PRINT# FLN,
2370 :     R%=(40-LEN(D$(K)))/2
2380 :     PRINT# FLN,SPC(R%)D$(K)
2390 :     PRINT# FLN,
2400 :     PRINT# FLN,SPC(10)"NUMBER IN STOCK ";N%(K,0)
2410 :     PRINT# FLN,SPC(10)"REORDER LEVEL   ";N%(K,1)
2420 :     PRINT# FLN,SPC(10)"REORDER QUANTITY";N%(K,2)
2430 :     PRINT# FLN,
2440 :     PRINT# FLN,
2450 :     LP=1
2460 :  LOOP
2470 :  K=K+1
2480 LOOP
2490 PRINT# FLN,
2500 PRINT# FLN,
2510 PRINT# FLN,
2520 PRINT# FLN,SPC(4)"NEGATIVE NUMBERS IN STOCK MEAN"
2530 PRINT# FLN,SPC(4)"CUSTOMERS WAITING FOR GOODS."
2540 PRINT# FLN,
2550 PRINT# FLN,
2560 PRINT# FLN,SPC(12)"HAVE A NICE DAY."
2570 PRINT# FLN,
2580 PRINT# FLN,SPC(18)"BYE!"
2590 CLOSE FLN
2600 RETURN
2610 REM*****************************
```

```
2620 REM**************************
2630 REM VERIFICATION CHECK FAILED
2640 PRINT TAB(8)"VERIFICATION CHECK FAILED"
2650 PRINT:PRINT
2660 PRINT TAB(4)"REWIND TAPE TO THE START OF THE"
2670 PRINT TAB(4)"NEW STOCK FILE THEN PRESS ANY"
2680 PRINT TAB(4)"KEY."
2690 GETKEY A$
2700 RETURN
2710 REM**************************
2720 REM**************************
2730 REM VERIFICATION CHECK PASSED
2740 PRINT TAB(8)"VERIFICATION CHECK PASSED
2750 PRINT:PRINT
2760 PRINT TAB(7)CHR$(18)" PRESS ANY KEY TO CONTINUE "
2770 GETKEY A$
2780 RETURN
2790 REM**************************
2800 REM**************************
```

P19 Stock file output

The program outputs a stock file, which has been saved on to tape, to the screen or to a printer. It will work both with printers which have an automatic line feed on carriage return (such as the Commodore MPS801) and those which do not.

COMMANDS

Key in the program and RUN.
Have your stock file tape handy.

```
10 REM STOCK FILE OUTPUT
20 REM *****************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,1:REM BLUE INK
60 COLOR 4,10,4:REM BROWN BORDER
70 SCNCLR
80 PRINT
90 PRINT
100 PRINT TAB(11)"STOCK FILE OUTPUT"
110 PRINT TAB(11)"================="
120 PRINT:PRINT
130 PRINT TAB(4)"THIS PROGRAM READS A STOCK FILE"
140 PRINT TAB(4)"OF TEN ITEMS AND PRINTS IT ON"
150 PRINT TAB(4)"TO THE SCREEN OR TO A PRINTER."
160 PRINT:PRINT:PRINT
170 PRINT TAB(6)CHR$(18)" PRESS ANY KEY TO CONTINUE "
180 GETKEY A$
190 REM*****************************
200 SCNCLR:PRINT:PRINT
210 DIM N%(9,2)
220 PRINT TAB(4)"WHAT IS THE NAME OF THE STOCK"
230 PRINT TAB(4):INPUT"FILE";N$
240 PRINT:PRINT
250 PRINT TAB(4)"POSITION TAPE AT START OF STOCK"
260 PRINT TAB(4)"FILE, THEN PRESS ANY KEY."
270 GETKEY A$
280 REM*****************************
290 REM READ STOCK FILE
300 SCNCLR:PRINT:PRINT
```

```
310 OPEN 1,1,0,N$
320 K=0
330 DO WHILE K<10 AND ST=0
340 :  INPUT# 1,D$(K)
350 :  J=0
360 :  DO WHILE J<3 AND ST=0
370 :     INPUT# 1,N%(K,J)
380 :     J=J+1
390 :  LOOP
400 :  K=K+1
410 LOOP
420 CLOSE 1
430 REM*****************************
440 REM PRINT OUT
450 SCNCLR:PRINT:PRINT
460 PRINT TAB(7)CHR$(18)" STOP CASSETTE RECORDER "
470 PRINT:PRINT
480 PRINT TAB(4)"SELECT THE OUTPUT DEVICE FOR"
490 PRINT TAB(4)"THE STOCK FILE BY PRESSING:"
500 PRINT
510 PRINT TAB(10)CHR$(18)" KEY 1 "CHR$(146)" - SCREEN"
520 PRINT
530 PRINT TAB(10)CHR$(18)" KEY 2 "CHR$(146)" - PRINTER"
540 DO:GET A$:LOOP UNTIL A$="1" OR A$="2"
550 A=VAL(A$)
560 ON A GOSUB 610,870
570 END
580 REM*****************************
590 REM*****************************
600 REM SCREEN OUTPUT
610 FOR K=0 TO 9
620 :  SCNCLR :PRINT
630 :  PRINT:PRINT
640 :  PRINT TAB(13)CHR$(18)" STOCK ITEM"STR$(K+1)" "
650 :  PRINT:PRINT:PRINT
660 :  R%=(40-LEN(D$(K)))/2
670 :  PRINT TAB(R%)D$(K)
680 :  PRINT:PRINT
690 :  PRINT TAB(10)"NUMBER IN STOCK ";N%(K,0)
700 :  PRINT
710 :  PRINT TAB(10)"REORDER LEVEL   ";N%(K,1)
720 :  PRINT
730 :  PRINT TAB(10)"REORDER QUANTITY";N%(K,2)
740 :  PRINT:PRINT:PRINT TAB(4)CHR$(18);
750 :  PRINT" PRESS ANY KEY FOR NEXT ITEM "
760 :  GETKEY A$
770 NEXT
```

```
780 SCNCLR:PRINT:PRINT
790 PRINT TAB(11)"PRINT OUT COMPLETE"
800 PRINT:PRINT
810 PRINT TAB(12)"HAVE A NICE DAY."
820 PRINT:PRINT
830 PRINT TAB(17)CHR$(18)" BYE! "
840 RETURN
850 REM*****************************
860 REM*****************************
870 Y$="N"
880 SCNCLR:PRINT:PRINT:
890 PRINT TAB(4)"DOES YOUR PRINTER HAVE AUTOMATIC"
900 PRINT TAB(4);
910 INPUT"LINE FEED ON CARRIAGE RETURN";Y$
920 IF ASC(Y$)=89 THEN FLN=3:ELSE FLN=131
930 OPEN FLN,4
940 R%=(29-LEN(N$))/2
950 PRINT# FLN,SPC(R%)"STOCK FILE "N$
960 B$="":REM NO SPACE
970 FOR N=1 TO 11+LEN(N$):B$=B$+"-":NEXT
980 PRINT# FLN,SPC(R%)B$
990 PRINT# FLN,:PRINT# FLN,
1000 FOR K=0 TO 9
1010 :  PRINT# FLN,:PRINT# FLN,
1020 :  PRINT# FLN,SPC(14)"STOCK ITEM";K+1
1030 :  PRINT# FLN,
1040 :  R%=(40-LEN(D$(K)))/2
1050 :  PRINT# FLN,SPC(R%)D$(K)
1060 :  PRINT# FLN,
1070 :  PRINT# FLN,SPC(10)"NUMBER IN STOCK ";N%(K,0)
1080 :  PRINT# FLN,SPC(10)"REORDER LEVEL   ";N%(K,1)
1090 :  PRINT# FLN,SPC(10)"REORDER QUANTITY";N%(K,2)
1100 :  PRINT# FLN,
1110 :  PRINT# FLN,
1120 NEXT
1130 PRINT# FLN
1140 PRINT# FLN
1150 PRINT# FLN
1160 CLOSE FLN
1170 RETURN
1180 REM*****************************
1190 REM*****************************
```

P20 Tax calculator

This program calculates the local or state tax on an item and tells you both the tax and the total cost.

The results of the calculation are rounded to the nearest cent.

If an amount ends in 10, 20, etc cents the last zero is not printed - e.g. a cost of $4.50 is printed 4.5. If this causes concern the solution is to change values from numbers to the equivalent string variables. The final zero may then be added if required. This is left as an exercise for the reader.

COMMANDS

Key in the program and RUN.
Enter data as required.

```
10 REM TAX CALCULATOR
20 REM **************
30 REM
40 SCNCLR
50 COLOR 0,15,1:REM DARK BLUE SCREEN
60 COLOR 1,4:REM CYAN INK
70 COLOR 4,15,1:REM DARK BLUE BORDER
80 PRINT  TAB(10)"$$$$$$$$$$$$$$$$$$"
90 PRINT  TAB(10)"$                $"
100 PRINT TAB(10)"$ TAX CALCULATOR $"
110 PRINT TAB(10)"$                $"
120 PRINT TAB(10)"$$$$$$$$$$$$$$$$$$"
130 PRINT:PRINT
140 PRINT TAB(4)"THIS PROGRAM CAN BE USED TO"
150 PRINT TAB(4)"CALCULATE THE LOCAL OR STATE TAX"
160 PRINT TAB(4)"ON AN ITEM. IN THE UK IT CAN BE"
170 PRINT TAB(4)"USED TO HELP MAKE UP VAT RETURNS."
180 PRINT:PRINT
190 PRINT TAB(4)"WHAT IS THE CURRENT RATE (%)";
200 INPUT TAX
210 TAX=TAX/100
220 REM************************
```

```
230 REM************************
240 DO
250 : SCNCLR
260 :  PRINT:PRINT:PRINT
270 :  DEF FNA(X)=INT(X*100+.5)/100
280 :  REM NUMBERS CORRECT TO 2 PLACES AFTER DECIMAL POINT
290 :  INPUT"COST OF ITEM =";CST
300 :  PRINT:PRINT
310 :  PRINT"TAX          =";FNA(CST*TAX)
320 :  PRINT:PRINT
330 :  PRINT"TOTAL COST   =";FNA(CST*(1+TAX))
340 :  PRINT:PRINT:PRINT
350 :  INPUT"ANOTHER RUN";Y$
360 LOOP UNTIL ASC(Y$)<>89
370 END
```

P21 True rate of interest

This program uses a simplistic approach to calculating the true rate of interest on a loan. It is assumed that repayments are on a monthly basis.

To calculate the true rate of interest we compute the amount of dollar-months that have been borrowed. (One dollar-month is equivalent to borrowing one dollar for one month, or fifty cents for two months.) We now calculate as a percentage the ratio of the total interest paid to the total number of dollar-months. We then multiply this number by twelve.

As no monetary units are entered, this program is not limited to dollars and will work for any currency.

COMMANDS

Key in the program and RUN.
Follow the instructions.

```
10 REM TRUE RATE OF INTEREST
20 REM **********************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,1:REM BLUE INK
60 COLOR 4,10,5:REM BROWN BORDER
70 SCNCLR
80 PRINT:PRINT
90 PRINT TAB(9)"TRUE RATE OF INTEREST"
100 PRINT TAB(9)"======================"
110 PRINT:PRINT
120 PRINT TAB(4)"THIS PROGRAM COMPUTES THE TRUE"
130 PRINT TAB(4)"RATE OF INTEREST ON A LOAN"
140 PRINT TAB(4)"TRANSACTION. THE PROGRAM REQUIRES"
150 PRINT TAB(4)"THE AMOUNT BORROWED, THE ANNUAL"
160 PRINT TAB(4)"INTEREST RATE AND THE NUMBER OF"
170 PRINT TAB(4)"MONTHS OVER WHICH THE LOAN IS"
180 PRINT TAB(4)"TAKEN."
190 PRINT:PRINT:PRINT
200 PRINT TAB(7)CHR$(18)" PRESS ANY KEY TO CONTINUE "
210 GETKEY A$
```

```
220 REM*****************************
230 SCNCLR:PRINT:PRINT
240 PRINT TAB(4):INPUT"AMOUNT BORROWED";AMT
250 PRINT
260 PRINT TAB(4):INPUT"ANNUAL INTEREST RATE (%)";RT
270 PRINT
280 PRINT TAB(4):INPUT"NUMBER OF MONTHS";NM
290 PRINT:PRINT
300 IT=AMT*RT*NM/1200:REM TOTAL INTEREST
310 PB=(AMT+IT)/NM:REM PAYBACK PER MONTH
320 FOR J=0 TO NM-1
330 :  BR=BR+AMT-PB*J
340 NEXT
350 REM BR=TOTAL DOLLAR MONTHS BORROWED
360 TRU=1200*IT/BR
370 TRU=INT(TRU*100+.5)/100
380 PRINT TAB(4)"TRUE RATE OF INTEREST IS";TRU;"%"
390 PRINT TAB(4)"================================"
400 PRINT:PRINT:PRINT
410 END
```

P22 Mail list creation

This program allows you to create a mailing list of up to fifty names and addresses and to save this information on cassette tape.

It could be extended to take more names and addresses and modified to use floppy disk storage.

COMMANDS

Key in the program and RUN.
Ensure you have a blank tape handy.
Follow the instructions.

```
10 REM MAILING LIST CREATION
20 REM ********************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,8,1:REM BLUE INK
60 COLOR 4,10,1:REM BROWN BORDER
70 SCNCLR
80 PRINT:PRINT
90 PRINT TAB(9)"MAILING LIST CREATION"
100 PRINT TAB(9)"====================="
110 PRINT:PRINT
120 PRINT TAB(4)"THIS PROGRAM ALLOWS YOU TO TYPE"
130 PRINT TAB(4)"IN NAMES AND ADDRESSES AND TO"
140 PRINT TAB(4)"SAVE THESE ON TAPE AS A MAILING"
150 PRINT TAB(4)"LIST FILE."
160 PRINT
170 PRINT TAB(4)"YOU ARE ALLOWED 50 NAMES AND"
180 PRINT TAB(4)"ADDRESSES. EACH ADDRESS SHOULD BE"
190 PRINT TAB(4)"LIMITED TO FOUR LINES INCLUDING"
200 PRINT TAB(4)"THE POST CODE."
210 PRINT
220 PRINT TAB(4)"PLEASE ENTER DATA WHEN PROMPTED"
230 PRINT:PRINT
240 PRINT TAB(6)CHR$(18);" PRESS ANY KEY TO CONTINUE "
250 GETKEY A$
260 REM**************************
270 DIM NM$(49):DIM AD$(49,3):DIM AN$(49):DIM AA$(49,3)
```

```
280 FOR K=0 TO 49
290 :  FS=0
300 :  SCNCLR:PRINT:PRINT
310 :  FOR J=0 TO 3:AD$(K,J)="...":NEXT
320 :  REM EMPTY STRINGS PREVENT VERIFICATION
330 :  PRINT TAB(6)CHR$(18)" ENTER END AS NAME TO FINISH "
340 :  PRINT
350 :  DO
360 :     PRINT TAB(4):INPUT"NAME";NM$(K)
370 :  LOOP UNTIL LEN(NM$(K))>0
380 :  REM**********************
390 :  DO WHILE NM$(K)<>"END" AND FS=0
400 :     PRINT TAB(4)"ADDRESS:"
410 :     PRINT TAB(4):INPUT"LINE 1";AD$(K,0)
420 :     PRINT TAB(4):INPUT"LINE 2";AD$(K,1)
430 :     PRINT TAB(4):INPUT"LINE 3";AD$(K,2)
440 :     PRINT TAB(4):INPUT"POST CODE";AD$(K,3)
450 :     FS=1
460 :  LOOP
470 :  IF NM$(K)="END"THEN K=49
480 NEXT
490 REM*************************
500 SCNCLR:PRINT:PRINT
510 PRINT TAB(4)"ENSURE THERE IS A TAPE IN THE"
520 PRINT TAB(4)"CASSETTE RECORDER READY FOR SAVING"
530 PRINT:PRINT
540 PRINT TAB(4)"WHAT IS THE NAME OF THE FILE"
550 PRINT TAB(4):INPUT N$
560 REM*************************
570 DO:REM MAIN PROGRAM LOOP
580 REM*************************
590 PRINT:PRINT TAB(4);:REM FILE SAVE STARTS HERE
600 OPEN 1,1,1,N$
610 R$=CHR$(13):REM RETURN KEY
620 FOR K=0 TO 49
630 :  FS=0
640 :  PRINT# 1,NM$(K)
650 :  DO WHILE NM$(K)<>"END" AND FS=0 AND ST=0
660 :     FOR J=0 TO 3:PRINT# 1,AD$(K,J) R$:NEXT
670 :     FS=1
680 :  LOOP
690 :  IF NM$(K)="END" OR ST<>0 THEN K=49
700 NEXT
710 CLOSE 1
720 REM*************************
730 SCNCLR
740 PRINT:PRINT
```

```
750 PRINT TAB(4)"REWIND CASSETTE RECORDER TO START"
760 PRINT TAB(4)"OF FILE"
770 PRINT:PRINT
780 PRINT TAB(4)"PRESS ANY KEY WHEN READY"
790 GETKEY A$
800 PRINT:PRINT
810 OPEN 1,1,0,N$
820 FOR K=0 TO 49
830 :  FS=0
840 :  INPUT# 1,AN$(K)
850 :  DO WHILE AN$(K)<>"END" AND FS=0 AND ST=0
860 :     FOR J=0 TO 3:INPUT# 1,AA$(K,J):NEXT
870 :     FS=1
880 :  LOOP
890 :  IF AN$(K)="END" OR ST<>0 THEN K=49
900 NEXT
910 CLOSE 1
920 REM**************************
930 SCNCLR:PRINT:PRINT
940 FG=1
950 FOR K=0 TO 49
960 :  FS=0
970 :  DO WHILE NM$(K)<>"END" AND FG=1 AND FS=0
980 :    IF NM$(K)<>AN$(K) THEN K=49:FG=2
990 :    FOR J=0 TO 3
1000 :       IF AD$(K,J)<>AA$(K,J) THEN J=3:K=49:FG=2
1010 :   NEXT
1020 :   FS=1
1030 : LOOP
1040 :  IF NM$(K)="END" THEN K=49
1050 NEXT
1060 ON FG GOSUB 1170,1240
1070 REM****************
1080 LOOP UNTIL FG=1
1090 REM****************
1100 END
1110 REM
1120 REM*********************
1130 REM      SUBROUTINES
1140 REM*********************
1150 REM
1160 REM FILE VERIFIED
1170 PRINT TAB(6)CHR$(18);" VERIFICATION CHECK PASSED "
1180 PRINT:PRINT
1190 PRINT TAB(6)"STOP CASSETTE RECORDER"
1200 RETURN
1210 REM*****************************
```

```
1220 REM*****************************
1230 REM VERIFICATION FAILED
1240 PRINT TAB(6)CHR$(18);" VERIFICATION CHECK FAILED "
1250 PRINT:PRINT
1260 PRINT TAB(4)"REWIND TAPE TO INITIAL POSITION."
1270 PRINT:PRINT
1280 PRINT TAB(4)"PRESS ANY KEY WHEN READY."
1290 GETKEY A$
1300 RETURN
1310 REM*****************************
1320 REM*****************************
```

P23 Mailing list maintenance

This program lets you amend a previously created mailing list and save the new list on cassette tape.

The program could be extended to deal with more than fifty names and addresses and could be modified to use floppy disk storage.

COMMANDS

Key in the program and RUN.
Ensure you have your mailing list tape and a blank tape ready.

```
10 REM MAILING LIST MAINTENANCE
20 REM ************************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,1:REM BLUE INK
60 COLOR 4,10,4:REM BROWN BORDER
70 SCNCLR
80 CHAR,8,2,"MAILING LIST MAINTENANCE"
90 CHAR,8,3,"========================"
100 CHAR,4,6,"THIS PROGRAM LETS YOU ADD NAMES"
110 CHAR,4,7,"AND ADDRESSES TO, OR DELETE NAMES"
120 CHAR,4,8,"AND ADDRESSES FROM A MAILING LIST"
130 CHAR,4,9,"WHICH YOU HAVE PREVIOUSLY CREATED"
140 CHAR,4,10,"AND SAVED ON CASSETTE TAPE."
150 CHAR,4,15,"POSITION THE TAPE AT THE START OF"
160 CHAR,4,16,"THE REQUIRED MAILING LIST FILE,"
170 CHAR,4,17,"THEN PRESS ANY KEY."
180 GETKEY A$
190 REM***************************
200 DIM NM$(49):DIM AD$(49,3):DIM AN$(49):DIM AA$(49,3)
210 SCNCLR
220 CHAR,4,2,"":INPUT"WHAT IS THE FILENAME";N$
230 PRINT:PRINT
240 OPEN 1,1,0,N$
250 FOR K=0 TO 49
260 :  INPUT# 1,NM$(K)
270 :  IF NM$(K)="END" OR ST<>0 THEN F=1:K=49:ELSE F=0
```

```
280 :  IF F=0 THEN FOR J=0 TO 3:INPUT# 1,AD$(K,J):NEXT
290 NEXT
300 CLOSE 1
310 REM**************************
320 SL=4
330 DO
340 :  SCNCLR
350 :  IF SL=4 THEN CHAR,8,2," STOP CASSETTE RECORDER "
360 :  CHAR,4,5,"SELECT THE OPERATION REQUIRED BY"
370 :  CHAR,4,6,"PRESSING:"
380 :  CHAR,6,10,"KEY 1 - ADD ADDRESSES"
390 :  CHAR,6,12,"KEY 2 - REMOVE ADDRESSES"
400 :  CHAR,6,14,"KEY 3 - SAVE NEW MAIL LIST"
410 :  CHAR,6,17,"ANY OTHER KEY ENDS PROGRAM"
420 :  GETKEY A$
430 :  SL=ASC(A$)-48:IF SL<0 THEN SL=4
440 :  ON SL GOSUB 590,1050,1500
450 LOOP UNTIL SL>3
460 END
470 REM**************************
480 REM**************************
490 REM
500 REM        ***************
510 REM        *             *
520 REM        * SUBROUTINES *
530 REM        *             *
540 REM        ***************
550 REM
560 REM**************************
570 REM**************************
580 REM SET POINTER AND CHECK IF LIST FULL
590 T=50
600 FOR K=0 TO 49
610 :  IF NM$(K)="END" THEN T=K:K=49
620 NEXT
630 SCNCLR
640 CHAR,11,2,""
650 IF T=50 THEN PRINT "MAILING LIST FULL":ELSE GOSUB 710
660 GOSUB 2260:REM ANY KEY
670 RETURN
680 REM**************************
690 REM**************************
700 REM ADD TO LIST
710 FOR K=T TO 49
720 :  SCNCLR
730 :  CHAR,8,2,CHR$(18)+" ENTERING NEW INFORMATION "
740 :  PRINT CHR$(146)
```

```
750 :  CHAR,6,6," ENTER END AS NAME TO FINISH "
760 :  NM$(K)=""
770 :  DO
780 :     CHAR,4,9,""
790 :     INPUT"NAME";NM$(K)
800 :  LOOP UNTIL LEN(NM$(K))>0
810 :  F=1
820 :  DO WHILE NM$(K)<>"END" AND F=1
830 :     FOR J=0 TO 3:AD$(K,J)="...":NEXT
840 :     CHAR,4,11,"ADDRESS:"
850 :     PRINT
860 :     PRINT TAB(4):INPUT"LINE 1";AD$(K,0)
870 :     PRINT TAB(4):INPUT"LINE 2";AD$(K,1)
880 :     PRINT TAB(4):INPUT"LINE 3";AD$(K,2)
890 :     PRINT TAB(4):INPUT"POST CODE";AD$(K,3)
900 :     F=0
910 :  LOOP
920 :  IF NM$(K)="END" THEN K=49
930 NEXT
940 REM*************************
950 SCNCLR
960 CHAR,6,2,"THE NEW INFORMATION HAS BEEN"
970 CHAR,6,3,"ADDED TO THE MAILING LIST."
980 CHAR,6,5,"REMEMBER YOU STILL HAVE TO"
990 CHAR,6,6,"SAVE THE NEW LIST IF YOU"
1000 CHAR,6,7,"HAVE FINISHED UPDATING IT."
1010 RETURN
1020 REM*************************
1030 REM*************************
1040 REM REMOVE ADDRESS
1050 DO
1060 :  SCNCLR
1070 :  CHAR,9,2,CHR$(18)+" DELETING INFORMATION "
1080 :  PRINT CHR$(146)
1090 :  CHAR,4,5,"PLEASE ENTER THE EXACT NAME"
1100 :  CHAR,4,6,"CORRESPONDING TO THE ENTRY YOU"
1110 :  CHAR,4,7,"WISH TO DELETE. IF THERE ARE TWO"
1120 :  CHAR,4,8,"ADDRESSES CORRESPONDING TO THE"
1130 :  CHAR,4,9,"SAME NAME THE FIRST OF THESE"
1140 :  CHAR,4,10,"WILL BE DELETED."
1150 :  CHAR,4,14,"":INPUT"NAME";NA$
1160 :  T=50:R=50
1170 :  FOR K=0 TO 49
1180 :     IF NM$(K)=NA$ AND R=50 THEN R=K
1190 :     IF NM$(K)="END" THEN T=K:K=49
1200 :  NEXT
1210 :  SCNCLR
```

```
1220 :   IF R=50 THEN CHAR,13,6,"NAME NOT FOUND"
1230 :   F=1
1240 :   DO WHILE R<50 AND F=1
1250 :      CHAR,16,6,"UPDATING"
1260 :      CHAR,1,8,""
1270 :      FOR K=R TO T-2
1280 :         NM$(K)=NM$(K+1)
1290 :         FOR J=0 TO 3:AD$(K,J)=AD$(K+1,J):NEXT
1300 :         PRINT".";
1310 :      NEXT
1320 :      NM$(T-1)="END"
1330 :      F=0
1340 :   LOOP
1350 :   Y$="N"
1360 :   CHAR,9,14,"":INPUT"ANOTHER DELETION (Y/N)";Y$
1370 LOOP WHILE ASC(Y$)=89
1380 REM*************************
1390 SCNCLR
1400 CHAR,6,2,"THE REQUIRED DELETIONS HAVE"
1410 CHAR,6,3,"BEEN MADE."
1420 CHAR,6,5,"REMEMBER YOU STILL HAVE TO"
1430 CHAR,6,6,"SAVE THE NEW LIST IF YOU"
1440 CHAR,6,7,"HAVE FINISHED UPDATING IT."
1450 GOSUB 2260:REM ANY KEY
1460 RETURN
1470 REM*************************
1480 REM*************************
1490 REM SAVE NEW MAILING LIST
1500 SCNCLR
1510 CHAR,4,2,"POSITION THE CASSETTE TAPE TO"
1520 CHAR,4,3,"WHERE YOU WISH TO SAVE THE NEW"
1530 CHAR,4,4,"FILE."
1540 CHAR,4,7,""
1550 INPUT"WHAT IS THE FILENAME";N$
1560 REM*************************
1570 DO
1580 :   CHAR,4,9,"":REM FILE SAVE STARTS HERE
1590 :   OPEN 1,1,1,N$
1600 :   R$=CHR$(13):REM RETURN KEY
1610 :   FOR K=0 TO 49
1620 :      PRINT# 1,NM$(K)
1630 :      F=1
1640 :      DO WHILE NM$(K)<>"END" AND F=1 AND ST=0
1650 :         FOR J=0 TO 3:PRINT# 1,AD$(K,J):NEXT
1660 :         F=0
1670 :      LOOP
1680 :      IF NM$(K)="END" OR ST<>0 THEN K=49
```

```
1690 :   NEXT
1700 :   CLOSE 1
1710 :   REM*************************
1720 :   SCNCLR
1730 :   CHAR,4,2,"REWIND CASSETTE RECORDER TO START"
1740 :   CHAR,4,3,"OF FILE, THEN PRESS ANY KEY."
1750 :   CHAR,4,7,""
1760 :   GETKEY A$
1770 :   OPEN 1,1,0,N$
1780 :   FOR K=0 TO 49
1790 :      F=1
1800 :      INPUT# 1,AN$(K)
1810 :      DO WHILE AN$(K)<>"END" AND ST=0 AND F=1
1820 :         FOR J=0 TO 3:INPUT# 1,AA$(K,J):NEXT
1830 :         F=0
1840 :      LOOP
1850 :      IF AN$(K)="END" OR ST<>0 THEN K=49
1860 :   NEXT
1870 :   CLOSE 1
1880 :   BAD=2
1890 :   SCNCLR
1900 :   FOR K=0 TO 49
1910 :      F=1
1920 :      IF NM$(K)<>AN$(K) THEN BAD=1
1930 :      DO WHILE NM$(K)<>"END" AND F=1 AND BAD=2
1940 :         FOR J=0 TO 3
1950 :            IF AD$(K,J)<>AA$(K,J) THEN BAD=1
1960 :         NEXT
1970 :         F=0
1980 :      LOOP
1990 :      IF NM$(K)="END" OR BAD=1 THEN K=49
2000 :   NEXT
2010 :   ON BAD GOSUB 2080,2180
2020 LOOP UNTIL BAD=2
2030 GOSUB 2260:REM ANYKEY
2040 RETURN
2050 REM*************************
2060 REM*************************
2070 REM CHECK FAILED
2080 CHAR,6,10,CHR$(18)+" VERIFICATION CHECK FAILED "
2090 PRINT CHR$(146)
2100 CHAR,4,12,"REWIND TAPE TO INITIAL POSITION."
2110 CHAR,4,14,"PRESS ANY KEY WHEN READY."
2120 GETKEY A$
2130 SCNCLR
2140 RETURN
2150 REM*************************
```

```
2160 REM*************************
2170 REM CHECK OK
2180 CHAR,6,10,CHR$(18)+" VERIFICATION CHECK PASSED "
2190 PRINT CHR$(146)
2200 CHAR,9,13,"STOP CASSETTE RECORDER"
2210 PRINT:PRINT
2220 RETURN
2230 REM*************************
2240 REM*************************
2250 REM ANY KEY
2260 PRINT:PRINT:PRINT
2270 PRINT TAB(6)CHR$(18)" PRESS ANY KEY FOR MAIN MENU "
2280 GETKEY B$
2290 DO:GET B$:LOOP UNTIL LEN(B$)=0
2300 REM FLUSH KEYBOARD BUFFER
2310 RETURN
2320 REM*************************
2330 REM*************************
```

P24 Mailing list output

This program lets you print the names and addresses on a mailing list on to labels or to display them on the screen.

A useful additional feature would be a search facility which would allow you to specify that only names starting with (say) a particular letter would be printed out. Another useful feature would be the facility to display a telephone dialling code on the screen when a name is typed in.

If labels are to be printed a printer is required. The program will work both with printers which have an automatic line feed on carriage return (such as the Commodore MPS801) and with those which do not.

COMMANDS

Key in the program and RUN.
Ensure you have your mailing list tape ready.
If necessary, adjust the lines indicated by the REM statements in the code to get the optimum print positioning on your labels.

```
10 REM MAILING LIST OUTPUT
20 REM ******************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,1:REM BLUE INK
60 COLOR 4,10,4:REM BROWN BORDER
70 SCNCLR
80 CHAR,10,2,"MAILING LIST OUTPUT"
90 CHAR,10,3,"=================="
100 CHAR,4,6,"THIS PROGRAM LETS YOU VIEW THE"
110 CHAR,4,7,"NAMES AND ADDRESSES ON A MAILING"
120 CHAR,4,8,"WHICH YOU HAVE PREVIOUSLY CREATED"
130 CHAR,4,9,"AND SAVED ON CASSETTE TAPE."
140 CHAR,4,12,"IT ALSO LETS YOU PRINT THE LIST"
150 CHAR,4,13,"ON TO ADDRESS LABELS."
160 CHAR,4,15,"POSITION THE TAPE AT THE START OF"
170 CHAR,4,16,"THE REQUIRED MAILING LIST FILE,"
180 CHAR,4,17,"THEN PRESS ANY KEY."
```

```
190 GETKEY A$
200 REM**************************
210 DIM NM$(49):DIM AD$(49,3)
220 SCNCLR
230 CHAR,4,2,"":INPUT"WHAT IS THE FILENAME";N$
240 PRINT:PRINT
250 OPEN 1,1,0,N$
260 FOR K=0 TO 49
270 :  INPUT# 1,NM$(K)
280 :  IF NM$(K)="END" OR ST<>0 THEN F=1:K=49:ELSE F=0
290 :  IF F=0 THEN FOR J=0 TO 3:INPUT# 1,AD$(K,J):NEXT
300 NEXT
310 CLOSE 1
320 REM**************************
330 SL=3
340 DO
350 :  SCNCLR
360 :  IF SL=3 THEN CHAR,8,2," STOP CASSETTE RECORDER "
370 :  CHAR,4,6,"SELECT THE OPERATION REQUIRED BY"
380 :  CHAR,4,7,"PRESSING:"
390 :  CHAR,6,10,"KEY 1 - VIEW ADDRESSES"
400 :  CHAR,6,12,"KEY 2 - PRINT LABELS"
410 :  CHAR,6,15,"ANY OTHER KEY ENDS PROGRAM"
420 :  GETKEY A$
430 :  SL=ASC(A$)-48:IF SL<0 THEN SL=3
440 :  ON SL GOSUB 590,790
450 LOOP UNTIL SL>2
460 END
470 REM**************************
480 REM**************************
490 REM
500 REM        ***************
510 REM        *             *
520 REM        * SUBROUTINES *
530 REM        *             *
540 REM        ***************
550 REM
560 REM**************************
570 REM**************************
580 REM VIEW ADDRESSES
590 K=0
600 REM DO LOOP EASIER TO USE THAN FOR NEXT LOOP HERE
610 DO UNTIL K=50 OR NM$(K)="END"
620 :  SCNCLR
630 :  CHAR,10,8,NM$(K)
640 :  FOR J=0 TO 3
650 :     CHAR,10,10+J,AD$(K,J)
```

```
660 :  NEXT
670 :  CHAR,10,16,CHR$(18)+" PRESS ANY KEY "+CHR$(146)
680 :  GETKEY A$
690 :  K=K+1
700 LOOP
710 REM**************************
720 SCNCLR
730 CHAR,10,8,"END OF ADDRESS FILE"
740 GOSUB 1130 REM ANY KEY
750 RETURN
760 REM**************************
770 REM**************************
780 REM PRINT LABELS
790 SCNCLR
800 Y$="N"
810 CHAR,2,8,"DOES YOUR PRINTER HAVE AN AUTOMATIC"
820 CHAR,2,9,"LINE FEED ON CARRIAGE RETURN (Y/N)"
830 CHAR,2,10,"":INPUT Y$
840 IF ASC(Y$)=89 THEN FLN=3:ELSE FLN=131
850 REM SET PRINTER FILE NUMBER
860 REM**************************
870 SCNCLR
880 CHAR,7,6,"ENSURE LABELS ARE CORRECTLY"
890 CHAR,7,7,"ALIGNED AND PRINTER IS ON."
900 CHAR,7,10,CHR$(18)+" PRESS ANY KEY WHEN READY "
910 PRINT CHR$(146)
920 GETKEY A$
930 OPEN FLN,4
940 FOR K=0 TO 3:PRINT# FLN,:NEXT
950 REM ADJUST PREVIOUS LINE TO POSITION PRINT ON LABELS
960 K=0
970 DO UNTIL K>49 OR NM$(K)="END"
980 :  SP=10:PRINT# FLN,SPC(SP)NM$(K)
990 :  REM ADJUST SP TO GET CORRECT MARGIN
1000 :  FOR J=0 TO 3:PRINT# FLN,SPC(SP)AD$(K,J):NEXT
1010 :  FOR J=0 TO 3:PRINT# FLN,:NEXT
1020 :  REM ADJUST PREVIOUS LINE TO KEEP LABELS ALIGNED
1030 :  K=K+1
1040 LOOP
1050 CLOSE FLN
1060 CHAR,13,13,"PRINT COMPLETE"
1070 PRINT:PRINT
1080 GOSUB 1130:REM ANY KEY
1090 RETURN
1100 REM**************************
1110 REM**************************
1120 REM ANY KEY
```

```
1130 PRINT:PRINT:PRINT
1140 PRINT TAB(6)CHR$(18)" PRESS ANY KEY FOR MAIN MENU "
1150 GETKEY B$
1160 DO:GET B$:LOOP UNTIL LEN(B$)=0
1170 REM FLUSH KEYBOARD BUFFER
1180 RETURN
1190 REM**************************
1200 REM**************************
```

P25 Four weekly moving average

When attempting to measure trends in sales data it can be useful to plot the sales data along with a continually updated average of the last four weeks' sales.

This program performs such a task, with the sales data held in the form of data statements. Note that the data are terminated by an imaginary negative sale.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM  - FOUR WEEKLY MOVING AVERAGE
20 PRINT "[CS CD CD CD CD CD CD CD CD CD ]"
30 PRINT "            FOUR WEEKLY MOVING"
40 PRINT:PRINT "                 AVERAGE"
50 A=TI
60 IF TI<A+100 THEN GOTO 60
70 PRINT "[CS ]"
80 PRINT "THIS PROGRAM CAN BE USED TO HELP YOU"
90 PRINT "TO FORECAST SALES BASED ON "
100 PRINT "A 4-WEEK MOVING AVERAGE."
110 PRINT
120 PRINT "THIS CAN BE A FAIRLY USEFUL PROGRAM,"
130 PRINT "BUT BEAR IN MIND THAT IT DOES NOT"
140 PRINT "CONSIDER SEASONAL VARIATIONS."
150 PRINT:PRINT:PRINT
160 PRINT "DATA ARE HELD IN DATA STATEMENTS, AND"
170 PRINT "UP TO 52 WEEKS CAN BE HANDLED."
180 PRINT:PRINT
190 PRINT "IF NECESSARY, STOP THE PROGRAM AND ADD"
200 PRINT "TO OR CHANGE THE DATA."
210 PRINT:PRINT:PRINT
220 DIM SA(53),MA(50)
230 REM SCALE THE DATA
240 NO=0:MX=0
```

```
250 REM LOOP BACK POINT
260 NO=NO+1
270 READ SA(NO)
280 IF SA(NO)<0 THEN GOTO 310
290 IF MX<SA(NO) THEN MX=SA(NO)
300 GOTO 260
310 REM CALCULATE MOVING AVERAGES
320 FOR I=4 TO NO-1
330 MA(I-3)=(SA(I)+SA(I-1)+SA(I-2)+SA(I-3))/4
340 NEXT I
350 GOSUB 450
360 I=1
370 REM LOOP BACK POINT
380 I=I+1:LI=LI+1
390 PRINT I,SA(I),,MA(I)
400 IF LI=18 THEN GOSUB 450
410 IF SA(I)<0 THEN GOTO 430
420 GOTO 380
430 END
440 END
450 PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
460 GET A$:IF A$="" THEN 460
470 PRINT "[CS ]";
480 PRINT"[RON ]            SALES DATA 1984            [ROF
]"
490 PRINT
500 LI=0
510 PRINT "PERIOD    SALES VALUE    MOVING AVERAGE"
520 RETURN
530 DATA 112,224,115,212,118,215,113,214,115,216,112
540 DATA 223,126,224,125,265,145,293,116,216,193,293
550 DATA 187,315,220,354,232,367,198,354,267,365,287
560 DATA 398,254,254,176,234,144,201,101,350,190,483
570 DATA 190,190
580 DATA -9
```

P26 Cost of sales calculation

This program uses three different methods of calculating the cost of sales, and uses the results of these calculations to produce a report on Gross Profit

The methods are:

1. First-In-First-Out
2. Last-In-First-Out
3. The method of Weighted Average

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM COST OF SALES
20 REM THIS PROGRAM COMPUTES COST OF
30 REM SALES USING 3 DIFFERENT METHODS
40 REM
50 REM FIFO, LIFO, WEIGHTED AVERAGE
60 :
70 :
80 REM ENTER PURCHASE DATA
90 :
100 PRINT CHR$(147)
110 PRINT:PRINT:PRINT:PRINT
120 PRINT "PURCHASE DATA"
130 PRINT "-------------"
140 PRINT:PRINT:PRINT:PRINT
150 INPUT "NUMBER OF PURCHASES MADE ";N
160 DIM V(N),P(N):REM DIMENSION VOLUME AND PRICE ARRAYS
170 FOR I=1 TO N
175 PRINT "PURCHASE NUMBER "I
180 INPUT "VOLUME OF PURCHASE ";V(I)
190 INPUT "PRICE OF PURCHASE ";P(I)
195 FOR R=1 TO 3:PRINT CHR$(145);:NEXT R
196 TP=TP+V(I)
200 NEXT I
210 :
220 :
230 REM ENTER SALES DATA
240 PRINT CHR$(147)
```

```
250 PRINT:PRINT:PRINT:PRINT
260 PRINT "SALES  DATA"
270 PRINT "-----------"
280 PRINT:PRINT:PRINT:PRINT
290 INPUT "NUMBER OF SALES MADE";M
300 DIM A(M),S(M):REM DIMENSION VOLUME AND PRICE ARRAYS
305 TT=0:RV=0
310 FOR I=1 TO M
315 PRINT "SALE NUMBER "I
320 INPUT "VOLUME OF SALE ";A(I)
330 INPUT "PRICE OF SALE ";S(I)
331 TT=TT+A(I)
332 RV=RV+A(I)*S(I)
335 FOR R=1 TO 3:PRINT CHR$(145);:NEXT R
340 NEXT I
350 :
355 IF TT>TP THEN  PRINT:PRINT:PRINT:PRINT:PRINT  "OVERSOLD":
END
360 :
370 REM USING THE METHOD OF FIRST IN
380 REM FIRST OUT
390 :
400 T=TT:C=0:I=1
401 PRINT I
410 DO WHILE T>0
420 IF T>V(I) THEN T=T-V(I):C=C+V(I)*P(I)  :ELSE  C=C+T*P(I):
T=0
430 I=I+1
440 LOOP
450 :
460 :
470 REM PRINT OUT RESULTS
480 PRINT CHR$(147):PRINT:PRINT:PRINT
490 PRINT "USING FIFO :"
495 PRINT:PRINT:PRINT
496 PRINT "REVENUE = "RV
500 PRINT "COST OF SALES="C
510 PRINT:PRINT "GROSS PROFIT = "RV-C
520 PRINT:PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
530 GET A$:IF A$="" THEN 530
531 :
532 :
540 REM USING THE METHOD OF LAST IN
550 REM FIRST OUT
560 :
570 T=TT:C=0:I=N
580 DO WHILE T>0
```

```
590 IF T>V(I) THEN T=T-V(I):C=C+V(I)*P(I)  :ELSE  C=C+T*P(I):
T=0
600 I=I-1
610 LOOP
620 :
630 :
640 REM PRINT OUT RESULTS
650 PRINT CHR$(147):PRINT:PRINT:PRINT
660 PRINT "USING LIFO :"
675 PRINT:PRINT:PRINT
686 PRINT "REVENUE = "RV
690 PRINT "COST OF SALES="C
700 PRINT:PRINT "GROSS PROFIT = "RV-C
710 PRINT:PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
720 GET A$:IF A$="" THEN 720
730 :
740 :
750 REM USING THE METHOD OF WEIGHTED
760 REM AVERAGE
770 FOR J=1 TO N
780 PP=PP+V(J)*P(J)
790 NEXT J
800 AV=PP/TP
810 :
820 C=TT*AV
830 :
840 :
850 REM PRINT OUT RESULTS
860 PRINT CHR$(147):PRINT:PRINT:PRINT
870 PRINT "USING WEIGHTED AVERAGE:"
880 PRINT:PRINT:PRINT
890 PRINT "REVENUE = "RV
900 PRINT "COST OF SALES="C
910 PRINT:PRINT "GROSS PROFIT = "RV-C
920 PRINT:PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
930 GET A$:IF A$="" THEN 930
940 :
950 END
```

PRICE LIST SUITE

This is our example of a suite of programs. It should give you an idea of how commercial programs are developed. The programs as presented are disk based, since this is the only feasible way for a commercial set of programs to work. If, however, you have only a tape based system you can still, with care, use these programs. You will have to make the following changes:

Where you see a command of the form

```
DLOAD "filename"
```

replace it with the set of commands

```
PRINT "WIND TAPE TO POSITION xx"
GETKEY A$
LOAD "filename".
```

Position xx is where the file is held on the cassette.

Let's get back to our suite of programs.
These programs are designed to turn the Commodore-16 into a point of sale machine. The suite consists of five programs.

1. PRICELIST MENU : This is the main menu of the system. This program controls which subsidiary program is called in. All programs return to this program when finished.

2. FDUMP: This program can be used to dump the contents of any file to either the screen or the printer. It can be useful outwith this system.

3. MAINTENANCE: This program allows the user to maintain pricing information.

4. PRINT: This is the main report program in the system.

5. POS: This is the point of sale program. It is used to produce the price that the customer pays.

P27 PRICELIST – Main menu program

This is the main control program of the system.

COMMANDS

Please ensure that the whole system is loaded before running this prgram.

Key in program.
Save the program to your backing storage device.
RUN the program.
Select from the menu.

Note - make amendments if you are using tape.

```
10 REM PRICE LIST SUITE
20 REM PROGRAM - MAIN MENU
30 PRINT "[CS RON ]
[ROF ]";
40 PRINT "[RON ]            PRICE LIST MENU             [ROF
]";
50 PRINT "[RON ]                                        [ROF
]"
60 PRINT:PRINT
70 PRINT "[RON ]   1.  CHANGE PRICE LIST                [ROF
]"
80 PRINT
90 PRINT "[RON ]   2.  USE CBM-16 AS POS TERMINAL       [ROF
]"
100 PRINT
110 PRINT "[RON ]   3.  PRINT OUT PRICE LIST             [ROF
]"
120 PRINT "[RON ]   4.  DUMP A DISK FILE TO SCREEN       [ROF
]";
130 PRINT "[RON ]       OR PRINTER                       [ROF
]"
140 PRINT "[RON ]   5.  QUIT THE PROGRAM                 [ROF
]"
150 PRINT "[CD CD ]SELECT NUMBER FOR FUNCTION FROM MENU"
160 GET A$:IF A$<>"" THEN GOTO 160
```

```
170 GET A$:IF A$="" THEN GOTO 170
180 A=VAL(A$)
190 IF A<>1 AND A<>2 AND A<>3 AND A<>4 AND A<>5 THEN GOTO 160
200 PRINT A
210 IF A=1 THEN PRINT "ABOUT TO LOAD PRICE LIST MAINTENANCE"
220 IF A=2 THEN PRINT "ABOUT TO USE CBM AS POS"
230 IF A=3 THEN PRINT "ABOUT TO LOAD PRINT PROGRAM"
240 IF A=4 THEN PRINT "ABOUT TO LOAD DUMP PROGRAM"
250 IF A=5 THEN PRINT "ABOUT TO QUIT PROGRAM"
260 PRINT "OK? ENTER Y OR N"
270 GET A$:IF A$<>"" THEN GOTO 270
280 GET A$:IF A$="" THEN GOTO 280
290 IF A$="N" THEN RUN
300 IF A=1 THEN DLOAD "MAINT"
310 IF A=2 THEN DLOAD "POS"
320 IF A=3 THEN DLOAD "PRINT"
330 IF A=4 THEN DLOAD "FDUMP"
```

P28 MAINT–
Price list maintenance program

This program can be used to:

enter a new price list, or
change the values on a price list.

This program allows the user to keep price lists up to date.

COMMANDS

Key in the program and save to disk or tape.
Call the program from the main menu.
Follow instructions.

```
10 REM PRICE LIST SUITE
20 REM PROGRAM - MAINTENANCE OF PRICE LIST
30 REM
40 PRINT "[CS CD CD RON ]
    [ROF ]";
50 PRINT "[RON ]            PRICE LIST MENU              [ROF
]";
60 PRINT "[RON ]                                         [ROF
]"
70 PRINT:PRINT:PRINT
80 PRINT "[RON ]   1.  ENTER A NEW PRICE LIST            [ROF
]"
90 PRINT
100 PRINT "[RON ]   2.  CHANGE THE VALUES ON A PRICE LIST[ROF
]"
110 PRINT
120 PRINT "[RON ]   3.  GO BACK TO MAIN MENU              [ROF
]"
130 PRINT
140 PRINT "ENTER NUMBER CORRESPONDING TO FUNCTION"
150 GET A$:IF A$="" THEN GOTO 150
160 A=VAL(A$)
170 IF A<>1 AND A<>2 AND A<>3 THEN GOTO 150
180 IF A=1 THEN GOSUB 1000
190 IF A=2 THEN GOSUB 2000
```

```
200 IF A=3 THEN DLOAD "PRICELIST"
210 GOTO 10
990 REM ROUTINE TO SET UP NEW PRICE LIST
1000 PRINT "[CS RON ]                    NEW PRICE LIST
[ROF ]"
1010 PRINT:PRINT:PRINT
1020 PRINT "WHAT IS THE NAME OF THE PRICE LIST":INPUT N$
1030 PRINT "THE FORMAT OF THE PRICE LIST IS AS"
1040 PRINT "FOLLOWS:"
1050 PRINT
1060 PRINT "SUPPLIER NUMBER,DESCRIPTION,PRICE"
1070 PRINT
1080 PRINT "THE LIST SHOULD BE IN ALPHABETIC ORDER"
1090 PRINT "BASED ON THE DESCRIPTION"
1100 PRINT
1110 PRINT "ENTER 0 FOR LAST SUPPLIER NUMBER"
1120 PRINT
1130 OPEN 2,8,2,"0:"+N$+",S,W":T$=""
1140 REM LOOP BACK POINT
1150 INPUT "SUPPLIER NUMBER (0 FOR LAST)";S
1160 INPUT "DESCRIPTION";D$
1170 INPUT "PRICE $";P
1180 IF NOT(D$>T$) THEN PRINT "WRONG ORDER":GOTO 1150
1190 T$=D$
1200 PRINT
1210 PRINT#2,S
1220 PRINT#2,D$
1230 PRINT#2,P
1240 IF S<>0 THEN GOTO 1150
1250 CLOSE 2
1260 RETURN
1990 REM ROUTINE TO CHANGE PRICE LIST
2000 PRINT "[CS RON ]                CHANGE PRICE LIST
[ROF ]"
2010 PRINT:PRINT:PRINT
2020 PRINT "WHAT IS THE NAME OF THE OLD PRICE LIST":INPUT OP$
2030 PRINT "WHAT IS THE NAME OF THE NEW PRICE LIST":INPUT NP$
2040 OPEN 2,8,2,"0:"+OP$+",S,R"
2050 DIM S(100),D$(100),P(100):REM UP TO 100 ITEMS ON PRICE
LIST
2060 REM READ IN THE OLD PRICE LIST
2070 I=0
2080 PRINT "READING IN DATA.";
2090 I=I+1 : REM LOOP BACK POINT
2100 PRINT ".";
2110 INPUT#2,S(I)
2120 INPUT#2,D$(I)
```

```
2130 INPUT#2,P(I)
2140 IF S(I)<>0 AND I<100 THEN GOTO 2090
2150 PRINT
2160 PRINT "THE PRICE LIST HAS NOW BEEN READ IN"
2170 PRINT "PRESS ANY KEY TO CONTINUE"
2180 GET A$:IF A$="" THEN GOTO 2180
2190 PRINT "[CS ]"
2200 FOR J=1 TO I
2210 PRINT "[CS ]DO YOU WISH TO CHANGE THIS ITEM (Y/N)"
2220 PRINT:PRINT:PRINT
2230 PRINT "SUPPLIER NUMBER ";S(J)
2240 PRINT "DESCRIPTION ";D$(J)
2250 PRINT "PRICE ";P(J)
2260 GET A$:IF A$="" THEN GOTO 2260
2270 IF A$="N" THEN GOTO 2540
2280 PRINT:PRINT "ENTER NEW DETAILS"
2290 PRINT "[CU CU CU CU CU CU ]"
2300 PRINT CHR$(28);"SUPPLIER NUMBER  ";
2310 S$=""
2320 GET A$:IF A$="" THEN GOTO 2320
2330 PRINT A$;
2340 IF ASC(A$)=13 THEN 2370
2350 S$=S$+A$
2360 GOTO 2320
2370 S(J)=VAL(S$)
2380 PRINT "DESCRIPTION ";
2390 D$=""
2400 GET A$:IF A$="" THEN GOTO 2400
2410 PRINT A$;
2420 IF ASC(A$)=13 THEN 2450
2430 D$=D$+A$
2440 GOTO 2400
2450 D$(J)=D$
2460 PRINT "PRICE  ";
2470 P$=""
2480 GET A$:IF A$="" THEN GOTO 2480
2490 PRINT A$;
2500 IF ASC(A$)=13 THEN 2530
2510 P$=P$+A$
2520 GOTO 2480
2530 P(J)=VAL(P$)
2535 PRINT CHR$(5);
2540 NEXT J
2550 OPEN 4,8,4,"0:"+NP$+",S,W"
2560 FOR J=1 TO I
2570 PRINT#4,S(I)
2580 PRINT#4,D$(I)
```

```
2590 PRINT#4,P(I)
2600 NEXT J
2610 CLOSE2
2620 CLOSE4
2630 RETURN
```

P29 POS – Point of sale program

This program is used as the point of sale utility in the system. In order to be able to use the program a price list has to be set up beforehand.

The program uses the binary search method of finding an item in the price list.

COMMANDS

Key in program and save to disk.
Ensure that a price list exists on the disk.
Call the program from the main menu.
Follow instructions.

```
10 REM PRICE LIST SUITE
20 REM PROGRAM - POINT OF SALE PROGRAM
30 REM
40 PRINT "[CS ]GOOD MORNING - WELCOME TO THE CBM-16"
50 PRINT "POINT OF SALE PROGRAM."
60 PRINT
70 PRINT"BEFORE STARTING WE HAVE SET UP SOME"
80 PRINT "INITIAL DATA."
90 PRINT
100 INPUT "WHAT IS TODAY'S DATE";DT$
110 INPUT "WHAT IS THE PRICE LIST";PL$
120 INPUT "WHAT IS THE STORE NAME";S$
130 PRINT
140 PRINT "PLEASE ENSURE THAT THE CORRECT PAPER"
150 PRINT "IS IN THE PRINTER, AND THAT IT IS LINED"
160 PRINT "UP CORRECTLY."
170 PRINT
180 PRINT"PLEASE ENSURE THAT YOU HAVE A CLEAN"
190 PRINT "NEW'ED DISK TO STORE YOUR DAILY RECORD"
200 PRINT "ON."
210 PRINT
220 PRINT"IF ALL THESE ACTIONS HAVE BEEN CARRIED"
230 PRINT "OUT. THEN PRESS ANY KEY."
240 GET A$:IF A$="" THEN GOTO 240
250 PRINT "[CS ]THE DATA ENTERED TODAY WILL BE HELD ON"
260 PRINT "DISC WITH THE NAME "DT$
```

```
270 PRINT "[CD ]TO USE THE PROGRAM ENTER THE DESCRIPTION";
280 PRINT "OF THE ITEM, AND THE QUANTITY SOLD"
290 PRINT "[CD ]TO OBTAIN A TOTAL ENTER 'T'"
300 PRINT "A TOTAL SLIP WILL THEN BE PRINTED"
310 PRINT "[CD ]YOU WILL THEN ENTER AMOUNT TENDERED"
320 PRINT "AND THE PROGRAM WILL INFORM YOU OF THE"
330 PRINT "AMOUNT OF CHANGE TO BE RETURNED"
340 PRINT "[CD ]ENTER 'EOD' WHEN THE DAY'S WORK IS DONE"
350 PRINT "[CD ]IF EVERYTHING IS OK THEN PRESS ANY KEY"
360 GET A$:IF A$="" THEN GOTO 360
370 OPEN 2,8,2,"0:"+PL$+",S,R"
380 DIM N(100),D$(100),P(100)
390 I=0
400 I=I+1
410 INPUT#2,N(I)
420 INPUT#2,D$(I)
430 INPUT#2,P(I)
440 IF N(I)<>0 THEN GOTO 400
450 I=I-1
460 CLOSE2
470 PRINT "PRICE LIST NOW IN MEMORY"
480 PRINT "PUT IN NEW DATA DISK, THEN PRESS ANY KEY"
485 OPEN2,8,2,"0:"+DT$+",S,W"
490 GET A$:IF A$="" THEN GOTO 490
500 OPEN1,4:REM OPEN PRINTER
505 REM LOOP BACK POINT
510 PRINT "[CS RON ]          CBM-16  POS PROGRAM
[ROF ]"
520 T=0:A$="":REM FOR NEW CUSTOMER
530 PRINT "[CH CD CD CD CD CD CD ]" :D$="":REM FOR NEXT ITEM
534 PRINT "                                                    "
535 PRINT "[CU ]DESCRIPTION   "+A$+"[CL CL ]";
540 INPUT D$:D$=A$+D$
550 IF D$="T" THEN GOTO 650
560 IF D$="EOD" THEN GOTO 820
570 GOSUB 1000: REM SEARCH PRICE LIST
575 IF F=1 THEN GOTO 635
576 PRINT "                                              ":PRINT"[CU
]";
580 INPUT "QUANTITY";Q
585 PRINT "                                              ":PRINT"[CU
]";
590 PRINT "PRICE ";P
595 PRINT "                                              ":PRINT"[CU
]";
600 PRINT "COST[]";P*Q
610 PRINT#1,D$,Q,P*Q
```

```
620 T=T+P*Q
630 PRINT#2,D$:PRINT#2,Q:PRINT#2,P:PRINT#2,P*Q
635 GET A$:IF A$="" THEN GOTO 635
640 GOTO 530
650 REM TOTAL PRINTING
660 PRINT#1,""
670 PRINT#1,"TOTAL IS ";T
680 PRINT#1,""
690 PRINT#1,"THANK YOU FOR SHOPPING AT "S$
700 PRINT#1,"DATE IS "DT$
710 PRINT#2,T
720 PRINT "[CD CD CD CD CD CD CD CD ]TOTAL IS "T
730 PRINT "ENTER AMOUNT TENDERED"
740 INPUT A
750 IF A<T THEN GOTO 730
760 PRINT "CHANGE IS "A-T
770 GT=GT+T
780 PRINT "PRESS ANY KEY FOR NEXT CUSTOMER"
790 GET A$:IF A$="" THEN GOTO 790
800 GOTO 510
810 REM END OF DAY ROUTINE
820 PRINT "[CS ]"
830 PRINT "TOTAL SALES WERE "GT
840 CLOSE 2
850 PRINT "PLACE MAIN PROGRAM DISK IN DRIVE"
860 PRINT "PRESS ANY KEY WHEN READY"
870 GET A$:IF A$="" THEN GOTO 870
880 DLOAD"PRICELIST"
890 END
950 REM BINARY SEARCH ALGORITHM
960 REM DESCRIPTION IN D$
970 REM DESCRIPTIONS IN D$(I)
980 REM PRICE IN P(I)
990 REM LIST IS ALPHABETICAL
1000 N=I:F=0:D$(0)=" "
1010 M=INT(N/2)
1020 L=M
1025 IF LEFT$(D$,1)=" "THEN D$=MID$(D$,2):GOTO 1025
1030 IF D$=D$(M) THEN P=P(M):RETURN
1040 IF D$<D$(M) THEN L=INT(L/2):M=M-L
1050 IF D$>D$(M) THEN L=INT(L/2):M=M+L
1060 IF M>N THEN L=INT(L/2):M=M-L
1070 IF M>N THEN GOTO 1060
1080 IF L=0 THEN L=2
1090 IF D$<D$(M+1) AND D$>D$(M) THEN PRINT "NOT IN
LIST":F=1:RETURN
1100 IF M=1 AND D$<D$(M) THEN PRINT "NOT IN LIST":F=1:RETURN
```

```
1110 GOTO 1030
1120 F=1
1130 RETURN
```

P30 PRINT–
Price list print program

This is the main report program of the system. This program allows the user to produce a nicely laid out report of the store prices.

COMMANDS

Key in program and save to disk or tape.
Call the program from the main menu.
Follow instructions.

```
10 REM PRICE LIST SUITE
20 REM PROGRAM - PRICE LIST PRINT PROGRAM
30 REM
40       PRINT       "[CS       CD       CD       RON       ]
    [ROF ]";
50 PRINT "[RON ]            PRINTER OPTIONS              [ROF
]";
60 PRINT "[RON ]                                         [ROF
]"
70 PRINT:PRINT:PRINT
80 PRINT "[RON ]   1.  PRINT LIST TO SCREEN              [ROF
]"
90 PRINT
100 PRINT "[RON ]   2.  PRINT LIST TO PRINTER            [ROF
]"
110 PRINT
120 PRINT "[RON ]   3.  PRINT LIST TO BOTH               [ROF
]"
130 PRINT
140 PRINT "[RON ]   4.  GO BACK TO MAIN MENU             [ROF
]"
150 PRINT
160 PRINT "ENTER NUMBER CORRESPONDING TO FUNCTION"
170 GET A$:IF A$="" THEN GOTO 170
180 A=VAL(A$)
190 IF A<>1 AND A<>2 AND A<>3 AND A<>4 THEN GOTO 150
200 IF A=4 THEN DLOAD "PRICELIST"
210 PRINT "[CS ]"
220 PRINT "WHAT IS THE NAME OF THE PRICE LIST":INPUT N$
```

```
230 OPEN2,8,2,"0:"+N$+",S,R"
240 I=0
250 IF A=2 THEN GOTO 300
260   PRINT   "[CS    RON    ]                    PRICE    LIST
[ROF ]"
270 PRINT
280 PRINT "NUMBER","DESCRIPTION","PRICE
290 IF A=1 THEN GOTO 330
300 OPEN 1,4
310 PRINT#1,"                PRICE LIST"
320 PRINT#1,"NUMBER","DESCRIPTION","PRICE"
330 I=I+1
340 INPUT#2,N
350 INPUT#2,D$
360 INPUT#2,P
365 IF N=0 THEN GOTO 470
370 IF A=1 OR A=3 THEN PRINT N,D$,P
380 IF A=2 OR A=3 THEN PRINT#1,N,D$,P
390 IF A=2 THEN GOTO 330
400 IF 20*INT(I/20)<>I THEN GOTO 330
410 PRINT "PRESS ANY KEY FOR NEXT PAGE"
420 GET A$:IF A$="" THEN GOTO 420
430   PRINT   "[CS    RON    ]                    PRICE    LIST
[ROF ]"            440 PRINT
450 PRINT "NUMBER","DESCRIPTION","PRICE
460 GOTO 330
470 CLOSE 1
480 CLOSE 2
490 PRINT:PRINT
500 PRINT "PRESS ANY KEY TO RETURN TO MENU"
510 GET A$:IF A$="" THEN GOTO 510
520 GOTO 10
```

P31 FDUMP – File dump utility

This program can be used to dump a sequential disk file to either the screen or printer. It is very useful to check that a data file has been set up correctly. Note that it cannot be used to dump program files.

We have, however, found it useful on other occasions. If you have a disk then the routine will find a useful place in your library.

The program can be amended to examine tape files as well.

COMMANDS

Key in program and save to disk or tape.
Call the program via MENU program.
Follow instructions.

```
10 REM PRICE LIST SUITE
20 REM FILE DUMP UTILITY
30 REM
40 PRINT "[CS RON ]
[ROF ]";
50 PRINT "[RON ]          FILE DUMP UTILITY              [ROF
]";
60 PRINT "[RON ]                                         [ROF
]"
70 PRINT:PRINT
80 PRINT "THIS PROGRAM CAN BE USED TO OBTAIN A"
90 PRINT "DUMP OF A DISK FILE TO SCREEN OR PRINTER";
100 PRINT "THE OPTIONS AVAILABLE ARE:"
110 PRINT:PRINT
120 PRINT "1. DUMP A FILE TO SCREEN"
130 PRINT "2. DUMP A FILE TO PRINTER"
140 PRINT "3. RETURN TO MAIN MENU"
150 PRINT
160 PRINT "ENTER NUMBER CORRESPONDING TO CHOICE":INPUT A
170 IF A<>1 AND A<>2 AND A<>3 THEN GOTO    160
180 IF A=3 THEN DLOAD"PRICELIST"
190 PRINT "ENTER THE NAME OF THE FILE TO BE "
200 INPUT "DUMPED";A$
```

```
210 OPEN 5,8,5,"0:"+A$+",S,R"
215 IF A=1 THEN OPEN1,4
220 GET#5,A$
230 IF ST<>0 THEN PRINT:PRINT"SYSTEM STATUS IS "ST:GOTO 260
240 IF A=1 THEN PRINT A$;
245 IF A=2 THEN PRINT#1,A$;
250 GOTO 220
260 IF A=2 THEN PRINT#1:CLOSE 1
270 CLOSE 5
280 PRINT "PRESS ANY KEY TO RETURN TO MENU"
290 GET A$:IF A$="" THEN GOTO 290
300 GOTO 10
310 END
```

P32 TIDY

This is the only program in the book that cannot easily be adapted for use with a cassette.

This program tidied up our outputs for the purposes of printing this book.

You might find this program useful when you send your listings to magazines, or whatever.

The program takes the CBM-16 cursor control characters and writes them out in a printable form.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM GRAPHICS TIDY
20 PRINT "[CS ]"
30 PRINT "[CD CD ]THIS PROGRAM IS USED TO TIDY UP PRINTER"
40 PRINT "OUTPUT FOR OUR LISTINGS. WE HAVE USED"
50 PRINT "THE FOLLOWING CONVENTION FOR PRINTING"
60 PRINT "GRAPHICS."
70 PRINT "[CD CD CR ]1. ALL GRAPHICS ARE ENCLOSED  BETWEEN  [
]";
80 PRINT "[CR ]2. THE SHIFT GRAPHIC KEYS ARE SHOWN AS"
90 PRINT "[CR CR CR CR ]THE NORMAL LETTER"
100 PRINT "[CR ]3. THE COMMODORE KEY GRAPHICS ARE SHOWN";
110 PRINT "[CR CR CR CR ]AS LOWER CASE LETTERS"
120 PRINT "[CR ]4. THE CURSOR CONTROL KEYS ARE SHOWN"
130 PRINT "[CR CR CR CR ]AS C-L,C-R,C-U AND C-D FOR LEFT"
140 PRINT "[CR CR CR CR ]RIGHT, UP AND DOWN"
150 PRINT "[CR ]5. CLEAR IS SHOWN AS C-S, HOME AS C-H"
160 PRINT "[CD CD ]WHAT IS THE NAME OF THE SEQUENTIAL"
170 PRINT "PROGRAM FILE";
180 INPUT F$
190 INPUT"ENTER OUTPUT FILE NAME";FO$
200 PRINT "PROCESSING";
```

```
210 OPEN 6,8,6,"0:"+F$+",S,R"
220 OPEN 5,8,5,"0:"+FO$+",S,W"
230 OPEN 15,8,15
240 GOSUB 520
250 Z=0:REM FLAG TO SAY WE ARE IN GRAPHICS
260 REM THE FOLLOWING LOOP IS THE TIDY LOOP
270 GET#6,A$
280 IF ST<>0 THEN 470
290 A=ASC(A$+CHR$(0))
300 IF A>31 AND A<95 THEN GOTO 430
310 IF A=13 THEN GOTO 430
320 IF Z=0 THEN PRINT#5,"[";:Z=1
330 IF A=17 THEN PRINT#5,"CD ";
340 IF A=19 THEN PRINT#5,"CH ";
350 IF A=29 THEN PRINT#5,"CR ";
360 IF A=145 THEN PRINT#5,"CU ";
370 IF A=157 THEN PRINT#5,"CL ";
380 IF A=147 THEN PRINT#5,"CS ";
390 IF A=18 THEN PRINT#5,"RON ";
400 IF A=146 THEN PRINT#5,"ROF ";
410 IF A=126 THEN PRINT#5,"PI ";
420 GOTO 450
430 IF Z=1 THEN PRINT#5,"]";:Z=0
440 PRINT#5,A$;
450 PRINT ".";
460 GOTO 270
470 CLOSE 6
480 CLOSE 5
490 PRINT:PRINT "JOB FINISHED"
500 PRINT "STATUS = "ST
510 END
520 REM CHECK SUCCESSFUL OPEN
530 INPUT#15,A$,B$,C$,D$
540 IF VAL(A$)>0 THEN PRINT "DISK  FAULT":PRINT  A$,B$,C$,D$:
STOP
550 RETURN
```

P33 Monthly accounts

In our households, one of the tasks which has to be done is the monthly budget. It seemed to us that this was an ideal use for the Commodore 16.

In this program we again make use of the alternative character set of the C-16, all words presented are in both upper and lower case. Thus when entering the program we first tell you to press the Commodore and shift keys together.

Once the program is running, it prompts users to enter all their outgoings and income. The program ends with a report on to paper or screen of the monthly budget.

COMMANDS

Press the Commodore and shift keys together.
Key in the program and RUN.
Follow instructions.

```
10 rem program - monthly accounts
20 scnclr
30 print tab(10);"[ron ]monthly accounts[rof ]"
40 a=ti
50 if ti<a+150 then 50
60 print chr$(14)
70 print "[cs cd cd cd cd cd ]"
80 print " This program helps you to budget your"
90 print " monthly outgoings and incomings."
100 print " The program prompts you to enter your"
110 print " financial dealings and produces a"
120 print " balance sheet for the next month."
130 print "[cd cd ]"
140 print " If you want to use a printer then "
150 print " ensure that it is connected up now."
160 print "[cd cd ] Press any key to continue";
170 get a$:if a$="" then 170
180 print "[cs ]";
```

```
190 print "[ron ]          Regular Outgoings             [rof
]";
200 print
210 print "Enter mortgage repayments. $";
220 input m
230 input "Enter rent payments. $";re
240 input "Enter property taxes. $";pt
250 input "Enter electricity payments. $";e1
260 input "Enter gas payments. $";gs
270 cr=0
280 rem take in all credit card repayments
290 do while c<>0
300 print "[cd ron ]Credit Card Repayments[rof cd ]"
310 input "Enter credit repayment. 0 for last";c
320 cr=cr+c
330 loop
340 print "[cd ron ]Enter any other regular payments.[rof cd
]"
350 input "Enter total. $";ot
360 rg=m+re+pt+e1+gs+cr+ot:rem regular payments
370 print "[cs ]";
380 print "[ron ]      Other Outgoings This Month       [rof
]"
390 print "Use estimates if actual amounts are not"
400 print "known.[cd cd ]"
410 print "Enter total of all outstanding bills."
420 input "$";ob:rem old bills
430 input "Grocery bill. $";g
440 input "Butcher's bill. $";b
450 input "Entertainment allowance. $";en
460 print "[cs ]";
470 bt=g+b+en:rem bills this month
480 print "[ron ]      Other bills due this month       [rof
]"
490 print "[cd ]Enter other bills - use 0 to finish[cd ]"
500 do while bl<>0
510 input "Amount of bill. $";bl
520 bs=bs+bl
530 loop
540 bt=bt+bs
550 tt=rg+ob+bt:rem total outgoings
560 print "[cd cd ]Total outgoings = $";tt
570 print "[cd cd ]Press any key to continue"
580 get a$:if a$="" then 580
590 print "[cs ]";
600 print "[ron ]           Monthly Income             [rof
]"
```

```
610 input "[cd ]Enter your monthly net income. $";i1
620 print "[cd ]Enter your spouse's monthly net income"
630 input "$";i2
640 input "[cd ]Enter any other net income. $";i3
650 i=i1+i2+i3
660 ba=i-tt:rem balance
670 print "[cd cd ]Total income is $";i
680 print "[cd cd ]Press any key to continue"
690 get a$:if a$="" then 690
700 print "[cs cd cd cd cd cd ]"
710 print " Do you want the report to be printed"
720 print " on the screen or onto the printer?"
730 print
740 print "      1.   Screen"
750 print "      2.   Printer"
760 print
770 input " Enter 1 or 2 ";r
780 if r<>1 and r<>2 then print
"[cu ]                        [cu ]":goto 770
790 if r=2 then open 1,4,7:cmd 1
800 print "[cs ]";
810 print
"[ron ]            MONTHLY BUDGET                [rof ]"
820 print
830 print"[ron ]  COMMENT        OUT      IN      [rof ]"
840 print "Regular"
850 print "amounts";spc(18-len(str$(rg)));rg
860 print "[cd ]Bills this"
870 print "month  ";spc(18-len(str$(bt)));bt
880 print "[cd ]Outstanding "
890 print "bills from"
900 print"last month";spc(15-len(str$(ob)));ob
910 print
920 print "[ron ]Total Out[rof ]";spc(16-len(str$(tt)));tt
930 print
940 print "[ron ]Total Income[rof ]";spc(23-len(str$(i)));i
950 print
960 if ba<0 then print "[ron ]Debit Balance[rof ]";
spc(22-len(str$(ba)));ba:goto 980
970 print "[ron ]Credit Balance[rof ]";spc(11-len(str$(ba)));
ba
980 get a$:if a$="" then goto 980
990 if r=2 then close 1
1000 end
```

P34 Conversion

This is a general purpose conversion utility, which we have implemented with eighteen different conversion factors. It is fairly straightforward to choose other units to be converted by changing the data statements at the end of the program.

Eighteen conversion factors were chosen to make the main menu fit the screen.

Once the conversion has been chosen, you have to decide which way the conversion has to proceed. For example, centimeters to inches or inches to centimeters.

Note that the American gallon (6 pints) is used.

COMMANDS

Key in the program and RUN.
Select from menu.

```
10 REM PROGRAM - CONVERSION UTILITY
20 PRINT "[CS CD CD CD CD CD CD CD CD CD ]"
30 PRINT "           CONVERSIONS UTILITY"
40 A=TI
50 IF TI<A+150 THEN GOTO 50
60 PRINT "[CS ]"
70 PRINT " THIS PROGRAM IS USED AS A GENERAL"
80 PRINT " PURPOSE CONVERSION UTILITY. THE DATA"
90 PRINT " FOR THE CONVERSIONS ARE HELD WITHIN"
100 PRINT " THE PROGRAM."
110 PRINT "[CD ]PRESS ANY KEY TO CONTINUE"
120 DIM C$(18,2),D(18)
130 GET A$:IF A$="" THEN GOTO 130
140 PRINT
"[CS RON ]            CONVERSIONS                  [ROF ]"
150 RESTORE
160 FOR I=1 TO 18
170 READ C$(I,1),C$(I,2)
180 PRINT I SPC(6-LEN(STR$(I)));C$(I,1);" TO ";C$(I,2)
190 NEXT I
200 PRINT
```

```
210 PRINT "CHOOSE ITEM BY ENTERING THE REQUIRED"
220 INPUT "NUMBER";C
225 IF C=0 THEN STOP
230 FOR I=1 TO 18:READ D(I):NEXT I
240 PRINT "[CS ]"
250 PRINT "[RON ]          OPTIONS AVAILABLE              [ROF
]"
260 PRINT:PRINT
270 PRINT "1. ";C$(C,1);" TO ";C$(C,2)
280 PRINT "2. ";C$(C,2);" TO ";C$(C,1)
290 PRINT
300 PRINT "CHOOSE OPTION BY ENTERING APPROPRIATE "
310 INPUT "NUMBER";CH
320 IF CH=2 THEN D(C)=1/D(C):T$=C$(C,1):C$(C,1)=C$(C,2):
C$(C,2)=T$
330 REM IF ALTERNATIVE OPTION REQUIRED
340 REM THEN INVERT FACTOR AND SWAP ITEMS
350 PRINT:PRINT
360 PRINT "ENTER VALUE OF "C$(C,1);
370 INPUT A
380 B=A*D(C)
390 PRINT:PRINT
400 PRINT A;" ";C$(C,1);" = ";B;" ";C$(C,2)
410 PRINT:PRINT
420 PRINT "PRESS ANY KEY TO CONTINUE"
430 GET A$:IF A$="" THEN 430
440 GOTO  140
450 END
1000 DATA INCHES,CM,FEET,METERS,YARDS,METERS,MILES,KM
1010 DATA TEASPOONS,CC,TABLESPOONS,CC,CUPS,LITERS,PINTS
1020 DATA LITERS,QUARTS,LITERS,GALLONS,LITERS,OZ,GRAMS
1030 DATA POUNDS,KILOGRAMS,TONS,KILOGRAMS,MILES PER HOUR,
METERS PER SEC
1040 DATA SQUARE YARDS,SQUARE METERS,YEARS,SECONDS
1050 DATA ATMOSPHERES,CM HG,ACRES,HECTARES
1060 REM CONVERSION DATA
1070 DATA 2.540,.3048,.9144,1.609,4.929,14.788,.2366
1080 DATA .4732,.7463,3.785,28.3495,.4536,907.2,.447,.8361,
31536000
1090 DATA 76,.4047
```

P35 Birthday list

This program uses one of the SORT routines developed elsewhere in the book, the so-called Bubble Sort.

The Birthday List program is used to store all those birthdays that you have to remember. The program sorts the birthdays into date order, and then prints out all bithdays left in the current year. All birthdays are stored in data statements.

COMMANDS

Key in the program and RUN.
Follow instructions.
When you can folllow the operation of the program, amend the data statements.

```
10 REM PROGRAM - BIRTHDAY LIST
20 SCNCLR:PRINT TAB(11);"BIRTHDAYS":PRINT:PRINT
30 PRINT "THE BIRTHDAYS HELD IN THIS PROGRAM ARE"
40 PRINT"HELD IN THE FORM OF DATA STATEMENTS"
50 PRINT "STARTING AT LINE 2000."
60 PRINT
70 PRINT "THE PROGRAM CAN BE USED TO HOLD UP TO "
80 PRINT "100 BIRTHDAYS, BUT OF COURSE THIS COULD"
90 PRINT "BE AMENDED IF YOU SO WISH."
100 PRINT "THE FORMAT OF THE DATA IS AS FOLLOWS:"
110 PRINT
120 PRINT "DATA  NAME,MMDD"
130 PRINT
140 PRINT "WHERE MM IS THE MONTH NUMBER AND DD IS"
150 PRINT "THE DAY NUMBER"
160 PRINT
170 PRINT "MAKE THE LAST TWO ELEMENTS OF THE DATA"
180 PRINT "BE  EOF,0"
190 PRINT
200 PRINT "PRESS ANY KEY TO CONTINUE"
```

```
210 GETKEY A$
220 PRINT "[CS CD CD CD ]"
230 DIM N$(100),D(100)
240 REM READ IN DATA
250 I=0
260 DO
270 I=I+1
280 READ N$(I),D(I)
290 LOOP UNTIL N$(I)="EOF"
300 I=I-1
310 GOSUB 620
320 REM USEFUL GET DATE ROUTINE
330 PRINT "WHAT IS TODAY'S DATE (MM/DD)[CL CL CL CL CL CL
]";:D$=""
340 GETKEY A$
350 PRINT A$;:D$=D$+A$
360 GETKEY A$
370 PRINT A$;:D$=D$+A$
380 PRINT "/";
390 GETKEY A$
400 PRINT A$;:D$=D$+A$
410 GETKEY A$
420 PRINT A$;:D$=D$+A$
430 FOR Z=1 TO 4
440 V=ASC(MID$(D$,Z,1))
450 IF V<48 OR V>57 THEN Z=4:T=1
460 NEXT Z
470 IF T=1 THEN PRINT:T=0:D$="":GOTO 330
480 D=VAL(D$)
490 PRINT "[CS ]"
500 PRINT : PRINT : PRINT
510 PRINT "THE BIRTHDAYS LEFT THIS YEAR ARE:"
520 PRINT : PRINT : PRINT
530 MT$="JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC"
540 FOR J=1 TO I
550 IF D(J)<D THEN NEXT J
560 DAY=D(J)-100*INT(D(J)/100)
570 MO=INT(D(J)/100)-1
580 PRINT N$(J);"'S BIRTHDAY IS ";DAY;" ";MID$(MT$,3*MO+1,3)
590 NEXT J
600 END
610 REM SORT ROUTINE
620 FOR X=1 TO I-1
630 FOR Y=X+1 TO I
640 IF D(Y)<D(X) THEN GOSUB 690
650 NEXT Y
660 NEXT X
```

```
670 RETURN
680 REM SWAP ROUTINE
690 T=D(Y):T$=N$(Y)
700 D(Y)=D(X):N$(Y)=N$(X)
710 D(X)=T:N$(X)=T$
720 RETURN
730 REM BIRTHDAYS
740 DATA JIM SMITH,1130,BILL SMYTHE,0812,LIZ GRAHAM,0303
750 DATA JOHN GORDON,0419,JOHN FERGUSON,0407,TERESA
GORDON,1222,EOF,0
```

P36 Calendar

This program can be used to print out the calendar for any month in the twentieth century.

COMMANDS

Key in the program and RUN.
Enter month and year when requested.

```
10 REM PROGRAM - CALENDAR
20 PRINT "[CS CD CD CD ]THIS PROGRAM CAN BE USED TO PRINT THE
"
30 PRINT "CALENDAR FOR ANY MONTH IN THE TWENTIETH"
40 PRINT "CENTURY."
50 DIM C(7,6)
60 RESTORE
65 PRINT "[CD CD ]"
70 INPUT "WHICH MONTH DO YOU REQUIRE";M$
80 INPUT "WHICH YEAR DO YOU REQUIRE";Y
90 PRINT "CALCULATING - PLEASE WAIT"
100 D$="MONTUEWEDTHUFRISATSUN"
110 REM GET NUMBER OF DAYS IN MONTH
120 M=0
130 READ Q$,F,D
140 M=M+1
150 IF M=13 THEN PRINT "INVALID MONTH":GOTO 60
160 IF LEFT$(M$,3)<>LEFT$(Q$,3) THEN GOTO 130
170 IF Y>1900 THEN Y=Y-1900
180 L=INT(Y/4)
190 OF=Y+L
200 IF Y-INT(Y/4)*4=0 AND M<3 THEN OF=OF-1
210 IF Y-INT(Y/4)*4=0 AND M=2 THEN D=29
220 OF=OF-INT(OF/7)*7
230 F=F+OF
240 F=F-INT(F/7)*7
250 IF F=0 THEN F=7
260 FOR I=1 TO F-1
270 C(I,1)=0
280 NEXT I
290 DY=1
300 FOR I=F TO 7
```

```
310 C(I,1)=DY
320 DY=DY+1
330 NEXT I
340 FOR J=2 TO 6
350 FOR I=1 TO 7
360 C(I,J)=DY
370 DY=DY+1
380 IF DY>D THEN I=7:J=6
390 NEXT I
400 NEXT J
410 PRINT "[CS ]"
420 PRINT "[CD CD CD CR CR CR CR CR CR CR CR CR CR CR CR CR
CR ]";M$;"   ";Y+1900;"[CD CD ]"
430 FOR I=1 TO 7
440 PRINT "     ";MID$(D$,(I-1)*3+1,3)
450 NEXT I
460 PRINT "[CH CD CD CD CD CD CD CD CR CR CR CR CR CR CR CR
CR ]";
470 FOR J=1 TO 6
480 FOR I=1 TO 7
485 C$=STR$(C(I,J)):IF LEN(C$)=2 THEN C$=" "+C$
490 IF C(I,J)<>0 THEN PRINT C$;"[CD CL CL CL ]";
495 IF C(I,J)=0 THEN PRINT"   ";"[CD CL CL CL ]";
500 NEXT I
510 PRINT "[CU CU CU CU CU CU CU CR CR CR ]";
520 NEXT J
530 GET A$:IF A$="" THEN 530
540 DATA
JANUARY,1,31,FEBRUARY,4,28,MARCH,4,31,APRIL,7,30,MAY,2,31,JUN
E,5,30
550 DATA JULY,7,31,AUGUST,3,31,SEPTEMBER,6,30,OCTOBER,1,31
560 DATA NOVEMBER,4,30,DECEMBER,6,31
570 DATA ,,
```

P37 Telephone list

This program allows the user to select a telephone number from a list held as data statements.

You do not have to enter the full name to select the number. As long as the string entered is part of one of the names, then the telephone number is displayed.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM - TELEPHONE LIST
20 SCNCLR
30 PRINT
40 PRINT "THIS PROGRAM IS USED AS A PERSONAL "
50 PRINT "TELEPHONE DIRECTORY."
60 PRINT
70 PRINT "THE DATA ARE HELD IN THE LINES FROM 1000"
80 PRINT "THE PROGRAM CAN HOLD A DATABASE OF UP TO";
90 PRINT "100 NAMES AND TELEPHONE NUMBERS. THIS"
100 PRINT "CAN OF COURSE BE AMENDED BY THE USER."
110 PRINT
120 PRINT "THE PROGRAM WILL SEARCH THROUGH THE"
130 PRINT "DATABASE LOOKING FOR A MATCH AGAINST"
140 PRINT "A NAME INPUT TO THE COMPUTER."
150 PRINT
160 PRINT "A PARTIAL MATCH WILL ALSO GIVE NUMBER."
170 PRINT "FOR EXAMPLE, IF YOU INPUT 'BILL', THEN"
180 PRINT "A MATCH WILL BE FOUND FOR BILL SMITH "
190 PRINT "AND BILL JONES"
200 PRINT
210 PRINT "PRESS ANY KEY TO CONTINUE"
220 GETKEY A$
230 DIM N$(100,2)
240 I=0
250 DO
260 I=I+1
270 READ N$(I,1),N$(I,2)
```

```
280 LOOP UNTIL N$(I,1)="EOF"
290 I=I-1
300 C=0
310 SCNCLR
320 PRINT "WHOSE NUMBER DO YOU REQUIRE"
330 INPUT N$
340 LA=LEN(N$):F=0
350 FOR J=1 TO I
360 LB=LEN(N$(J,1))
370 IF LA>LB THEN NEXT J
380 FOR K=1 TO LB-LA
390 IF N$=MID$(N$(J,1),K,LA) THEN F=1
400 NEXT K
410 IF F=1 THEN PRINT N$(J,1);"'S NUMBER IS
";N$(J,2):F=0:C=C+1
420 NEXT J
430 IF C=0 THEN PRINT "WE HAVE NO NUMBER FOR ";N$
440 PRINT "ANOTHER NUMBER (Y/N)"
450 GETKEY A$
460 IF A$="Y" THEN GOTO 300
470 END
480 DATA BILL SMITH,1235,BILL JONES,45678
490 DATA ALICE GRAHAM,0101 256 3456
500 DATA JOHN GORDON,03552 49400
510 DATA EOF,0
```

P38 Investments

This program calculates how much income can be generated from capital if the interest earned by that capital is known.

The user is presented with the choice of increasing his or her income at the expense of reducing capital.

COMMANDS

Key in the program and RUN.

```
10 REM PROGRAM - INVESTMENTS
20 SCNCLR
30 PRINT "THIS PROGRAM ALLOWS YOU TO PLAN YOUR "
40 PRINT "FUTURE INVESTMENTS, GIVEN THAT YOU WISH"
50 PRINT "YOUR INVESTMENTS TO SUPPLY A REGULAR"
60 PRINT "INCOME."
70 PRINT
80 PRINT "PRESS ANY KEY TO CONTINUE"
90 GETKEY A$
100 PRINT
110 INPUT "MONTHLY INCOME REQUIRED $";I
120 PRINT "ENTER CURRENT ANNUAL INTEREST RATE (%)"
130 INPUT R
140 R=R/100
150 IN=I*12/R
160 PRINT "AN INVESTMENT OF $";INT(IN*100)/100
170 PRINT "WILL PROVIDE A MONTHLY INCOME OF $";I
180 PRINT
190 PRINT "NOTE THAT NO CAPITAL HAS BEEN USED"
200 PRINT
210 PRINT "IF YOU WISH TO USE UP YOUR CAPITAL."
220 PRINT "HOW MANY YEARS DO YOU WISH AN INCOME"
230 PRINT "FOR? ENTER NUMBER OF YEARS."
240 INPUT YR
250 R=R/12
260 T=(1+R)^(12*YRS)
```

```
270 IN=I*(T-1)/R/T
280 PRINT
290 PRINT "AN INVESTMENT OF $";INT(IN*100)/100
300 PRINT "WILL PROVIDE AN INCOME OF $";I
310 PRINT "FOR ";YR;" YEARS"
320 END
```

P39 Loan repayment schedule

We use this program to try to dissuade ourselves from buying something using a loan from a finance house.

It can be quite illuminating to have a note of the full schedule of repayments for a loan and to see how slowly the amount owed drops.

If the interest rate changes during the period of a loan, then simply RUN the program again as if you had taken out a new loan, with a reduced amount borrowed.

The program allows you to choose either a screen or a printer output. If you have a printer which does not have automatic line feed on carriage return, Then use a file number greater than 128 for the printer control instructions. As it stands the program will work with the Commodore MPS801 and with most other printers.

COMMANDS

Key in the program and RUN.
Follow the instructions.

```
10 REM LOAN REYPAYMENT SCHEDULE
20 REM ************************
30 REM
40 COLOR 0,1:REM BLACK SCREEN
50 COLOR 1,4,6:REM CYAN INK
60 COLOR 4,1:REM BLACK BORDER
70 SCNCLR
80 PRINT:PRINT
90 PRINT TAB(8)"LOAN REPAYMENT SCHEDULE"
100 PRINT TAB(8)"$$$$$$$$$$$$$$$$$$$$$$$"
110 PRINT:PRINT
120 PRINT TAB(4)"THIS PROGRAM CAN BE USED TO PLAN"
130 PRINT TAB(4)"THE REPAYMENTS SCHEDULE FOR A"
140 PRINT TAB(4)"LOAN. THE LOAN IS PAID BACK AT A"
150 PRINT TAB(4)"FIXED MONTHLY RATE. THE INTEREST"
160 PRINT TAB(4)"IS ALSO FIXED. THE PROGRAM"
170 PRINT TAB(4)"PRODUCES A REPAYMENT TABLE."
```

```
180 PRINT
190 PRINT TAB(4)"IF YOU WISH TO USE A PRINTER,"
200 PRINT TAB(4)"ENSURE ONE IS CONNECTED NOW. THE"
210 PRINT TAB(4)"PROGRAM ASSUMES THE PRINTER IS"
220 PRINT TAB(4)"DEVICE NUMBER 4."
230 PRINT:PRINT
240 PRINT TAB(6)CHR$(18)" PRESS ANY KEY TO CONTINUE "
250 GETKEY A$
260 REM*****************************
270 SCNCLR:PRINT
280 INPUT"WHAT IS THE AMOUNT OF THE LOAN";AM
290 PRINT
300 INPUT"WHAT IS THE INTEREST/MONTH (%)";IT
310 PRINT
320 PRINT"WHAT IS THE NUMBER (1-12) OF THE"
330 INPUT"FIRST REYPAYMENT MONTH";MN
340 PRINT
350 INPUT"WHAT IS THE MONTHLY REPAYMENT";RP
360 PRINT
370 INPUT"WHAT YEAR IS IT";YR
380 REM NO TRAPS ON ENTRIES. SILLY ENTRIES  WILL  GIVE  SILLY
RESULTS.
390 REM*****************************
400 SCNCLR:PRINT:PRINT:PRINT:PRINT
410 PRINT TAB(4)"SELECT YOUR OUTPUT DEVICE BY"
420 PRINT TAB(4)"PRESSING:"
430 PRINT
440 PRINT TAB(10)CHR$(18)" KEY 1 "CHR$(146)" - SCREEN"
450 PRINT
460 PRINT TAB(10)CHR$(18)" KEY 2 "CHR$(146)" - PRINTER"
470 DO:GET A$: LOOP UNTIL A$="1" OR A$="2"
480 DV=VAL(A$)
490 REM*****************************
500 SCNCLR
510 ON DV GOSUB 780,850:REM HEADINGS
520 PD=0:REM AMOUNT PAID BACK
530 REM*****************************
540 REM CALCULATION
550 DO
560 :  MN=MN+1
570 :  IF MN=13 THEN YR=YR+1:MN=1
580 :  AM=AM*(1+IT/100)
590 :  AM=AM-RP
600 :  PD=PD+RP
610 :  IF AM<0 THEN PD=PD+INT(100*AM+.5)/100:AM=0
620 :  ON DV GOSUB 930,980:REM PRINT RESULTS
630 LOOP UNTIL AM=0
```

```
630 LOOP UNTIL AM=0
640 IF DV=2 THEN CLOSE 3
650 END
660 REM*****************************
670 REM*****************************
680 REM
690 REM       ***************
700 REM       *             *
710 REM       * SUBROUTINES *
720 REM       *             *
730 REM       ***************
740 REM
750 REM*****************************
760 REM*****************************
770 REM SCREEN HEADING
780 PRINT TAB(3)"YEAR","MONTH","AMOUNT","AMOUNT"
790 PRINT TAB(21)"PAID"," OWED"
800 PRINT TAB(3)"----","-----","------","------"
810 RETURN
820 REM*****************************
830 REM*****************************
840 REM PRINTER HEADING
850 OPEN 3,4
860 PRINT#3,TAB(3)"YEAR","MONTH","AMOUNT","AMOUNT"
870 PRINT#3,TAB(33)"PAID","  OWED"
880 PRINT#3,TAB(3)"----","-----","------","------"
890 RETURN
900 REM*****************************
910 REM*****************************
920 REM SCREEN RESULTS
930 PRINT TAB(2)YR,MN,PD,INT(AM*100+.5)/100
940 RETURN
950 REM*****************************
960 REM*****************************
970 REM PRINTER RESULTS
980 PRINT#3,TAB(2)YR,MN,PD,INT(AM*100+.5)/100
990 RETURN
1000 REM*****************************
1010 REM*****************************
```

YEAR	MONTH	AMOUNT PAID	AMOUNT OWED
1984	7	200	4875
1984	8	400	4748.13
1984	9	600	4619.35
1984	10	800	4488.64
1984	11	1000	4355.97
1985	12	1200	4221.31
1985	1	1400	4084.63
1985	2	1600	3945.9
1985	3	1800	3805.08
1985	4	2000	3662.16
1985	5	2200	3517.09
1985	6	2400	3369.85
1985	7	2600	3220.4
1985	8	2800	3068.7
1985	9	3000	2914.73
1985	10	3200	2758.45
1985	11	3400	2599.83
1986	12	3600	2438.83
1986	1	3800	2275.41
1986	2	4000	2109.54
1986	3	4200	1941.18
1986	4	4400	1770.3
1986	5	4600	1596.86
1986	6	4800	1420.81
1986	7	5000	1242.12
1986	8	5200	1060.75
1986	9	5400	876.67
1986	10	5600	689.82
1986	11	5800	500.16
1987	12	6000	307.66
1987	1	6200	112.28
1987	2	6313.96	0

P40 Space shapes

This program generates some space type shapes which you may find useful in (say) games programs.

COMMANDS

Key in the program and RUN.
Press the space bar to return to the normal character set.

```
10 REM SPACE GAME SHAPES
20 REM *****************
30 REM
40 POKE 52,47:POKE 56,47:CLR:REM RESERVE RAM
50 REM**************************
60 MC=12032:REM START OF MACHINE CODE ROUTINE
70 RA=12288:REM START OF CHARACTER RAM
80 CC=65298:REM CONTROLS CHARACTER LOCATION
90 COLOR 0,1:REM BLACK SCREEN
100 COLOR 1,2:REM WHITE INK
110 COLOR 4,7,3:REM BLUE BORDER
120 REM**************************
130 SCNCLR:PRINT:PRINT
140 PRINT TAB(18)"WAIT"
150 RESTORE
160 FOR N=0 TO 33
170 :  READ A:POKE MC+N,A
180 NEXT:REM LOAD MACHINE CODE ROUTINE
190 SYS MC:REM TRANSFER CHARACTER SET
200 R1=PEEK(CC)
210 R2=PEEK(CC+1)
220 POKE CC,R1 AND 251:REM CHARACTERS READ FROM RAM
230 POKE CC+1,(R2 AND 3) OR 48:REM POINT TO CHARACTERS
240 REM**************************
250 FOR N=33 TO 37
260 :  FOR K=0 TO 7
270 :     READ A
280 :     POKE RA+K+N*8,A
290 :  NEXT
300 NEXT
310 REM**************************
320 SCNCLR
```

```
330 Y=5
340 CH=33
350 CHAR,5,Y,"MERCURIAN MUTANT"
360 PRINT TAB(30)CHR$(CH)
370 CH=CH+1:Y=Y+2
380 COLOR 1,4,5
390 CHAR,5,Y,"NEPTUNIAN NASTY"
400 PRINT TAB(30)CHR$(CH)
410 CH=CH+1:Y=Y+2
420 COLOR 1,6,6
430 CHAR,5,Y,"SILUSIAN SAUCER"
440 PRINT TAB(30)CHR$(CH)
450 CH=CH+1:Y=Y+2
460 COLOR 1,8,5
470 CHAR,5,Y,"CANNON"
480 PRINT TAB(30)CHR$(CH)
490 CH=CH+1:Y=Y+2
500 COLOR 1,9,7
510 CHAR,5,Y,"SHELL"
520 PRINT TAB(30)CHR$(CH)
530 DO
540 :  GET A$
550 LOOP UNTIL A$=" ":REM SPACE BAR STOPS PROGRAM
560 POKE CC,R1
570 POKE CC+1,R2
580 END
590 REM**********************
600 REM**********************
610 REM
620 REM
630 REM         **********
640 REM         *        *
650 REM         *        *
660 REM         *  DATA  *
670 REM         *        *
680 REM         *        *
690 REM         **********
700 REM
710 REM
720 REM******************************
730 REM DATA FOR MACHINE CODE ROUTINE
740 REM******************************
750 REM
760 DATA 169,0,133,3,133,5,169,48,133,4,169,208
770 DATA 133,6,160,0,177,5,145,3,200,208,249
780 DATA 230,4,230,6,169,216,197,6,208,239,96
790 REM
```

```
800 REM*************************
810 REM DATA FOR CHARACTER SHAPES
820 REM*************************
830 REM
840 DATA 56,124,84,124,254,254,170,170
850 DATA 66,60,189,165,255,60,36,231
860 DATA 0,60,126,171,255,126,36,36
870 DATA 24,24,24,24,24,255,255,255
880 DATA 0,0,24,24,24,24,0,0
890 REM*************************
```

P41 More shapes

This program generates some more user defined shapes. It also demonstrates how a larger picture may be generated - in this case a city is built up from a number of city units.

COMMANDS

Key in the program and RUN.
Press the space bar to return to the normal character set.

```
10 REM MORE SHAPES
20 REM ***********
30 REM
40 POKE 52,47:POKE 56,47:CLR:REM RESERVE RAM
50 REM**************************
60 MC=12032:REM START OF MACHINE CODE ROUTINE
70 RA=12288:REM START OF CHARACTER RAM
80 CC=65298:REM CONTROLS CHARACTER LOCATION
90 COLOR 0,1:REM BLACK SCREEN
100 COLOR 1,2:REM WHITE INK
110 COLOR 4,7,3:REM BLUE BORDER
120 REM**************************
130 SCNCLR:PRINT:PRINT
140 PRINT TAB(18)"WAIT"
150 RESTORE
160 FOR N=0 TO 33
170 :  READ A:POKE MC+N,A
180 NEXT:REM LOAD MACHINE CODE ROUTINE
190 SYS MC:REM TRANSFER CHARACTER SET
200 R1=PEEK(CC)
210 R2=PEEK(CC+1)
220 POKE CC,R1 AND 251:REM CHARACTERS READ FROM RAM
230 POKE CC+1,(R2 AND 3) OR 48:REM POINT TO CHARACTERS
240 REM**************************
250 FOR N=33 TO 38
260 :  FOR K=0 TO 7
270 :     READ A
280 :     POKE RA+K+N*8,A
290 :  NEXT
300 NEXT
310 REM**************************
```

```
320 SCNCLR
330 Y=3
340 CH=33
350 CHAR,5,Y,"FIGURE"
360 PRINT TAB(30)CHR$(CH)
370 CH=CH+1:Y=Y+2
380 COLOR 1,4,5
390 CHAR,5,Y,"MONSTER"
400 PRINT TAB(30)CHR$(CH)
410 CH=CH+1:Y=Y+2
420 COLOR 1,6,6
430 CHAR,5,Y,"PIT"
440 PRINT TAB(30)CHR$(CH)
450 CH=CH+1:Y=Y+2
460 COLOR 1,8,5
470 CHAR,5,Y,"BOMB"
480 PRINT TAB(30)CHR$(CH)
490 CH=CH+1:Y=Y+2
500 COLOR 1,9,7
510 CHAR,5,Y,"GUNSHIP"
520 PRINT TAB(30)CHR$(CH)
530 CH=CH+1:Y=Y+2
540 COLOR 1,14,5
550 CHAR,5,Y,"CITY"
560 C$=CHR$(38):S$=CHR$(32)
570 D$=C$+C$+S$+S$+S$+S$+S$+S$+S$+S$+S$+S$+S$+C$+C$+C$+S$
580 D$=D$+S$+S$+S$+S$+S$+S$+S$+S$+C$+S$+S$+S$+S$+S$+S$
590 E$=C$+C$+S$+S$+S$+S$+S$+S$+S$+C$+C$+C$+S$+C$+C$+C$+S$
600 E$=E$+S$+S$+S$+S$+S$+S$+C$+S$+C$+C$+C$+C$+S$+S$+S$
610 F$=C$+C$+S$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+S$
620 F$=F$+S$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+S$+C$
630 G$=C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+S$
640 G$=G$+S$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$+C$
650 PRINT:PRINT:PRINT:PRINT
660 PRINT TAB(4)D$
670 PRINT TAB(4)D$
680 PRINT TAB(4)E$
690 PRINT TAB(4)E$
700 PRINT TAB(4)F$
710 PRINT TAB(4)G$
720 PRINT TAB(4)G$
730 DO
740 :  GET A$
750 LOOP UNTIL A$=" ":REM SPACE BAR STOPS PROGRAM
760 POKE CC,R1
770 POKE CC+1,R2
780 END
```

```
790 REM**********************
800 REM**********************
810 REM
820 REM
830 REM         **********
840 REM         *        *
850 REM         *        *
860 REM         *  DATA  *
870 REM         *        *
880 REM         *        *
890 REM         **********
900 REM
910 REM
920 REM******************************
930 REM DATA FOR MACHINE CODE ROUTINE
940 REM******************************
950 REM
960 DATA 169,0,133,3,133,5,169,48,133,4,169,208
970 DATA 133,6,160,0,177,5,145,3,200,208,249
980 DATA 230,4,230,6,169,216,197,6,208,239,96
990 REM
1000 REM*************************
1010 REM DATA FOR CHARACTER SHAPES
1020 REM*************************
1030 REM
1040 DATA 28,28,8,127,28,20,20,54
1050 DATA 31,124,200,248,248,120,60,31
1060 DATA 60,126,255,254,254,62,30,28
1070 DATA 60,60,24,60,60,60,60,24
1080 DATA 24,126,231,219,219,231,126,0
1090 DATA 255,153,153,255,255,129,129,255
1100 REM*************************
```

P42 Destroyer

In this program three user defined characters make up the composite shape of a destroyer, which is then moved back and forth across the screen under program control. Notice how the destroyer is made to reverse direction by overprinting it with a mirror image.

COMMANDS

Key in the program and RUN.
Press the space bar to return to the normal character set

```
10 REM DESTROYER
20 REM *********
30 REM
40 POKE 52,47:POKE 56,47:CLR:REM RESERVE RAM
50 REM**************************
60 MC=12032:REM START OF MACHINE CODE ROUTINE
70 RA=12288:REM START OF CHARACTER RAM
80 CC=65298:REM CONTROLS CHARACTER LOCATION
90 COLOR 0,7,4:REM BLUE SCREEN
100 COLOR 1,1:REM BLACK INK
110 COLOR 4,1:REM BLACK BORDER
120 REM**************************
130 SCNCLR:PRINT:PRINT
140 PRINT TAB(18)"WAIT"
150 RESTORE
160 FOR N=0 TO 33
170 :  READ A:POKE MC+N,A
180 NEXT:REM LOAD MACHINE CODE ROUTINE
190 SYS MC:REM TRANSFER CHARACTER SET
200 R1=PEEK(CC)
210 R2=PEEK(CC+1)
220 POKE CC,R1 AND 251:REM CHARACTERS READ FROM RAM
230 POKE CC+1,(R2 AND 3) OR 48:REM POINT TO CHARACTERS
240 REM**************************
250 FOR N=33 TO 38
260 :  FOR K=0 TO 7
270 :     READ A
280 :     POKE RA+K+N*8,A
290 :  NEXT
```

```
300 NEXT
310 REM**************************
320 SCNCLR
330 DX=1
340 X=0
350 C$=CHR$(33)+CHR$(34)+CHR$(35)
360 D$=CHR$(38)+CHR$(37)+CHR$(36)
370 S$=CHR$(32)+CHR$(32)+CHR$(32)
380 DO
390 :  IF DX=1 THEN CHAR,X,10,D$
400 :  IF DX=-1 THEN CHAR,X,10,C$
410 :  FOR N=0 TO 100:NEXT
420 :  CHAR,X,10,S$
430 :  X=X+DX
440 :  IF X>35 OR X<1 THEN DX=-DX
450 :  GET A$
460 LOOP UNTIL A$=" ":REM SPACE BAR STOPS PROGRAM
470 POKE CC,R1
480 POKE CC+1,R2
490 END
500 REM**********************
510 REM**********************
520 REM
530 REM
540 REM         **********
550 REM         *        *
560 REM         *        *
570 REM         *  DATA  *
580 REM         *        *
590 REM         *        *
600 REM         **********
610 REM
620 REM
630 REM******************************
640 REM DATA FOR MACHINE CODE ROUTINE
650 REM******************************
660 REM
670 DATA 169,0,133,3,133,5,169,48,133,4,169,208
680 DATA 133,6,160,0,177,5,145,3,200,208,249
690 DATA 230,4,230,6,169,216,197,6,208,239,96
700 REM
710 REM**************************
720 REM DATA FOR CHARACTER SHAPES
730 REM**************************
740 REM
750 DATA 0,0,1,1,19,255,127,63
760 DATA 128,128,160,160,241,249,255,255
```

```
770 DATA 0,0,128,128,128,152,255,254
780 DATA 0,0,128,128,200,255,254,252
790 DATA 1,1,5,5,143,159,255,255
800 DATA 0,0,1,1,1,25,255,127
810 REM********************
```

P43 Submarine

This program is similar to the previous one except that a submarine shape is generated and movement across the screen is controlled by the cursor control keys.

COMMANDS

Key in the program and RUN.
Press the space bar to return to the normal character set

```
10 REM SUBMARINE
20 REM *********
30 REM
40 POKE 52,47:POKE 56,47:CLR:REM RESERVE RAM
50 REM**************************
60 MC=12032:REM START OF MACHINE CODE ROUTINE
70 RA=12288:REM START OF CHARACTER RAM
80 CC=65298:REM CONTROLS CHARACTER LOCATION
90 COLOR 0,7,4:REM BLUE SCREEN
100 COLOR 1,1:REM BLACK INK
110 COLOR 4,1:REM BLACK BORDER
120 REM**************************
130 SCNCLR:PRINT:PRINT
140 PRINT TAB(18)"WAIT"
150 PRINT:PRINT:PRINT
160 PRINT TAB(6)"CURSOR CONTROL KEYS WILL MOVE"
170 PRINT TAB(6)"SUBMARINE FROM SIDE TO SIDE"
180 PRINT TAB(6)"ON THE SCREEN
190 RESTORE
200 FOR N=0 TO 33
210 :  READ A:POKE MC+N,A
220 NEXT:REM LOAD MACHINE CODE ROUTINE
230 SYS MC:REM TRANSFER CHARACTER SET
240 R1=PEEK(CC)
250 R2=PEEK(CC+1)
260 POKE CC,R1 AND 251:REM CHARACTERS READ FROM RAM
270 POKE CC+1,(R2 AND 3) OR 48:REM POINT TO CHARACTERS
280 REM**************************
290 FOR N=33 TO 38
300 :  FOR K=0 TO 7
310 :     READ A
```

```
320 :     POKE RA+K+N*8,A
330 :  NEXT
340 NEXT
350 REM**************************
360 SCNCLR
370 F=1
380 X=0
390 C$=CHR$(33)+CHR$(34)+CHR$(35)
400 D$=CHR$(38)+CHR$(37)+CHR$(36)
410 S$=CHR$(32)+CHR$(32)+CHR$(32)
420 DO
430 :  DX=0
440 :  IF ASC(A$)=157 THEN DX=-1
450 :  IF ASC(A$)=29 THEN DX=1
460 :  IF DX=-1 AND F=1 THEN F=0:DX=0
470 :  IF DX=1 AND F=0 THEN F=1:DX=0
480 :  IF X+DX>36 OR X+DX<0 THEN DX=0
490 :  IF DX<>0 THEN CHAR,X,10,S$
500 :  X=X+DX
510 :  IF F=1 THEN CHAR,X,10,D$
520 :  IF F=0 THEN CHAR,X,10,C$
530 :  GETKEY A$
540 LOOP UNTIL A$=" ":REM SPACE BAR STOPS PROGRAM
550 POKE CC,R1
560 POKE CC+1,R2
570 END
580 REM**********************
590 REM**********************
600 REM
610 REM
620 REM         **********
630 REM         *        *
640 REM         *        *
650 REM         *  DATA  *
660 REM         *        *
670 REM         *        *
680 REM         **********
690 REM
700 REM
710 REM******************************
720 REM DATA FOR MACHINE CODE ROUTINE
730 REM******************************
740 REM
750 DATA 169,0,133,3,133,5,169,48,133,4,169,208
760 DATA 133,6,160,0,177,5,145,3,200,208,249
770 DATA 230,4,230,6,169,216,197,6,208,239,96
780 REM
```

```
790 REM*************************
800 REM DATA FOR CHARACTER SHAPES
810 REM*************************
820 REM
830 DATA 0,0,0,0,3,127,255,127
840 DATA 64,64,112,112,255,255,255,255
850 DATA 0,0,0,0,0,247,255,247
860 DATA 0,0,0,0,192,254,255,254
870 DATA 2,2,14,14,255,255,255,255
880 DATA 0,0,0,0,0,239,255,239
890 REM*******************
```

P44 Runner

This program demonstrates how animation may be created by printing one shape over another. The runner is two character spaces wide and is moved forward one space at a time, giving a reasonably continuous motion.

In this program only two figure positions are used. More intermediate positions would give smoother motion.

COMMANDS

Key in the program and RUN.
Press the space bar to return to the normal character set.

```
10 REM RUNNER
20 REM ******
30 REM
40 POKE 52,47:POKE 56,47:CLR:REM RESERVE RAM
50 REM**************************
60 MC=12032:REM START OF MACHINE CODE ROUTINE
70 RA=12288:REM START OF CHARACTER RAM
80 CC=65298:REM CONTROLS CHARACTER LOCATION
90 COLOR 0,1:REM BLACK SCREEN
100 COLOR 1,2:REM WHITE INK
110 COLOR 4,7,3:REM BLUE BORDER
120 REM**************************
130 SCNCLR:PRINT:PRINT
140 PRINT TAB(18)"WAIT"
150 RESTORE
160 FOR N=0 TO 33
170 :  READ A:POKE MC+N,A
180 NEXT:REM LOAD MACHINE CODE ROUTINE
190 SYS MC:REM TRANSFER CHARACTER SET
200 R1=PEEK(CC)
210 R2=PEEK(CC+1)
220 POKE CC,R1 AND 251:REM CHARACTERS READ FROM RAM
230 POKE CC+1,(R2 AND 3) OR 48:REM POINT TO CHARACTERS
240 REM**************************
250 FOR N=33 TO 40
260 :  FOR K=0 TO 7
270 :     READ A
```

```
280 :     POKE RA+K+N*8,A
290 :  NEXT
300 NEXT
310 REM**************************
320 SCNCLR
330 X=0
340 DO
350 :  CHAR,X,10,CHR$(33)+CHR$(34)
360 :  CHAR,X,11,CHR$(35)+CHR$(36)
370 :  FOR N=0 TO 100:NEXT
380 :  CHAR,X,10,CHR$(32)+CHR$(32)
390 :  CHAR,X,11,CHR$(32)+CHR$(32)
400 :  X=X+1
410 :  CHAR,X,10,CHR$(37)+CHR$(38)
420 :  CHAR,X,11,CHR$(39)+CHR$(40)
430 :  FOR N=0 TO 100:NEXT
440 :  CHAR,X,10,CHR$(32)+CHR$(32)
450 :  CHAR,X,11,CHR$(32)+CHR$(32)
460 :  X=X+1
470 :  IF X>37 THEN X=0
480 :  GET A$
490 LOOP UNTIL A$=" ":REM SPACE BAR STOPS PROGRAM
500 POKE CC,R1
510 POKE CC+1,R2
520 END
530 REM**********************
540 REM**********************
550 REM
560 REM
570 REM         **********
580 REM         *        *
590 REM         *        *
600 REM         *  DATA  *
610 REM         *        *
620 REM         *        *
630 REM         **********
640 REM
650 REM
660 REM******************************
670 REM DATA FOR MACHINE CODE ROUTINE
680 REM******************************
690 REM
700 DATA 169,0,133,3,133,5,169,48,133,4,169,208
710 DATA 133,6,160,0,177,5,145,3,200,208,249
720 DATA 230,4,230,6,169,216,197,6,208,239,96
730 REM
740 REM**************************
```

```
750 REM DATA FOR CHARACTER SHAPES
760 REM************************
770 REM
780 DATA 0,0,0,0,1,1,1,0
790 DATA 6,14,15,14,248,112,127,240
800 DATA 1,15,48,32,32,0,0,0
810 DATA 252,4,7,0,0,0,0,0
820 DATA 0,0,0,0,0,0,0,1
830 DATA 0,0,0,0,6,14,207,238
840 DATA 1,0,7,7,118,30,1,1
850 DATA 120,240,240,220,0,0,0,192
860 REM********************
```

P45 Bat'n'moth

You've heard of bat'n'ball - well here's bat'n'moth. The bat is a composite shape made up from four user defined characters. It may be moved across and up and down the screen by the cursor control keys. A little moth flutters around the bat, always remaining just out of reach. Notice that two moth shapes are used to make the moth flutter. This program could be converted into a game by having several moths which the bat has to catch within a certain time

COMMANDS

Key in the program and RUN.
Press the space bar to return to the normal character set

```
10 REM BAT'N'MOTH
20 REM **********
30 REM
40 POKE 52,47:POKE 56,47:CLR:REM RESERVE RAM
50 REM**************************
60 MC=12032:REM START OF MACHINE CODE ROUTINE
70 RA=12288:REM START OF CHARACTER RAM
80 CC=65298:REM CONTROLS CHARACTER LOCATION
90 COLOR 0,7,3:REM BLUE SCREEN
100 COLOR 1,2:REM WHITE INK
110 COLOR 4,7,3:REM BLUE BORDER
120 REM**************************
130 SCNCLR
140 CHAR,6,4,"THE FOUR CURSOR CONTROL KEYS"
150 CHAR,6,5,"CONTROL THE BAT, BUT WE BET"
160 CHAR,6,6,"YOU CAN'T CATCH THE MOTH."
170 CHAR,11,16,"PRESS ANY KEY"
180 GETKEY A$
190 SCNCLR
200 CHAR,18,6,"WAIT"
210 RESTORE
220 FOR N=0 TO 33
230 :  READ A:POKE MC+N,A
240 NEXT:REM LOAD MACHINE CODE ROUTINE
250 SYS MC:REM TRANSFER CHARACTER SET
260 R1=PEEK(CC)
```

```
270 R2=PEEK(CC+1)
280 POKE CC,R1 AND 251:REM CHARACTERS READ FROM RAM
290 POKE CC+1,(R2 AND 3) OR 48:REM POINT TO CHARACTERS
300 REM**************************
310 FOR N=33 TO 38
320 :  FOR K=0 TO 7
330 :     READ A
340 :     POKE RA+K+N*8,A
350 :  NEXT
360 NEXT
370 REM**************************
380 SCNCLR
390 X=19
400 Y=10
410 B1$=CHR$(35)+CHR$(36)
420 B2$=CHR$(37)+CHR$(38)
430 S$=CHR$(32)+CHR$(32)
440 DO
450 :  DX=0:DY=0:F=0
460 :  IF ASC(A$)=157 THEN DX=-1
470 :  IF ASC(A$)=29 THEN DX=1
480 :  IF ASC(A$)=145 THEN DY=-1
490 :  IF ASC(A$)=17 THEN DY=1
500 :  IF X+DX>37 OR X+DX<0 THEN DX=0
510 :  IF Y+DY>21 OR Y+DY<0 THEN DY=0
520 :  IF DX<>0 OR DY<>0 THEN F=1
530 :  IF F=1 THEN CHAR,X,Y,S$:CHAR,X,Y+1,S$
540 :  X=X+DX:Y=Y+DY
550 :  COLOR 1,1
560 :  CHAR,X,Y,B1$
570 :  CHAR,X,Y+1,B2$
580 :  CHAR,M1%,M2%,CHR$(32)
590 :  DO
600 :     IF RND(1)<.5 THEN SN=1:ELSE SN=-1
610 :     M1%=X+SN*(4*RND(1)+3)
620 :     IF RND(1)<.5 THEN SN=1:ELSE SN=-1
630 :     M2%=Y+SN*(4*RND(1)+3)
640 :  LOOP UNTIL M1%>0 AND M1%<37 AND M2%>0 AND M2%<24
650 :  COLOR 1,2
660 :  CHAR,M1%,M2%,CHR$(33)
670 :  CHAR,M1%,M2%,CHR$(34)
680 :  CHAR,M1%,M2%,CHR$(33)
690 :  CHAR,M1%,M2%,CHR$(34)
700 :  CHAR,M1%,M2%,CHR$(33)
710 :  GET A$
720 LOOP UNTIL A$=" ":REM SPACE BAR STOPS PROGRAM
730 POKE CC,R1
```

```
740 POKE CC+1,R2
750 END
760 REM**********************
770 REM**********************
780 REM
790 REM
800 REM          **********
810 REM          *        *
820 REM          *        *
830 REM          *  DATA  *
840 REM          *        *
850 REM          *        *
860 REM          **********
870 REM
880 REM
890 REM********************************
900 REM DATA FOR MACHINE CODE ROUTINE
910 REM********************************
920 REM
930 DATA 169,0,133,3,133,5,169,48,133,4,169,208
940 DATA 133,6,160,0,177,5,145,3,200,208,249
950 DATA 230,4,230,6,169,216,197,6,208,239,96
960 REM
970 REM**************************
980 REM DATA FOR CHARACTER SHAPES
990 REM**************************
1000 REM
1010 DATA 129,129,195,231,255,231,195,129
1020 DATA 36,102,102,102,126,102,102,36
1030 DATA 130,130,129,195,199,195,225,227
1040 DATA 65,65,129,195,227,195,135,199
1050 DATA 231,255,255,239,231,197,137,156
1060 DATA 231,255,255,247,231,163,145,57
1070 REM********************
```

P46 Jimmy

In this program a machine code routine is read into memory and RUN. This routine transfers characters from ROM to RAM where their shape may be redefined. The user defined characters thus created are used to create the cartoon character Jimmy.

We have included a few lines in the program to draw Jimmy and make him wave. These lines are merely an example of how Jimmy can be drawn and moved. You can generate different movements using the shapes given, or expand the program further by defining new shapes.

COMMANDS

Key in the program and RUN.
Stop the program by pressing the RUN/STOP and RESTORE keys simultaneously.
Key in your own routines in lines to make Jimmy move.

```
10 REM JIMMY
20 REM *****
30 REM
40 POKE 52,47:POKE 56,47:CLR:REM RESERVE RAM
50 REM**************************
60 MC=12032:REM START OF MACHINE CODE ROUTINE
70 RA=12288:REM START OF CHARACTER RAM
80 CC=65298:REM CONTROLS CHARACTER LOCATION
90 COLOR 0,1:REM BLACK SCREEN
100 COLOR 1,2:REM WHITE INK
110 COLOR 4,7,3:REM BLUE BORDER
120 REM**************************
130 SCNCLR:PRINT:PRINT
140 PRINT TAB(18)"WAIT"
150 RESTORE
160 FOR N=0 TO 33
170 :  READ A:POKE MC+N,A
180 NEXT:REM LOAD MACHINE CODE ROUTINE
190 SYS MC:REM TRANSFER CHARACTER SET
200 R1=PEEK(CC)
210 R2=PEEK(CC+1)
```

```
220 POKE CC,R1 AND 251:REM CHARACTERS READ FROM RAM
230 POKE CC+1,(R2 AND 3) OR 48:REM POINT TO CHARACTERS
240 REM**************************
250 FOR N=33 TO 47
260 :  FOR K=0 TO 7
270 :     READ A
280 :     POKE RA+K+N*8,A
290 :  NEXT
300 NEXT
310 REM**************************
320 REM DISPLAY POKE CODES 33 TO 47
330 REM (CHR$(33) TO CHR$(47)) ARE NOW
340 REM REDEFINED TO GIVE THE SHAPES
350 REM WHICH MAKE UP JIMMY.
360 REM**************************
370 REM YOU CAN ENTER ANY PROGRAM YOU
380 REM WISH IN LINES 550 TO 4500,
390 REM USING THE SHAPES TO BUILD THE
400 REM CHARACTER.
410 REM**************************
420 REM WE HAVE PUT IN A SIMPLE ROUTINE
430 REM AS AN EXAMPLE. THIS MAKES JIMMY
440 REM WAVE.
450 REM**************************
460 REM**************************
470 REM
480 REM
490 SCNCLR
500 CHAR,15,2,"HELLO JIMMY"
510 COLOR 1,12,7:REM PINK
520 CHAR,19,10,CHR$(33):REM HEAD
530 COLOR 1,11,6:REM GREEN
540 CHAR,18,11,CHR$(38):REM UPPER ARM
550 CHAR,19,11,CHR$(34):REM UPPER TRUNK
560 CHAR,20,11,CHR$(36):REM UPPER ARM
570 CHAR,18,12,CHR$(39):REM LOWER ARM
580 CHAR,19,12,CHR$(35):REM LOWER TRUNK
590 CHAR,20,12,CHR$(37):REM LOWER ARM
600 COLOR 1,5,4:REM PURPLE
610 CHAR,19,13,CHR$(42):REM LEGS
620 COLOR 1,2:REM WHITE
630 CHAR,18,13,CHR$(44):REM FOOT
640 CHAR,20,13,CHR$(43):REM FOOT
650 REM**********************
660 REM MAKE JIMMY WAVE
670 COLOR 1,11,6:REM GREEN
680 DO
```

```
690 :  CHAR,20,11,CHR$(36):REM UPPER ARM
700 :  CHAR,20,12,CHR$(37):REM LOWER ARM
710 :  FOR N=0 TO 200:NEXT
720 :  CHAR,20,12,CHR$(32):REM SPACE
730 :  CHAR,20,10,CHR$(40):REM ARM WAVE
740 :  CHAR,20,11,CHR$(41):REM ARM WAVE
750 :  FOR N=0 TO 200:NEXT
760 :  CHAR,20,10,CHR$(32):REM SPACE
770 :  GET A$
780 LOOP UNTIL A$=" ":REM SPACE BAR STOPS PROGRAM
790 POKE CC,R1
800 POKE CC+1,R2
810 END
820 REM**********************
830 REM**********************
840 REM
850 REM
860 REM         **********
870 REM         *        *
880 REM         *        *
890 REM         *  DATA  *
900 REM         *        *
910 REM         *        *
920 REM         **********
930 REM
940 REM
950 REM******************************
960 REM DATA FOR MACHINE CODE ROUTINE
970 REM******************************
980 REM
990 DATA 169,0,133,3,133,5,169,48,133,4,169,208
1000 DATA 133,6,160,0,177,5,145,3,200,208,249
1010 DATA 230,4,230,6,169,216,197,6,208,239,96
1020 REM
1030 REM*************************
1040 REM DATA FOR CHARACTER SHAPES
1050 REM*************************
1060 REM
1070 DATA 24,60,90,126,126,102,62,24
1080 REM HEAD
1090 REM********************
1100 DATA 24,255,255,255,231,126,102,126
1110 REM UPPER TRUNK
1120 REM********************
1130 DATA 102,60,60,60,126,231,231,231
1140 REM LOWER TRUNK
1150 REM********************
```

```
1160 DATA 0,128,192,224,112,48,48,48
1170 REM*************************
1180 REM UPPER LEFT ARM
1190 REM*********************
1200 DATA 48,48,48,0,0,0,0,0
1210 REM LOWER LEFT ARM
1220 REM*********************
1230 DATA 0,1,3,7,14,12,12,12
1240 REM UPPER RIGHT ARM
1250 REM*********************
1260 DATA 12,12,12,0,0,0,0,0
1270 REM LOWER RIGHT ARM
1280 REM*********************
1290 DATA 0,0,0,48,48,48,48,48
1300 REM RAISE UPPER LEFT ARM
1310 REM*********************
1320 DATA 112,240,192,128,0,0,0,0
1330 REM RAISE LOWER LEFT ARM
1340 REM*********************
1350 DATA 231,231,231,231,231,231,231,231
1360 REM LEGS AT ATTENTION
1370 REM*********************
1380 DATA 0,0,0,0,0,0,192,192
1390 REM LEFT FOOT
1400 REM*********************
1410 DATA 0,0,0,0,0,0,3,3
1420 REM RIGHT FOOT
1430 REM*********************
1440 DATA 0,0,0,36,60,60,255,255
1450 REM HAT
1460 REM*********************
1470 DATA 0,24,60,126,126,60,24,0
1480 REM BALL
1490 REM*********************
1500 DATA 60,126,255,126,60,24,24,24
1510 REM BAT
1520 REM*********************
1530 REM*********************
```

P47 Worm

In this program pixels are inked in at the front of a squiggly shape and deleted behind it. As a result a little worm slithers across the screen. Multicolor high resolution mode is used.

COMMANDS

Key in the program and RUN.
Stop the program by pressing any key. This puts the machine back into text mode.

```
10 REM WORM
20 REM ****
30 REM
40 COLOR 0,1:REM BLACK SCREEN
50 COLOR 1,2:REM COLOR 1 WHITE
60 COLOR 2,16,2:REM COLOR 2 GREEN
70 COLOR 4,7:REM BLUE BORDER
80 GRAPHIC 3,1:REM MULTICOLOR HI-RES MODE
90 DO
100 :  FOR N=0 TO 199
110 :     X=N:IF X>159 THEN X=159
120 :     R=N-40:IF R<0 THEN R=0
130 :     Y=INT(90+10*SIN(X/4))
140 :     W=INT(90+10*SIN(R/4))
150 :     C=2+(N/2=INT(N/2))
160 :     DRAW C,X,Y
170 :     DRAW 0,R,W
180 :     GET A$
190 :     IF LEN(A$)>0 THEN N=199:F=1:ELSE F=0
200 :  NEXT
210 LOOP UNTIL F=1
220 GRAPHIC 0,1
230 REM PRESS ANY KEY TO RETURN TO TEXT MODE
240 END
```

P48 Colors

This program is used to exhibit the colors available on the Commodore 16. The program first of all asks for your name in order to have some text to print out. The text is the printed using all the colors available.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM COLORS
20 REM THIS PROGRAM SHOWS THE RANGE OF
30 REM COLORS AVAILABLE WITH THE C-16
40 :
50 SCNCLR
60 CHAR ,10,10,"COLORS"
70 FOR A=1 TO 200:NEXT A
80 SCNCLR
90 CHAR ,2,10,"ENTER YOUR NAME"
100 INPUT N$
110 N$=N$+"..."
120 SCNCLR
130 CHAR ,5,20,"C O L O R S  A V A I L A B L E"
140 PRINT "[CH CD CD CD CD CD CR CR CR CR CR CR CR CR CR CR
CR CR ]"+CHR$(27)+"T"+"[CD CD CD CD CD CR CR CR CR CR CR CR
CR CR CR CR CR CR CR ]"+CHR$(27)+"B"
150 REM WE HAVE SET UP A WINDOW TO SHOW
160 REM TEXT. WE CAN NOW PLAY WITH COLOR
170 REM AND INTENSITY
180 B=INT(RND(1)*16+1)
190 F=INT(RND(1)*16+1)
200 BR=INT(RND(1)*16+1)
210 DO
220 :  I1=INT(RND(1)*8)
230 :  I2=INT(RND(1)*8)
240 :  I3=INT(RND(1)*8)
250 :  COLOR 0,B,I1
260 :  COLOR 4,BR,I3
270 :  B=B+1:IF B=17 THEN B=1
280 :  DO
```

```
290 :       COLOR 1,F,I2
300 :       PRINT N$;
310 :       F=F+1:IF F=17 THEN F=1
320 :       FOR A=1 TO 100:NEXT A
330 :   LOOP UNTIL F=1
340 :   BR=BR+1:IF BR=17 THEN BR=1
350 LOOP UNTIL 0
```

P49 Shading

This program uses the PLOT command to draw a line by placing a series of dots along it. The density of the dots gives a measure of the illumination of the line.

The points on the line are calculated by using the formula:

$$Y=M*X+C.$$

This formula leads to the following rules for calculating M and C.

$$M=(Y2-Y1)/(X2-X1)$$
$$C=Y2-M*X2$$

where X1,Y1 and X2,Y2 are two points on the line. These rules fall down when the line is vertical and the program has a small adjustment when this situation arises.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM -SHADING
20 SCNCLR
30 PRINT"[CD CD CD CD CD CD CD CD ]"
40 PRINT "                SHADING"
50 A=TI
60 IF TI<A+150 THEN 60
70 SCNCLR
80 PRINT"THIS PROGRAM USES THE PLOT COMMAND"
90 PRINT "TO SHOW THE EFFECT OF 'SHADING' A LINE"
100 PRINT "THE EFFECT OF SHADING IS ACHIEVED BY "
110 PRINT "BY PLACING A SERIES OF DOTS ALONG THE "
120 PRINT "LINE, THE NUMBER OF DOTS CORRESPONDING"
130 PRINT "TO THE ILLUMINATION OF THE LINE."
```

```
140 PRINT "THE ILLUMINATION CONSTANT LIES BETWEEN"
150 PRINT ".05 AND .9"
160 PRINT
170 INPUT "ENTER FIRST POINT X1,Y1";X1,Y1
180 IF X1<0 OR X1>319 OR Y1<0 OR Y1>199 THEN PRINT "INVALID":
GOTO 170
190 INPUT "ENTER SECOND POINT X2,Y2";X2,Y2
200 IF X2<0 OR X2>319 OR Y2<0 OR Y2>199 THEN PRINT "INVALID":
GOTO 190
210 INPUT "ENTER ILLUMINATION CONSTANT";I
220 IF I<.05 OR I>.9 THEN PRINT "INVALID":GOTO 210
230 GRAPHIC 1,1
240 GOSUB 310
250 GETKEY A$
260 GRAPHIC 0,0
270 END
280 REM SHADE LINE SUBROUTINE
290 REM LINE IS DRAWN FROM X1,Y1 TO X2,Y2
300 REM WITH ILLUMINATION CONSTANT I
310 IF X2=X1 THEN GOTO 430
320 M=(Y2-Y1)/(X2-X1)
330 C=Y1-M*X1
340 D=SQR((X1-X2)^2+(Y2-Y1)^2)
350 N=D*I
360 DX=(X2-X1)/N
370 FOR K=1 TO N
380 J=INT(X1+K*DX)
390 L=INT(M*J+C)
400 DRAW 1,J,L
410 NEXT K
420 RETURN
430 REM TAKE CARE OF VERTICAL LINES
440 D=ABS(Y2-Y1)
450 N=D*I
460 IF Y2<Y1 THEN Y2=T:Y2=Y1:Y1=T
470 DY=(Y2-Y1)/N
480 FOR K=1 TO N
490 J=INT(Y1+K*DY)
500 L=INT(X1)
510 DRAW  ,L,J
520 NEXT K
530 RETURN
```

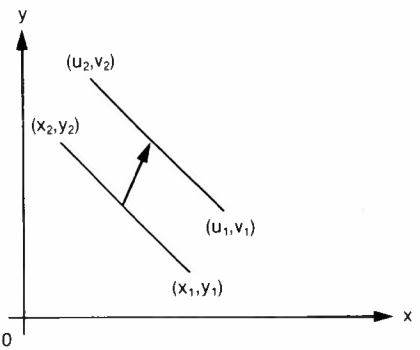
P50 Translation

To translate, or to move a line, we must compute the new end points of the line and then draw it.

If we can move a single line, then we have the capability of moving line drawings about the screen.

To move a line we must know the end points of the original line and the distance to be moved in both the X and Y directions.

The program uses the shading routine developed earlier to redraw the line.



COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM -TRANSLATION
20 SCNCLR
30 PRINT"[CD CD CD CD CD CD CD CD ]"
40 PRINT "              TRANSLATION"
50 A=TI
60 IF TI<A+150 THEN 60
70 SCNCLR
80 PRINT"THIS PROGRAM INTRODUCES A SIMPLE "
90 PRINT "FOR CHANGING THE POSITION OF A LINE. IF"
```

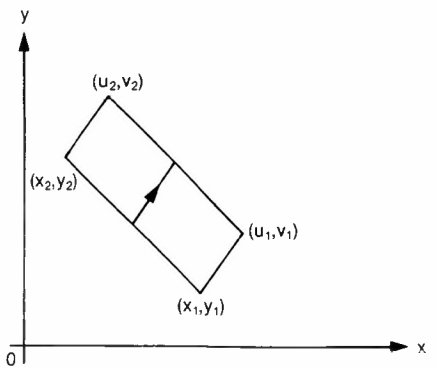
```
100 PRINT "YOU HAVE A ROUTINE FOR MOVING SINGLE"
110 PRINT "LINES THEN YOU CAN MOVE COMPLETE LINE"
120 PRINT "DRAWINGS ABOUT THE SCREEN AT YOUR OWN"
130 PRINT "CONVENIENCE.[CD CD CD ]"
140 PRINT "TO MOVE A LINE WE MUST KNOW ITS ORIGINAL";
150 PRINT"POSITION AND THE DISTANCE TO BE MOVED."
160 PRINT "IN BOTH THE X AND Y DIRECTIONS."
170 INPUT "ENTER FIRST POINT X1,Y1";X1,Y1
180 IF X1<0 OR X1>319 OR Y1<0 OR Y1>199 THEN PRINT "INVALID":
GOTO 170
190 INPUT "ENTER SECOND POINT X2,Y2";X2,Y2
200 IF X2<0 OR X2>319 OR Y2<0 OR Y2>199 THEN PRINT "INVALID":
GOTO 190
210 INPUT "ENTER ILLUMINATION CONSTANT";I
220 IF I<.05 OR I>.9 THEN PRINT "INVALID":GOTO 210
230 INPUT "ENTER X MOVEMENT";K1
240 INPUT "ENTER Y MOVEMENT";K2
250 GRAPHIC 1,1
260 GOSUB 340
270 GOSUB 620
280 GETKEY A$
290 GRAPHIC 0,0
300 END
310 REM SHADE LINE SUBROUTINE
320 REM LINE IS DRAWN FROM X1,Y1 TO X2,Y2
330 REM WITH ILLUMINATION CONSTANT I
340 IF X2=X1 THEN GOTO 460
350 M=(Y2-Y1)/(X2-X1)
360 C=Y1-M*X1
370 D=SQR((X1-X2)^2+(Y2-Y1)^2)
380 N=D*I
390 DX=(X2-X1)/N
400 FOR K=1 TO N
410 J=INT(X1+K*DX)
420 L=INT(M*J+C)
430 DRAW 1,J,L
440 NEXT K
450 RETURN
460 REM TAKE CARE OF VERTICAL LINES
470 D=ABS(Y2-Y1)
480 N=D*I
490 IF Y2<Y1 THEN Y2=T:Y2=Y1:Y1=T
500 DY=(Y2-Y1)/N
510 FOR K=1 TO N
520 J=INT(Y1+K*DY)
530 L=INT(X1)
540 DRAW 1,L,J
```

```
550 NEXT K
560 RETURN
570 END
580 REM TRANSLATION SUBROUTINE
590 REM LINE FROM X1,Y1 TO X2,Y2 IS
600 REM MOVED K1 IN X-DIRECTION
610 REM AND K2 IN THE Y-DIRECTION
620 X1=X1+K1:Y1=Y1+K2
630 X2=X2+K1:Y2=Y2+K2
640 GOSUB 340
650 RETURN
```

P51 Parallelogram

Using the routines developed in the previous programs we can shade a parallelogram.

The parallelogram is drawn by taking a vector (a straight line) and moving it to a new position, drawing many intermediate lines between the starting and finishing vectors. As before we use the shading routine to mimic illumination.



COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM -PARALLELOGRAM
20 SCNCLR
30 PRINT"[CD CD CD CD CD CD CD CD ]"
40 PRINT "              PARALLELOGRAM"
50 A=TI
60 IF TI<A+150 THEN 60
70 SCNCLR
80 PRINT"THIS PROGRAM USES THE ROUTINES DEVELOPED";
90 PRINT "IN THE PREVIOUS PROGRAMS TO SHADE A"
100 PRINT "PARALLELOGRAM. AS POINTED OUT IN THE "
110 PRINT "DESCRIPTION, WE CAN NOW SPECIFY"
120 PRINT "ILLUMINATION IN TWO DIRECTIONS TO "
130 PRINT "PRODUCE VARIABLE SHADING.[CD CD ]"
```

```
140 PRINT "TO DRAW THE PARALLELOGRAM WE NEED TO KNOW"
150 PRINT"THE GENERATING VECTOR POSITION AND THE"
160 PRINT "DISTANCE TO BE MOVED IN BOTH THE X AND"
170 PRINT "Y DIRECTIONS."
180 INPUT "ENTER FIRST POINT X1,Y1";X1,Y1
190 IF X1<0 OR X1>319 OR Y1<0 OR Y1>199 THEN PRINT "INVALID":
GOTO 180
200 INPUT "ENTER SECOND POINT X2,Y2";X2,Y2
210 IF X2<0 OR X2>319 OR Y2<0 OR Y2>199 THEN PRINT "INVALID":
GOTO 200
220 INPUT "ENTER X-ILLUMINATION CONSTANT";I1
230 IF I1<.05 OR I1>.9 THEN PRINT "INVALID":GOTO 220
240 INPUT "ENTER Y-ILLUMINATION CONSTANT";I2
250 IF I2<.05 OR I2>.9 THEN PRINT "INVALID":GOTO 240
260 INPUT "ENTER X MOVEMENT";R1
270 INPUT "ENTER Y MOVEMENT";R2
280 GRAPHIC 1,1
290 GOSUB 670
300 GETKEY A$
310 GRAPHIC 0,0
320 END
330 REM SHADE LINE SUBROUTINE
340 IF X2=X1 THEN GOTO 460
350 M=(Y2-Y1)/(X2-X1)
360 C=Y1-M*X1
370 D=SQR((X1-X2)^2+(Y2-Y1)^2)
380 N=D*I
390 DX=(X2-X1)/N
400 FOR K=1 TO N
410 J=INT(X1+K*DX)
420 L=INT(M*J+C)
430 DRAW 1,J,L
440 NEXT K
450 RETURN
460 REM TAKE CARE OF VERTICAL LINES
470 D=ABS(Y2-Y1)
480 N=D*I
490 IF Y2<Y1 THEN Y2=T:Y2=Y1:Y1=T
500 DY=(Y2-Y1)/N
510 FOR K=1 TO N
520 J=INT(Y1+K*DY)
530 L=INT(X1)
540 DRAW 1,L,J
550 NEXT K
560 RETURN
570 END
580 REM TRANSLATION SUBROUTINE
```

```
590 REM LINE FROM X1,Y1 TO X2,Y2 IS
600 REM MOVED K1 IN X-DIRECTION
610 REM AND K2 IN THE Y-DIRECTION
620 X1=X1+K1:Y1=Y1+K2
630 X2=X2+K1:Y2=Y2+K2
640 GOSUB 340
650 RETURN
660 REM PARALLELOGRAM SHADING ROUTINE
670 I=I1
680 GOSUB 340:REM DRAW VECTOR
690 LN=SQR(R1*R1+R2*R2)
700 M=LN*I2
710 D1R=R1/M
720 D2R=R2/M
730 FOR P=1 TO M
740 K1=D1R
750 K2=D2R
760 I=I1
770 GOSUB 620
780 NEXT P
790 RETURN
```

P52 Shape grabber

When dealing with line drawings one needs to know only the end points of each line and the method of drawing the lines. The end points of the lines can be held in a two dimensional array, and there are various methods of drawing the lines. In this program the array is set up in such a way that the lines are drawn in order.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM - SHAPE GRABBER
20 PRINT "[CS CD CD CD CD CD CD CD CD CD ]"
30 PRINT "             SHAPE GRABBER"
40 A=TI
50 IF TI<A+150 THEN GOTO 50
60 PRINT "[CS ]"
70 PRINT "THIS PROGRAM IS USED TO DRAW SHAPES"
80 PRINT "IN HI RES GRAPHICS. THE SHAPE IS"
90 PRINT "RECORDED AS AN ARRAY OF POINTS, WHICH"
100 PRINT "WHEN JOINED TOGETHER FORM A LINE DRAWING";
110 PRINT "[CD ]NOTE THAT THE ORDER IN WHICH THE POINTS"
120 PRINT"ARE DRAWN IS IMPORTANT"
130 PRINT "[CD ]PRESS ANY KEY TO CONTINUE"
140 GET A$:IF A$="" THEN GOTO 140
170 PRINT "[CS ]   SHAPE DRAWING ROUTINE"
180 PRINT "[CD CD ]ENTER NUMBER OF POINTS IN SHAPE"
190 INPUT N
200 DIM S(2,N)
210 PRINT "ENTER THE COORDINATES OF THE POINTS"
220 PRINT "IN THE SHAPE"
230 FOR I=1 TO N
240 PRINT "POINT ";I;" X,Y";
250 INPUT S(1,I),S(2,I)
260 NEXT I
```

```
270 PRINT "[CS ]THE POINTS ARE:"
280 L=1
290 FOR I=1 TO N
300 PRINT S(1,I),S(2,I)
310 L=L+1
320 IF L<20 THEN GOTO 360
330 PRINT "PRESS ANY KEY TO CONTINUE"
340 GET A$:IF A$="" THEN GOTO 340
350 L=0
360 NEXT I
370 PRINT "PRESS ANY KEY TO SEE DRAWING"
380 PRINT "ANOTHER KEY TO RELEASE DRAWING"
390 GET A$:IF A$="" THEN GOTO 390
400 GRAPHIC 1,1
430 X=INT(S(1,1)):Y=INT(S(2,1))
440 DRAW ,X,Y
450 FOR I=2 TO N
460 X=INT(S(1,I)):Y=INT(S(2,I))
470 DRAW TO X,Y
480 NEXT I
490 X=INT(S(1,1)):Y=INT(S(2,1))
500 DRAW TO X,Y
510 GET A$:IF A$="" THEN GOTO 510
520 GRAPHIC 0,0
530 END
```

P53 Rotation

This program uses the shape grabbing routine to get a user defined shape. The shape is then rotated through an angle of PI/2 radians, the screen is cleared and the new view is shown.

The program shows the advantage of using arrays to process shapes. The rotation problem is reduced to that of matrix (array) multiplication.

COMMANDS

Key in the program and RUN.
Enter coordinates when prompted.
Press any key to rotate figure.

```
10 REM PROGRAM - ROTATION
20 PRINT "[CS CD CD CD CD CD CD CD CD CD ]"
30 PRINT "            ROTATION"
40 A=TI
50 IF TI<A+150 THEN GOTO 50
60 PRINT "[CS ]"
70 PRINT "THIS PROGRAM FIRST OF ALL ALLOWS THE"
80 PRINT "USER TO ENTER A SHAPE, THE SHAPE IS"
90 PRINT "THEN ROTATED ABOUT THE ORIGIN THROUGH"
100 PRINT "90 DEGREES. THE ROTATION CAN BE REPEATED";
110 PRINT "[CD ]PRESS ANY KEY TO CONTINUE"
120 GET A$:IF A$="" THEN GOTO 120
130 GOSUB 370:REM GET SHAPE
140 DIM NS(2,N)
150 GOSUB 250:REM DRAW SHAPE
160 REM LOOP BACK POSITION
170 GOSUB 640:REM COMPUTE NEW SHAPE
180 FOR I=1 TO N
190 S(1,I)=NS(1,I):S(2,I)=NS(2,I)
200 NEXT I
210 GOSUB 250:REM DRAW NEW SHAPE
220 GOTO 170
230 END
240 REM SHAPE DRAWING ROUTINE
```

```
250 GET A$:IF A$="" THEN GOTO 250
260 GRAPHIC 1,1
270 X=INT(S(1,1)+159):Y=INT(S(2,1)+100)
280 DRAW 1,X,Y
290 FOR I=2 TO N
300 X=INT(S(1,I)+159):Y=INT(S(2,I)+100)
310 DRAW TO X,Y
320 NEXT I
330 X=INT(S(1,1)+159):Y=INT(S(2,1)+100)
340 DRAW TO X,Y
350 RETURN
360 REM SHAPE GRABBING ROUTINE
370 PRINT "[CS ]   SHAPE GRABBING ROUTINE"
380 PRINT "[CD CD ]ENTER NUMBER OF POINTS IN SHAPE"
390 INPUT N
400 DIM S(2,N)
410 PRINT "ENTER THE COORDINATES OF THE POINTS"
420 PRINT "IN THE SHAPE"
430 FOR I=1 TO N
440 PRINT "POINT ";I;" X,Y";
450 INPUT S(1,I),S(2,I)
460 NEXT I
470 PRINT "[CS ]THE POINTS ARE:"
480 L=1
490 FOR I=1 TO N
500 PRINT S(1,I),S(2,I)
510 L=L+1
520 IF L<20 THEN GOTO 560
530 PRINT "PRESS ANY KEY TO CONTINUE"
540 GET A$:IF A$="" THEN GOTO 540
550 L=0
560 NEXT I
570 PRINT "PRESS ANY KEY TO SEE DRAWING"
580 PRINT "ANY OTHER KEY TO ROTATE
590 GET A$:IF A$="" THEN 590
600 RETURN
610 REM ROTATION SUBROUTINE
620 REM OLD SHAPE HELD IN S(2,N)
630 REM NEW SHAPE HELD IN NS(2,N)
640 FOR I=1 TO N
650 NS(1,I)=-S(2,I)
660 NS(2,I)=S(1,I)
670 NEXT I
680 RETURN
```

P54 Transformations

This program uses the shape grabbing routine to allow the user to enter a line drawing, and this is then displayed on the screen.

The user can then use the program to perform various transformations on the line drawing.

The program informs the user of the options available.

COMMANDS

Key in the program and RUN.
Follow instructions.

This program first of all allows the user to enter a shape. The user then choses which transformation to put the shape through. The choices are :

I-Identity	R-Reflection in y=x
H-Half turn	X-Reflection in x-axis
Q-Quarter turn	Y-Reflection in y-axis
	B-Back quarter turn

Note that a single letter suffices.

```
10 REM PROGRAM - TRANSFORMATIONS
20 GOSUB 360:REM GET SHAPE
30 DIM NS(2,N),T(2,2)
40 REM
50 REM LOOP BACK POSITION
60 GET A$
70 IF A$="I" THEN T(1,1)=1:T(1,2)=0:T(2,1)=0:T(2,2)=1
80 IF A$="R" THEN T(1,1)=0:T(1,2)=1:T(2,1)=1:T(2,2)=0
90 IF A$="H" THEN T(1,1)=-1:T(1,2)=0:T(2,1)=0:T(2,2)=-1
100 IF A$="X" THEN T(1,1)=1:T(1,2)=0:T(2,1)=0:T(2,2)=-1
110 IF A$="Q" THEN T(1,1)=0:T(1,2)=-1:T(2,1)=1:T(2,2)=0
120 IF A$="Y" THEN T(1,1)=-1:T(1,2)=0:T(2,1)=0:T(2,2)=1
130 IF A$="B" THEN T(1,1)=0:T(1,2)=1:T(2,1)=-1:T(2,2)=0
140 A=A$<>"I"ANDA$<>"R"ANDA$<>"H"ANDA$<>"X"ANDA$<>"Q"
ANDA$<>"Y"ANDA$<>"B"
150 IF A THEN 50
```

```
160 GOSUB 640:REM COMPUTE NEW SHAPE
170 FOR I=1 TO N
180 S(1,I)=NS(1,I):S(2,I)=NS(2,I)
190 NEXT I
200 GOSUB 240:REM DRAW NEW SHAPE
210 GOTO 50
220 END
230 REM SHAPE DRAWING ROUTINE
240 REM
250 GRAPHIC 1,1
260 X=INT(S(1,1)+159):Y=INT(S(2,1)+100)
270 DRAW 1,X,Y
280 FOR I=2 TO N
290 X=INT(S(1,I)+159):Y=INT(S(2,I)+100)
300 DRAW TO X,Y
310 NEXT I
320 X=INT(S(1,1)+159):Y=INT(S(2,1)+100)
330 DRAW TO X,Y
340 RETURN
350 REM SHAPE GRABBING ROUTINE
360 PRINT "[CS ]   SHAPE GRABBING ROUTINE"
370 PRINT "[CD CD ]ENTER NUMBER OF POINTS IN SHAPE"
380 INPUT N
390 DIM S(2,N)
400 PRINT "ENTER THE COORDINATES OF THE POINTS"
410 PRINT "IN THE SHAPE"
420 FOR I=1 TO N
430 PRINT "POINT ";I;" X,Y";
440 INPUT S(1,I),S(2,I)
450 NEXT I
460 PRINT "[CS ]THE POINTS ARE:"
470 L=1
480 FOR I=1 TO N
490 PRINT S(1,I),S(2,I)
500 L=L+1
510 IF L<20 THEN GOTO 550
520 PRINT "PRESS ANY KEY TO CONTINUE"
530 GET A$:IF A$="" THEN GOTO 530
540 L=0
550 NEXT I
560 PRINT "PRESS ANY KEY TO SEE DRAWING"
570 PRINT "ANY OTHER KEY TO ROTATE
580 GET A$:IF A$="" THEN 580
590 GRAPHIC 1,1
600 RETURN
610 REM ROTATION SUBROUTINE
620 REM OLD SHAPE HELD IN S(2,N)
```

```
630 REM NEW SHAPE HELD IN NS(2,N)
640 FOR I=1 TO N
650 NS(1,I)=T(1,1)*S(1,I)+T(1,2)*S(2,I)
660 NS(2,I)=T(2,1)*S(1,I)+T(2,2)*S(2,I)
670 NEXT I
680 RETURN
```

P55 General transformation

This program allows the user to enter a shape and then to rotate it round the origin.

COMMANDS

Key in the program and RUN.

This program first of all allows the user to enter a shape, then to enter the angle of rotation of the transformation to be executed. The user then has the opportunity of Repeating the rotation (R) or Quitting (Q). The user presses the appropriate key as required.

```
10 REM PROGRAM - GENERAL TRANSFORMATION
20 GOSUB 270:REM GET SHAPE
30 DIM NS%(2,N),T%(2,2)
40 GOSUB 180:REM DRAW SHAPE
50 REM LOOP BACK POSITION
60 GETKEY A$
70 IF A$="Q" THEN GOTO 150
80 IF A$<>"R" THEN GOTO 50
90 GOSUB 560:REM COMPUTE NEW SHAPE
100 FOR I=1 TO N
110 S%(1,I)=NS%(1,I):S%(2,I)=NS%(2,I)
120 NEXT I
130 GOSUB 180:REM DRAW NEW SHAPE
140 GOTO 50
150 GRAPHIC 0,0
160 END
170 REM SHAPE DRAWING ROUTINE
180 REM SHAPE HELD IN ARRAY S(2,N)
190 GRAPHIC 1,1
200 DRAW 1,S%(1,1)+159,S%(2,1)+100
210 FOR I=2 TO N
220 DRAW TO S%(1,I)+159,S%(2,I)+100
230 NEXT I
240 DRAW TO S%(1,1)+159,S%(2,1)+100
250 RETURN
260 REM SHAPE GRABBING ROUTINE
270 PRINT "[CS ]   SHAPE GRABBING ROUTINE"
```

```
280 PRINT "[CD CD ]ENTER NUMBER OF POINTS IN SHAPE"
290 INPUT N
300 DIM S%(2,N)
310 PRINT "ENTER THE COORDINATES OF THE POINTS"
320 PRINT "IN THE SHAPE"
330 FOR I=1 TO N
340 PRINT "POINT ";I;" X,Y";
350 INPUT S%(1,I),S%(2,I)
360 NEXT I
370 PRINT "[CS ]THE POINTS ARE:"
380 L=1
390 FOR I=1 TO N
400 PRINT S%(1,I),S%(2,I)
410 L=L+1
420 IF L<20 THEN GOTO 460
430 PRINT "PRESS ANY KEY TO CONTINUE"
440 GETKEY A$
450 L=0
460 NEXT I
470 PRINT "ENTER ROTATION ANGLE (RADIANS)"
480 INPUT TH
490 PRINT "PRESS ANY KEY TO SEE DRAWING"
500 GETKEY A$
510 RETURN
520 REM ROTATION SUBROUTINE
530 REM OLD SHAPE HELD IN S(2,N)
540 REM NEW SHAPE HELD IN NS(2,N)
550 REM ANGLE OF ROTATION TH
560 S=SIN(TH):C=COS(TH)
570 FOR I=1 TO N
580 NS%(1,I)=C*S%(1,I)-S*S%(2,I)
590 NS%(2,I)=S*S%(1,I)+C*S%(2,I)
600 NEXT I
610 RETURN
```

P56 3D Rotation

The object of this program is to rotate a two dimensional shape in the plane of the screen.

The shape chosen is a regular polygon. (Note that in the limit the polygon becomes a circle.)

The polygon is centered at the point XO,YO,ZO and has NS sides.

The program uses the ideas of the rotation program presented earlier.

The polygon has a maximum of 10 sides. When this number of sides is chosen, we almost have a circle; so in the limit the routine will form a view of a sphere.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM - 3D ROTATION
20 PRINT "[CS ]"
30 PRINT "[CS CD CD CD CD CD ]"
40 DIM S%(3,10),NS%(3,10),C(3,3)
50 INPUT "ENTER POLYGON RADIUS";R
60 INPUT "ENTER NUMBER OF SIDES";N
70 INPUT "ENTER Y-AXIS ROTATION (DEGS)";B
80 B=B*[PI ]/180
90 TH=B
100 GOSUB 200:REM ROTATION-X MATRIX CALCULATION
110 GOSUB 310:REM SHAPE ROUTINE
120 GRAPHIC 1,1
130 GOSUB 410:REM DRAW SHAPE
140 FOR TH=B TO 2*[PI ] STEP B
150 GOSUB 490:REM CALCULATE NEW VIEW
160 GOSUB 410:REM DRAW SHAPE
170 NEXT TH
180 END
```

```
190 REM X-ROTATION MATRIX
200 C(1,1)=1
210 C(1,2)=0
220 C(1,3)=0
230 C(2,1)=0
240 C(2,2)=COS(TH)
250 C(2,3)=SIN(TH)
260 C(3,1)=0
270 C(3,2)=-SIN(TH)
280 C(3,3)=COS(TH)
290 RETURN
300 REM SHAPE SUBROUTINE
310 DA=2*[PI ]/N
320 A=-DA
330 FOR I=1 TO N
340 A=A+DA
350 S%(1,I)=R*COS(A)
360 S%(2,I)=R*SIN(A)
370 S%(3,I)=0
380 NEXT I
390 RETURN
400 REM DRAW SHAPE ROUTINE
410 REM
420 DRAW 1,S%(1,1)+160,S%(2,1)+100
430 FOR I=2 TO N
440 DRAW TO S%(1,I)+160,S%(2,I)+100
450 NEXT I
460 DRAW TO S%(1,1)+160,S%(2,1)+100
470 RETURN
480 REM NEW VIEW CALCULATION
490 FOR I=1 TO N
500 NS%(1,I)=S%(1,I)*C(1,1)+S%(2,I)*C(2,1)+S%(3,I)*C(3,1)
510 NS%(2,I)=S%(1,I)*C(1,2)+S%(2,I)*C(2,2)+S%(3,I)*C(3,2)
520 NS%(3,I)=S%(1,I)*C(1,3)+S%(2,I)*C(2,3)+S%(3,I)*C(3,3)
530 NEXT I
540 FOR J=1 TO N
550 FOR K=1 TO 3
560 S%(K,J)=NS%(K,J)
570 NEXT K
580 NEXT J
590 RETURN
```

P57 Perspective

Most people will remember from school days about perspective. The method usually remembered is to locate a vanishing point. All parallel lines should converge to that point. We use a slight variation on this idea in this program.

In this routine we imagine that we have a fixed view point, the point (0,0,0) say, and we calculate the projection of the three dimensional object on an image plane, Z=K say. We will then have a two dimensional representation of the three dimensional object which will be in perspective.

Normally we do not wish to view from the point (0,0,0) and with an image plane at Z=K. We will normally have an arbitrary view point (VX,VY,VZ) and use the plane Z=0 as the image plane.

The algorithm to find the co-ordinates of an image point is then:

1. Rewrite the co-ordinates of the point with respect to the view point.
2. Calculate the co-ordinates of the projection in the plane Z=-VZ.
3. Rewrite the co-ordinates with respect to the old co-ordinates.
4. Draw the shape with the new co-ordinates.

This program uses the above routine to implement a perspective algorithm.

COMMANDS

Key in the program and RUN.

```
10 REM PROGRAM - PERSPECTIVE
20 DIM H%(10,3),S%(10,3),P%(10,3)
30 FOR I=1 TO 10
40 FOR J=1 TO 3
50 READ H%(I,J):S%(I,J)=H%(I,J)
```

```
60 NEXT J
70 NEXT I
80 REM DATA FOR HOUSE
90 DATA 0,0,-100,50,0,-100,50,-40,-100,25,-50
100 DATA -100,0,-40,-100,0,0,-25,50 ,0,-25,50
110 DATA -40,-25,25,-50 ,-25,0,-40,-25
120 REM END OF DATA
130 GOSUB 390:REM DRAW HOUSE
140 REM LOOP BACK POINT
150 GETKEY A$
160 GRAPHIC 0,0
170 INPUT "VIEW POINT (X,Y,Z)";VX,VY,VZ
180 IF VZ=-100 THEN PRINT "CANNOT SEE SHAPE FROM INSIDE
WALL":GOTO 170
190 REM PERSPECTIVE ROUTINE
200 FOR I=1 TO 10
210 PX=S%(I,1)-VX
220 PY=S%(I,2)-VY
230 PZ=S%(I,3)-VZ
240 R=-VZ/PZ
250 QX=R*PX+VX
260 QY=R*PY+VY
270 P%(I,1)=QX
280 P%(I,2)=QY
290 P%(I,3)=0
300 NEXT I
310 FOR I=1 TO 10
320 FOR J=1 TO 3
330 H%(I,J)=P%(I,J)
340 NEXT J
350 NEXT I
360 GOSUB 390:REM DRAW HOUSE
370 GOTO 140
380 REM DRAW HOUSE
390 GRAPHIC 1,1
400 K=1:GOSUB 800
410 DRAW 1,X,Y
420 FOR J=2 TO 5
430 K=J:GOSUB 800
440 DRAW TO X,Y
450 NEXT J
460 K=1:GOSUB 800
470 DRAW TO X,Y
480 REM NEXT THE BACK OF THE HOUSE
490 K=6:GOSUB 800
500 DRAW 1,X,Y
510 FOR J=7 TO 10
```

```
520 K=J:GOSUB 800
530 DRAW TO X,Y
540 NEXT J
550 K=6:GOSUB 800
560 DRAW TO X,Y
570 REM NEXT JOIN THE BACK TO FRONT
580 K=6:GOSUB 800
590 DRAW 1,X,Y
600 K=1:GOSUB 800
610 DRAW TO X,Y
620 K=10:GOSUB 800
630 DRAW 1,X,Y
640 K=5:GOSUB 800
650 DRAW TO X,Y
660 K=9:GOSUB 800
670 DRAW 1,X,Y
680 K=4:GOSUB 800
690 DRAW TO X,Y
700 K=8:GOSUB 800
710 DRAW 1,X,Y
720 K=3:GOSUB 800
730 DRAW TO X,Y
740 K=7:GOSUB 800
750 DRAW 1,X,Y
760 K=2:GOSUB 800
770 DRAW TO X,Y
780 RETURN
790 REM CLIPPING ROUTINE
800 X=INT(H%(K,1)+160)
810 IF X>319 THEN X=319
820 IF X<0 THEN X=0
830 Y=INT(H%(K,2)+100)
840 IF Y>199 THEN Y=199
850 IF Y<0 THEN Y=0
860 RETURN
```

P58 Rotating house

Using the techniques developed in the previous programs, we present here a program which shows an object continuously rotating about the origin. The object chosen is a line drawing of a house.

COMMANDS

Key in the program and RUN.

```
10 REM PROGRAM - ROTATING HOUSE
20 DIM H%(10,3),S%(10,3),P%(10,3)
30 DEF FNX(K)=H%(K,1)+140
40 DEF FNY(K)=H%(K,2)+100
50 FOR I=1 TO 10
60 FOR J=1 TO 3
70 READ H%(I,J):S%(I,J)=H%(I,J)
80 NEXT J
90 NEXT I
100 DATA 0,0,-100,50,0,-100,50,-40,-100,25,-50
110 DATA -100,0,-40,-100,0,0,-25,50 ,0,-25,50
120 DATA -40,-25,25,-50 ,-25,0,-40,-25
130 REM END OF DATA
140 GOSUB 410:REM DRAW HOUSE
150 VX=500:VY=-500:VZ=800:REM VIEW POINT
160 C=COS([]/10):S=SIN([]/10)
170 REM LOOP BACK POINT
180 DATA 0,0,-100,50,0,-100,50,-40,-100,25,-50
190 REM PERSPECTIVE ROUTINE
200 FOR I=1 TO 10
210 PX=S%(I,1)-VX
220 PY=S%(I,2)-VY
230 PZ=S%(I,3)-VZ
240 R=-VZ/PZ
250 QX=R*PX+VX
260 QY=R*PY+VY
270 P%(I,1)=QX
280 P%(I,2)=QY
290 P%(I,3)=0
```

```
300 NEXT I
310 FOR I=1 TO 10
320 FOR J=1 TO 3
330 H%(I,J)=P%(I,J)
340 NEXT J
350 NEXT I
360 GOSUB 410:REM CLEAR DRAW HOUSE
370 GOSUB 640 :REM ROTATE HOUSE
380 GOTO 170
390 END
400 READ H(I,J):S(I,J)=H(I,J)
410 GRAPHIC 1,1
420 DRAW 1,FNX(1),FNY(1)
430 FOR J=2 TO 5
440 DRAW TO FNX(J),FNY(J)
450 NEXT J
460 DRAW TO FNX(1),FNY(1)
470 DRAW 1,FNX(6),FNY(6)
480 FOR J=7 TO 10
490 DRAW TO FNX(J),FNY(J)
500 NEXT J
510 DRAW TO FNX(6),FNY(6)
520 DRAW 1,FNX(6),FNY(6)
530 DRAW TO FNX(1),FNY(1)
540 DRAW 1,FNX(10),FNY(10)
550 DRAW TO FNX(5),FNY(5)
560 DRAW 1,FNX(9),FNY(9)
570 DRAW TO FNX(4),FNY(4)
580 DRAW 1,FNX(8),FNY(8)
590 DRAW TO FNX(3),FNY(3)
600 DRAW 1,FNX(7),FNY(7)
610 DRAW TO FNX(2),FNY(2)
620 RETURN
630 REM ROTATE HOUSE
640 FOR I=1 TO 10
650 T1=C*S%(I,1)+S*S%(I,3)
660 T3=-S*S%(I,1)+C*S%(I,3)
670 S%(I,1)=T1:S%(I,3)=T3
680 NEXT I
690 RETURN
```

P59 Interfering circles

It takes a little time for this program to finish, but we think that the effect is quite pretty.

COMMANDS

Key in the program and RUN.

```
10 REM THIS PROGRAM USES THE CIRCLE
20 REM COMMAND TO CREATE A NICE EFFECT
25 A$="C"
30 DO WHILE A$<>"Q"
40 GRAPHIC 3,1
41 BG=INT(RND(1)*16+1)
42 FG=INT(RND(1)*16+1):IF BG=FG THEN 41
43 M1=INT(RND(1)*16+1):IF M1=FG OR M1=BG THEN 43
44 M2=INT(RND(1)*16+1):IF M2=FG OR M2=BG OR M2=M1 THEN 44
45 COLOR0,BG:COLOR 1,FG:COLOR 2,M1:COLOR 3,M2
50 FOR N=1 TO 40
60 FOR C=1 TO 3
70 CIRCLE C,50,87,N
80 CIRCLE C,80,87,N
90 NEXT C
100 NEXT N
110 GET A$:IF A$="" THEN 110
120 LOOP
130 GRAPHIC 0
140 END
```

P60 Zoom

One way of zooming in or out of a picture is to redraw the whole scene to a different size.

This program draws a cube and allows the user to zoom the cube into and out of the screen. (Photographers may wish to reverse our definitions of in and out.)

COMMANDS

Key in the program and RUN.
Press key I to zoom in.
Press key O to zoom out.

```
10 REM PROGRAM - ZOOM
20 SCNCLR
30 CHAR ,17,13,"ZOOM"
40 A=TI
50 DO:LOOP UNTIL TI>A+150
60 GRAPHIC 1,1
70 S=10:GOSUB 170
80 GETKEY A$
90 IF A$="Q" THEN GRAPHIC 0,0:END
100 D=-(A$="I")+(A$="O")
110 SCNCLR
120 S=S+D:GOSUB 170
130 GOTO 80
140 END
150 REM SUBROUTINE TO DRAW CUBE
160 REM OF SIDE LENGTH S
170 IF S<0 THEN RETURN
180 DRAW 1,(-S+160),(S+100)
190 DRAW TO (S+160),(S+100)
200 DRAW TO (S+160),(-S+100)
210 DRAW TO (1.5*S+160),100
220 DRAW TO (1.5*S+160),(2*S+100)
230 DRAW TO (S/1.25-S+160),(2*S+100)
240 DRAW TO (-S+160),(S+100)
250 DRAW TO (-S+160),(-S+100)
260 DRAW TO (S+160),(-S+100)
```

```
270 DRAW 1,(S+160),(S+100)
280 DRAW TO (1.5*S+160),(2*S+100)
290 RETURN
```

P61 Interference

This program generates an interference type pattern.

Once the pattern has been generated, the user can change the colors on the screen by pressing keys.

COMMANDS

Key in the program and RUN.
When pattern has been generated, press any key to change colors.
Press Q to quit.

```
10 REM THIS PROGRAM USES THE DRAW
20 REM COMMAND TO CREATE A NICE EFFECT
30 A$="C"
40 DO WHILE A$<>"Q"
50 GRAPHIC 1,1
60 BG=INT(RND(1)*16+1)
70 FG=INT(RND(1)*16+1):IF BG=FG THEN 60
80 S=INT(RND(1)*4+3)
90 COLOR0,BG:COLOR 1,FG
100 FOR I=0 TO 199 STEP S
110 Y=199-I
120 DRAW 1,0,I TO 319,Y
130 NEXT I
140 FOR I=319 TO 0 STEP -S
150 DRAW 1,I,0 TO 319-I,199
160 NEXT I
170 GET A$:IF A$="" THEN 170
180 LOOP
190 GRAPHIC 0
200 COLOR 0,7:COLOR 1,1
210 END
```

P62 Doodle

This program allows the user to use the cursor control keys to doodle on the C-16 screen.

Full instructions for the program's use are presented to the user.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM DOODLE
20 SCNCLR
30 CHAR 0,17,13,"DOODLE"
40 A=TI
50 DO:LOOP UNTIL TI>A+150
60 SCNCLR
70 PRINT:PRINT:PRINT:PRINT
80 PRINT "THIS PROGRAM ALLOWS THE USER TO DOODLE"
90 PRINT "ON THE SCREEN USING THE CURSOR KEYS TO"
100 PRINT "CONTROL A DOT ON THE SCREEN.":PRINT:PRINT
110 PRINT "THE CURSOR KEYS CAN BE USED TO MOVE "
120 PRINT "THE DOT OR DRAW WITH THE DOT.":PRINT:PRINT
130 PRINT "ONCE AN AREA HAS BEEN ENCLOSED IT CAN "
140 PRINT "BE PAINTED.":PRINT:PRINT
150 PRINT "PRESS D TO DRAW"
160 PRINT "       M TO MOVE"
170 PRINT "       PN TO PAINT IN COLOR N"
180 PRINT :PRINT "PRESS ANY KEY TO START"
190 GETKEY A$
200 GRAPHIC 3,1
210 F=2:X=80:Y=100
220 DO
230 GET A$
240 IF A$="D" THEN F=1
250 IF A$="M" THEN F=2
260 IF A$="P" THEN F=3
270 A=ASC(A$)
280 X=X+(A=157)-(A=29)
```

```
290 Y=Y+(A=145)-(A=17)
300 ON F GOSUB 340,370,420
310 LOOP UNTIL A$="Q"
320 GRAPHIC 0,0
330 END
340 REM DRAWING SUBROUTINE, NORMAL DEFAULT
350 DRAW TO X,Y
360 RETURN
370 REM MOVING SUBROUTINE
380 LOCATE X,Y
390 DRAW (3-RDOT(2)),X,Y
400 DRAW (3-RDOT(2)),X,Y
410 RETURN
420 REM PAINT SUBROUTINE
430 GETKEY K$:K=VAL(K$)
440 IF K>3 THEN K=3
450 PAINT K,,1
460 F=2
470 RETURN
```

P63 Bar chart

This program can draw a chart of up to thirty bars on to the screen. The bars are automatically scaled to fit onto the screen. The chart is not labelled. This is left as an exercise for the reader.

COMMANDS

Key in the program and RUN.
Enter the number of bars, less than 30.
Enter the value of each bar as requested.

```
10 REM PROGRAM - BAR CHART
20 SCNCLR
30 CHAR ,15,12,"BAR CHART"
40 A=TI
50 DO:LOOP UNTIL TI>A+150
60 SCNCLR:PRINT:PRINT:PRINT
70 PRINT "THIS PROGRAM CAN BE USED TO PRESENT DATA";
80 PRINT "IN THE FORM OF A BAR CHART ON THE SCREEN";
90 PRINT "UP TO 30 BARS CAN BE DISPLAYED ON THE"
100 PRINT "SCREEN."
110 PRINT :PRINT "PRESS ANY KEY TO CONTINUE"
120 GETKEY A$
130 PRINT:PRINT
140 INPUT "ENTER NUMBER OF BARS";BR
150 IF BR>30 THEN GOTO 140
160 IF BR<1 THEN GOTO 140
170 BR=INT(BR):MX=0
180 DIM V(BR)
190 PRINT
200 PRINT "NOW ENTER YOUR DATA ONE ELEMENT AT A "
210 PRINT "TIME."
220 FOR I=1 TO BR
230 PRINT "ENTER VALUE OF BAR ";I;
240 INPUT V(I)
250 IF V(I)>MX THEN MX=V(I)
260 NEXT I
270 INPUT "ENTER TITLE OF CHART";T$
```

```
280 SCNCLR
290 SC=1
300 IF MX>20 THEN SC=MX/20
310 FOR I=1 TO BR
320 V(I)=INT(V(I)/SC)
330 NEXT I
340 REM DRAW SCALES
350 FOR I=0 TO 19
360 CHAR 1,5,I,CHR$(98)
370 NEXT I
380 CHAR 1,5,20,CHR$(123)
390 FOR I=6 TO 39
400 CHAR 1,I,20,CHR$(99)
410 NEXT I
420 REM LABEL BAR CHART
430 L=LEN(T$)
440 CHAR 1,INT((40-L)/2),2,T$
450 FOR I=1 TO 20 STEP 2
460 T=INT(I*SC+.5)+INT(SC)
470 TP$=MID$(STR$(T),2)
480 CHAR 1,0,20-I,TP$
490 NEXT I
500 REM NOW PLOT THE BARS
510 CL=1
520 FOR I=1 TO BR
530 CL=CL+1:IF CL=16 THEN CL=1
540 FOR J=0 TO V(I)-1
550 COLOR 1,CL
560 CHAR 1,7+I,19-J,CHR$(166)
570 NEXT J
580 NEXT I
590 GETKEY A$
```

P64 Mean and standard deviation

This program is used to find the mean and standard deviation of a list of data items.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM - MEAN AND STANDARD DEVIATION
20 SCNCLR:PRINT
30 PRINT "      MEAN AND STANDARD DEVIATION"
40 PRINT:PRINT "THIS PROGRAM CAN BE USED TO FIND THE"
50 PRINT "MEAN AND STANDARD DEVIATION OF A SERIES"
60 PRINT "OF NUMERICAL VALUES. THIS PROGRAM CAN BE";
70 PRINT "OF GREAT USE IN THE LABORATORY."
80 PRINT
90 PRINT "THE PROGRAM ASKS YOU TO FIRST OF ALL TO"
100 PRINT "PRINT THE TITLE OF THE REPORT TO BE "
110 PRINT "WRITTEN, THEN THE NUMERICAL READINGS"
120 PRINT "ARE ENTERED ONE AT A TIME. THE PROGRAM"
130 PRINT "THEN PRINTS A REPORT OF THE DATA."
140 PRINT
150 PRINT "PRESS ANY KEY TO CONTINUE"
160 GETKEY A$
170 SCNCLR
180 PRINT "WHAT IS THE TITLE OF THE"
190 INPUT "REPORT";T$
200 PRINT
210 INPUT "HOW MANY DATA ITEMS ARE THERE";N
220 DIM D(N):S=0:D=0
230 PRINT
240 PRINT "PLEASE ENTER THE DATA ONE ITEM AT A TIME"
250 FOR I=1 TO N
260 INPUT "NEXT ITEM";D(I):S=S+D(I)
270 NEXT
280 M=S/N
290 PRINT "CALCULATING.";
300 REM NOW DO THE CALCULATIONS
310 FOR I=1 TO N
```

```
320 PRINT ".";
330 D=D+(D(I)-M)^2
340 NEXT I
350 VR=SQR(D/(N-1))
360 SCNCLR
370 PRINT "               ";T$
380 PRINT
390 PRINT
400 PRINT "THERE WERE ";N;" DATA ITEMS."
410 PRINT
420 PRINT "THE DATA ARE"
430 PRINT
440 FOR I=1 TO N
450 PRINT D(I);" ";
460 NEXT I
470 PRINT :PRINT:PRINT
480 PRINT "THE MEAN IS";
490 PRINT USING "######.##";M
500 PRINT "THE STANDARD DEVIATION IS";
510 PRINT USING "######.##";VR
520 END
```

P65 Bubble sort

This program is demonstration of how the classical Bubble Sort works. The program sorts ten numbers on the screen.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM - BUBBLE SORT
20 DIM K(10):C=0:S=0
30 PRINT "[CS CD CD CD CD CD CD CD CD CD CD ]"
40 PRINT "                BUBBLE SORT"
50 A=TI
60 IF TI<A+100 THEN GOTO 60
70 SCNCLR
80 PRINT "[CD CD ]THIS PROGRAM EXHIBITS A SORTING ROUTINE"
90 PRINT "ON THE SCREEN."
100 PRINT: PRINT
110 PRINT "THE ROUTINE IS KNOWN AS THE BUBBLE SORT"
120 PRINT:PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
130 GET A$:IF A$="" THEN GOTO 130
140 PRINT "[CS ]"
150 PRINT "THE PROGRAM SHOWS HOW THE COMPUTER"
160 PRINT "COMPARES AND SWAPS ITEMS IN A LIST OF"
170 PRINT "NUMBERS, IN ORDER TO OBTAIN A SORTED"
180 PRINT "LIST."
190 PRINT
200 PRINT "NUMBERS BEING COMPARED ARE SHOWN IN"
210 PRINT "RED, NUMBERS BEING SWAPPED ARE SHOWN"
220 PRINT "IN YELLOW."
230 PRINT:PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
240 GET A$:IF A$="" THEN GOTO 240
250 PRINT "[CU CU ]THE PROGRAM ALLOWS THE USER TO ENTER"
260 PRINT "THE SPEED OF PROCESSING, IN THE RANGE"
270 PRINT "[CD ]1 TO 10, WITH 10 BEING FAST."
280 PRINT "THE PROGRAM PROMPTS THE USER TO ENTER"
290 PRINT "10 NUMBERS."
```

```
300 PRINT:PRINT
310 PRINT "FOR THE PURPOSES OF THIS DEMONSTRATION"
320 PRINT"ENSURE THAT ALL NUMBERS ENTERED HAVE"
330 PRINT "THE SAME NUMBER OF DIGITS."
340 PRINT:PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
350 GET A$:IF A$="" THEN GOTO 350
360 PRINT "[CS ]"
370 INPUT"SPEED OF PROCESSING";S
380 IF S>10 THEN S=10
390 IF S<1 THEN S=1
400 S=INT(11-S)*50
410 PRINT
420 PRINT "ENTER NUMBERS TO BE SORTED ONE AT A TIME"
430 INPUT "NUMBER";K(1)
440 L=LEN(STR$(K(1)))
450 FOR I=2 TO 10
460 INPUT "NUMBER";K(I)
470 IF LEN(STR$(K(I)))<>L THEN PRINT "PLEASE RE-ENTER":GOTO
460
480 NEXT I
490 GET A$:IF A$<>"" THEN 490:REM FLUSH BUFFER
500 PRINT "[CS RON ]              BUBBLE SORT
[ROF ]"
510 PRINT "[CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD
CD CD CD RON ]                                          [ROF
]";
520 FOR I=1 TO 10
530 CHAR 1,20,(I+5),STR$(K(I))
540 NEXT I
550 PRINT "[CH CD CD ]COMPARISONS=                SWAPS=";
560 REM SORTING ROUTINE
570 FOR I=1 TO 9
580 FOR K=I+1 TO 10
590 HU=3:GOSUB 690
600 C=C+1
610 PRINT "[CH CD CD CR CR CR CR CR CR CR CR CR CR CR CR ]"C;
620 FOR Z=1 TO S:NEXT Z:REM DELAY
630 IF K(K)>K(I) THEN GOSUB 750
640 HU=1:GOSUB 690
650 NEXT K
660 NEXT I
670 END
680 REM SUBROUTINE TO COLOR TWO ITEMS
690 COLOR 1,HU,6
700 CHAR 1,20,I+5,STR$(K(I))
710 CHAR 1,20,K+5,STR$(K(K))
720 COLOR 1,1
```

```
730 RETURN
740 REM SWAP SUBROUTINE
750 SW=SW+1
760 HU=8:GOSUB 690
770 T=K(K)
780 K(K)=K(I)
790 K(I)=T
800 HU=8:GOSUB 690
810 FOR Z=1 TO S:NEXT Z
820 HU=1:GOSUB 690
830 PRINT "[CH CD CD CR CR CR CR CR CR CR CR CR CR CR CR CR
CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR
CR ]"SW;
840 RETURN
```

P66 Shell sort

This is the classical fast Shell Sort routine. As with the Bubble sort, the sorting is carried out on the screen. It can be a very useful exercise to try and figure out why the Shell routine works.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM - SHELL  SORT
20 DIM K(10):C=0:SW=0
30 PRINT "[CS CD CD CD CD CD CD CD CD CD CD ]"
40 PRINT "                SHELL  SORT"
50 A=TI
60 IF TI<A+100 THEN GOTO 60
70 SCNCLR
80 PRINT "[CD CD ]THIS PROGRAM EXHIBITS A SORTING ROUTINE"
90 PRINT "ON THE SCREEN."
100 PRINT: PRINT
110 PRINT "THE ROUTINE IS KNOWN AS THE SHELL SORT"
120 PRINT:PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
130 GETKEY A$
140 PRINT "[CS ]"
150 PRINT "THE PROGRAM SHOWS HOW THE COMPUTER"
160 PRINT "COMPARES AND SWAPS ITEMS IN A LIST OF"
170 PRINT "NUMBERS, IN ORDER TO OBTAIN A SORTED"
180 PRINT "LIST."
190 PRINT
200 PRINT "NUMBERS BEING COMPARED ARE SHOWN IN"
210 PRINT "RED, NUMBERS BEING SWAPPED ARE SHOWN"
220 PRINT "IN YELLOW."
230 PRINT:PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
240 GETKEY A$
250 PRINT "[CU CU ]THE PROGRAM ALLOWS THE USER TO ENTER"
260 PRINT "THE SPEED OF PROCESSING, IN THE RANGE"
270 PRINT "[CD ]1 TO 10, WITH 10 BEING FAST."
280 PRINT "THE PROGRAM PROMPTS THE USER TO ENTER"
290 PRINT "10 NUMBERS."
```

```
300 PRINT:PRINT
310 PRINT "FOR THE PURPOSES OF THIS DEMONSTRATION"
320 PRINT"ENSURE THAT ALL NUMBERS ENTERED HAVE"
330 PRINT "THE SAME NUMBER OF DIGITS."
340 PRINT:PRINT:PRINT "PRESS ANY KEY TO CONTINUE"
350 GETKEY A$
360 PRINT "[CS ]"
370 INPUT"SPEED OF PROCESSING";S
380 IF S>10 THEN S=10
390 IF S<1 THEN S=1
400 S=INT(11-S)*50
410 PRINT
420 PRINT "ENTER NUMBERS TO BE SORTED ONE AT A TIME"
430 INPUT "NUMBER";K(1)
440 L=LEN(STR$(K(1)))
450 FOR I=2 TO 10
460 INPUT "NUMBER";K(I)
470 IF LEN(STR$(K(I)))<>L THEN PRINT "PLEASE RE-ENTER":GOTO
460
480 NEXT I
490 GET A$:IF A$<>"" THEN 490:REM FLUSH BUFFER
500 PRINT "[CS RON ]              SHELL  SORT
[ROF ]"
510 PRINT "[CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD
CD CD CD RON ]                                          [ROF
]";
520 FOR I=1 TO 10
530 CHAR 1,20,I+5,STR$(K(I))
540 NEXT I
550 PRINT "[CH CD CD ]COMPARISONS=                SWAPS=";
560 REM SORTING ROUTINE
570 D=4
580 FOR I=D+1 TO 10
590 FOR K=I-D TO 1 STEP -D
600 C=C+1:PRINT "[CH CD CD CR CR CR CR CR CR CR CR CR CR CR
CR ]"C;
610 P=K+D:HU=3:GOSUB 710
620 FOR Z=1 TO S:NEXT Z:REM DELAY
630 IF K(K)>K(P) THEN GOSUB 770
640 HU=1:GOSUB 710
650 NEXT K
660 NEXT I
670 D=INT(D/2)
680 IF D<>0 THEN GOTO 580
690 END
700 REM SUBROUTINE TO COLOR TWO ITEMS
710 COLOR 1,HU,6
```

```
720 CHAR 1,20,P+5,STR$(K(P))
730 CHAR 1,20,K+5,STR$(K(K))
740 COLOR 1,1
750 RETURN
760 REM SWAP SUBROUTINE
770 SW=SW+1
780 HU=8:GOSUB 710
790 T=K(K)
800 K(K)=K(P)
810 K(P)=T
820 HU=8:GOSUB 710
830 FOR Z=1 TO S:NEXT Z
840 HU=1:GOSUB 710
850 PRINT "[CH CD CD CR CR CR CR CR CR CR CR CR CR CR CR CR
CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR CR
CR ]"SW;
860 RETURN
```

P67 Merge

A common need in data processing is the ability to merge two sorted files to produce a third sorted file.

It is quicker to sort small files and then to merge the files to form larger ones. In this program, we mimic file handling by using arrays. The array elements are entered via the keyboard but the program could be amended to allow the elements to be entered via tape files.

COMMANDS

Key in the program and RUN.
Enter the array elements when prompted in increasing order.

```
10 REM PROGRAM - MERGE
20 PRINT "[CS CD CD CD CD CD CD CD CD ]"
30 PRINT "               MERGE"
40 A=TI
50 IF TI<A+100 THEN GOTO 50
60 PRINT "[CS ]"
70 PRINT "THE BASIS OF MANY DISK BASED SORT"
80 PRINT "ROUTINES IS THE CAPABILITY OF MERGING"
90 PRINT "TOGETHER TWO LISTS SORTED PREVIOUSLY."
100 PRINT "THE DISK BASED SORT ROUTINE RECOGNISES"
110 PRINT "THAT IT MIGHT BE IMPOSSIBLE TO READ A "
120 PRINT "WHOLE FILE INTO MEMORY FOR SORTING"
130 PRINT "WITH EITHER A BUBBLE OR SHELL SORT"
140 PRINT:PRINT
150 PRINT "PRESS ANY KEY TO CONTINUE"
160 GET A$:IF A$="" THEN GOTO 160
170 PRINT"[CU ]";
180 PRINT "THE FILE IS THEREFORE SPLIT INTO A"
190 PRINT "SERIES OF SUBFILES, SMALL ENOUGH TO READ";
200 PRINT "INTO MEMORY FOR SORTING."
210 PRINT
220 PRINT "THE SMALL SORTED SUBFILES ARE THEN "
230 PRINT "MERGED TO PRODUCE THE COMPLETELY "
240 PRINT "SORTED FILE.":PRINT
250 PRINT "PRESS ANY KEY TO CONTINUE"
260 GET A$:IF A$="" THEN GOTO 260
```

```
270 PRINT "[CS ]":PRINT:PRINT
280 PRINT
290 PRINT "THIS PROGRAM DEMONSTRATES A MERGE"
300 PRINT "ROUTINE, BY MERGING TOGETHER TWO LISTS"
310 PRINT "OF NUMBERS EACH HOLDING UP TO 100 ITEMS"
320 PRINT:PRINT
330 PRINT "PRESS ANY KEY TO CONTINUE"
340 GET A$:IF A$="" THEN GOTO 340
350 DIM F(100),S(100),M(200)
360 PRINT "[CS ]":PRINT:PRINT
370 PRINT "ENTER THE FIRST LIST ONE ITEM AT A TIME"
380 PRINT:PRINT "MAKE SURE THAT THE DATA ITEMS ARE IN"
390 PRINT "INCREASING ORDER.":PRINT
400 PRINT "USE THE VALUE 0 TO TERMINATE THE LIST"
410 PRINT
420 PRINT "PRESS ANY KEY TO CONTINUE"
430 GET A$:IF A$="" THEN GOTO 430
440 I=1
450 PRINT "ENTER ELEMENT OF FIRST LIST";:INPUT F(I)
460 I=I+1
470 PRINT "ENTER ELEMENT OF FIRST LIST";:INPUT F(I)
480 IF F(I)<>0 AND NOT(F(I)<F(I-1)) AND I<101 THEN GOTO 460
490 IF F(I)=0 THEN GOTO 510
500 IF F(I)<F(I-1) THEN PRINT "OUT OF ORER":GOTO 470
510 I=I-1
520 PRINT "FIRST LIST COMPLETE, THERE ARE "I"ITEMS IN THE
LIST"
530 N1=I
540 PRINT:PRINT:PRINT
550 PRINT "PRESS ANY KEY TO CONTINUE"
560 GET A$:IF A$="" THEN GOTO 560
570 I=1
580 PRINT "ENTER ELEMENT OF SECOND LIST";:INPUT S(I)
590 I=I+1
600 PRINT "ENTER ELEMENT OF SECOND LIST";:INPUT S(I)
610 IF S(I)<>0 AND NOT(S(I)<S(I-1)) AND I<101 THEN GOTO 590
620 IF S(I)=0 THEN GOTO 640
630 IF S(I)<S(I-1) THEN PRINT "OUT OF ORER":GOTO 600
640 I=I-1
650 PRINT "FIRST LIST COMPLETE, THERE ARE "I"ITEMS IN THE
LIST"
660 N2=I
670 PRINT:PRINT:PRINT
680 PRINT "PRESS ANY KEY TO CONTINUE"
690 GET A$:IF A$="" THEN GOTO 690
700 PRINT "[CS ]"
710 I=1:J=1
```

```
720 PRINT "[CH CD CD CD ]FIRST LIST ELEMENT NUMBER ";
730 PRINT I
740 PRINT "VALUE ";F(I)
750 PRINT "[CH CD CD CD CD CD CD ]SECOND LIST ELEMENT NUMBER
";
760 PRINT J
770 PRINT "VALUE "S(J)
780 PRINT "[CH CD CD CD CD CD CD CD CD CD ]MERGE LIST ELEMENT
NUMBER ";
790 PRINT I+J-1
800 PRINT "VALUE ";
810 IF F(I)<S(J) THEN M(I+J-1)=F(I):PRINT M(I+J-1):I=I+1:GOTO
830
820 IF NOT(F(I)<S(J)) THEN M(I+J-1)=S(J):PRINTM(I+J-1):J=J+1
830 IF I=N1+1 THEN GOTO 860
840 IF J=N2+1 THEN GOTO 980
850 GOSUB 2000
855 GOTO 720
860 REM  RUNOUT J LIST
870 FOR K=J TO N2
875 GOSUB 2000
880 M(K+I-1)=S(K)
890 PRINT "[CH CD CD CD CD CD CD ]SECOND LIST ELEMENT NUMBER
";
900 PRINT K
910 PRINT "VALUE "S(K)
920 PRINT "[CH CD CD CD CD CD CD CD CD CD ]MERGE LIST ELEMENT
NUMBER ";
930 PRINT I+K-1
940 PRINT "VALUE "M(I+K-1)
950 NEXT K
960 GOTO 1070
970 REM RUNOUT I
980 FOR K=I TO N1
985 GOSUB 2000
990 M(K+J-1)=F(K)
1000 PRINT "[CH CD CD CD ]FIRST LIST ELEMENT NUMBER ";
1010 PRINT K
1020 PRINT "VALUE "F(K)
1030 PRINT "[CH CD CD CD CD CD CD CD CD CD ]MERGE LIST
ELEMENT NUMBER ";
1040 PRINT J+K-1
1050 PRINT "VALUE "M(J+K-1)
1060 NEXT K
1070 INPUT "DISPLAY RESULT (Y/N)";R$
1080 IF R$="N" THEN STOP
1090 PRINT "[CS ]"
```

```
1100 PRINT "FIRST LIST   SECOND LIST   MERGED LIST"
1110 FOR I=1 TO N1+N2
1120 PRINT F(I);TAB(14);S(I);TAB(28);M(I)
1130 NEXT I
1140 END
1990 REM DELAY SUBROUTINE
2000 PRINT "[CH CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD
CD CD ]PRESS ANY KEY FOR NEXT ITEM"
2100 GET A$:IF A$="" THEN GOTO 2100
2110 PRINT "[CU ]                                "
2120 RETURN
```

P68 Permutations

This program can be used to find the number of permutations of n objects taken r at a time. This is a very useful routine in statistics.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM - PERMUTATIONS
20 PRINT "[CS CD CD CD CD CD CD CD CD ]"
30 PRINT "               PERMUTATIONS"
40 A=TI
50 IF TI<A+100 THEN GOTO 50
60 PRINT "[CS ]"
70 PRINT "THIS PROGRAM CAN BE USED TO FIND THE "
80 PRINT "NUMBER OF PERMUTATIONS OF N OBJECTS,"
90 PRINT"TAKEN R AT A TIME."
100 PRINT
110 PRINT "FOR EXAMPLE, SUPPOSE THAT THERE ARE 4"
120 PRINT "PEOPLE IN A RACE, THEN HOW MANY WAYS"
130 PRINT "CAN THE FIRST 3 POSITIONS BE FILLED?"
140 PRINT
150 PRINT "SUPPOSE THE RACERS ARE CALLED A,B,C AND"
160 PRINT "D, THEN POSSIBLE FINISHING POSITIONS"
170 PRINT "WOULD BE AS FOLLOWS:"
180 PRINT
190 PRINT"ABC ABD ACB ACD ADB ADC"
200 PRINT "BAC BAD BCA BCD BDA BDC"
210 PRINT "CAB CAD CBA CBD CDA CDB"
220 PRINT "DAB DAC DBA DBC DCA DCB"
230 PRINT
240 PRINT "IN THIS EXAMPLE THERE ARE 24 WAYS OF"
250 PRINT "PLACING THE FIRST 3 IN THE RACE."
260 PRINT:PRINT
270 PRINT "PRESS ANY KEY TO CONTINUE"
280 GET A$:IF A$="" THEN GOTO 280
290 PRINT"[CS ]"
300 PRINT "THIS PROGRAM WILL ALLOW THE USER TO"
310 PRINT "CALCULATE THE NUMBER OF PERMUTATIONS"
```

```
320 PRINT:PRINT
330 PRINT "PRESS ANY KEY TO CONTINUE"
340 GET A$:IF A$="" THEN GOTO 340
350 PRINT:PRINT
360 PRINT "HOW MANY OBJECTS ARE TO BE SELECTED"
370 INPUT "FROM";OB
380 PRINT
390 PRINT "HOW MANY OBJECTS ARE TO BE SELECTED";:INPUT SE
400 PERMS=1:N=OB
410 FOR I=1 TO SE
420 PERMS=PERMS*N
430 N=N-1
440 IF N=0 THEN GOTO 460
450 IF (10^38)/N<PERMS THEN PRINT "TOO BIG":STOP
460 NEXT I
470 PRINT:PRINT
480 PRINT "THE NUMBER OF WAYS OF SELECTING ";SE
490 PRINT "OBJECTS FROM ";OB;" OBJECTS IS"
500 PRINT PERMS
510 END
```

P69 Combinations

This program finds the number of combinations of n objects taken r at a time.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM - COMBINATIONS
20 PRINT "[CS CD CD CD CD CD CD CD CD ]"
30 PRINT "               COMBINATIONS"
40 A=TI
50 IF TI<A+100 THEN GOTO 50
60 PRINT "[CS ]"
70 PRINT "THIS PROGRAM CAN BE USED TO FIND THE "
80 PRINT "NUMBER OF COMBINATIONS OF N OBJECTS,"
90 PRINT"TAKEN R AT A TIME."
100 PRINT
110 PRINT "FOR EXAMPLE, SUPPOSE THAT WE WISH TO "
120 PRINT "MAKE UP A COMMITTEE OF 3 PEOPLE OUT "
130 PRINT "OF A POSSIBLE OF 4 PEOPLE. HOW MANY "
140 PRINT "CAN THIS BE DONE"
150 PRINT "SUPPOSE THE PEOPLE ARE CALLED A,B,C AND"
160 PRINT "SUPPOSE THE PEOPLE ARE CALLED A,B,C AND"
170 PRINT "D, THEN POSSIBLE COMMITTEES WOULD BE"
180 PRINT "AS FOLLOWS:"
190 PRINT
200 PRINT"ABC ABD ACD BCD"
210 PRINT
220 PRINT "IN THIS EXAMPLE THERE ARE 4 WAYS OF"
230 PRINT "MAKING UP THE COMMITTEE."
240 PRINT:PRINT
250 PRINT "PRESS ANY KEY TO CONTINUE"
260 GET A$:IF A$="" THEN GOTO 260
270 PRINT"[CS ]"
280 PRINT "THIS PROGRAM WILL ALLOW THE USER TO"
290 PRINT "CALCULATE THE NUMBER OF PERMUTATIONS"
300 PRINT:PRINT
310 PRINT "PRESS ANY KEY TO CONTINUE"
320 GET A$:IF A$="" THEN GOTO 320
```

```
330 PRINT:PRINT
360 PRINT "HOW MANY OBJECTS ARE TO BE SELECTED"
370 INPUT "FROM";OB
380 PRINT
390 PRINT "HOW MANY OBJECTS ARE TO BE SELECTED";:INPUT SE
400 N=OB:GOSUB 1000
410 FC=FT
420 N=SE:GOSUB 1000
430 FR=FT
440 N=OB-SE:GOSUB 1000
450 FM=FT
460 COMBS=FC/FR/FM
470 PRINT:PRINT
480 PRINT "THE NUMBER OF WAYS OF SELECTING ";SE
490 PRINT "OBJECTS FROM ";OB;" OBJECTS IS"
500 PRINT COMBS
510 END
970 REM FACTORIAL SUBROUTINE
980 REM THIS PROGRAM EVALUATES N! AND PLACES THE RESULT
990 REM IN THE PARAMETER FT
1000 IF N<1 THEN PRINT "GOTCHA! N<1":STOP
1010 IF INT(N)<>N THEN PRINT "GOTCHA! N NOT AN INTEGER":STOP
1020 IF N>33 THEN PRINT "GOTCHA! TOO BIG FOR THE MACHINE":
STOP
1030 FT=1
1040 FOR I=1 TO N
1050 FT=FT*I
1060 NEXT I
1070 RETURN
```

P70 Least squares

This program uses the method of least squares to find the best straight line through a set of data points.

The straight line found is in the form

Y=MX+B

when the equation is formed it is printed out.

COMMANDS

Key in the program and RUN.
Enter the data items in the form X,Y.

```
10 REM PROGRAM - LEAST SQUARES
20 PRINT "[CS CD CD CD CD CD CD CD CD CD CD ]"
30 PRINT "              LEAST SQUARES"
40 A=TI
50 IF TI<A+100 THEN GOTO 50
60 PRINT "[CS ]"
70 PRINT "THIS PROGRAM IS USED TO FIND THE BEST"
80 PRINT "STRAIGHT LINE FIT TO A SET OF DATA "
90 PRINT "POINTS. THE BEST STRAIGHT LINE IS FOUND"
100 PRINT "BY THE METHOD OF LEAST SQUARES."
110 GOSUB 2000
120 PRINT "ENTER NUMBER OF DATA POINTS"
130 INPUT N
140 DIM X(N),Y(N)
150 PRINT:PRINT:PRINT
160 XS=0:SS=0:YS=0:XY=0
170 PRINT "ENTER DATA IN THE FORM OF NUMBER PAIRS"
180 PRINT "X AND Y"
190 FOR I= 1 TO N
200 PRINT "[CH CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD
CD ]X= ";
210 INPUT X(I)
220 INPUT "Y= ";Y(I)
230 XS=XS+X(I)
240 SS=SS+X(I)*X(I)
250 YS=YS+Y(I)
260 XY=XY+X(I)*Y(I)
```

```
270 NEXT I
280 D=N*SS-XS*XS
290 IF D=0 THEN PRINT "NO FIT POSSIBLE BY THIS ROUTINE":STOP
300 M=(N*XY-XS*YS)/D
310 B=YS/N-M*XS/N
320 PRINT "LINE IS Y=";M;"*X+";B
330 END
1990 REM GET KEY SUBROUTINE
2000 PRINT:PRINT:PRINT
2010 PRINT "PRESS ANY KEY TO CONTINUE"
2020 GET A$:IF A$="" THEN GOTO 2020
2030 RETURN
```

P71 Number of days

It can be interesting in many everyday problems to calculate the number of days between two dates. This program does this by calculating the number of days from day zero of the Gregorian Calendar for each date and then calculating the number of days between the two dates.

COMMANDS

Key in the program and type RUN.
Enter Gregorian dates when prompted.

```
10 REM PROGRAM - NUMBER OF DAYS
20 PRINT "[CS CD CD CD CD CD CD CD CD CD CD ]"
30 PRINT "           NUMBER OF DAYS "
40 A=TI
50 IF TI<A+100 THEN GOTO 50
60 PRINT "[CS ]"
70 PRINT "THIS PROGRAM EVALUATES THE NUMBER OF "
80 PRINT "DAYS BETWEEN TWO DATES."
90 PRINT :PRINT
100 PRINT "THE PROGRAM USES THE GREGORIAN CALANDER"
110 PRINT:PRINT
120 PRINT "ENTER FIRST DAY OF INTEREST"
130 INPUT "MONTH (1 TO 12)";M1
140 INPUT "DAY (1 TO 31)";D1
150 INPUT "YEAR";Y1
160 MM=M1:DD=D1:YY=Y1
170 GOSUB 2000
180 N1=DYS
190 PRINT:PRINT
200 PRINT "ENTER SECOND DAY OF INTEREST"
210 INPUT "MONTH (1 TO 12)";M2
220 INPUT "DAY (1 TO 31)";D2
230 INPUT "YEAR";Y2
240 MM=M2:DD=D2:YY=Y2
250 GOSUB 2000
260 N2=DYS
270 PRINT"[CS CD CD CD CD CD CD ]"
280 PRINT "THE NUMBER OF DAYS FROM":PRINT
290 PRINT M1;"/";D1;"/";Y1;"  TO":PRINT
300 PRINT M2;"/";D2;"/";Y2;"  IS":PRINT
```

```
310 PRINT N2-N1;"DAYS"
320 END
1900 REM SUBROUTINE TO CALCULATE NUMBER
1910 REM OF DAYS SINCE DAY 0 OF GREGORIAN
1920 REM CALENDAR
1930 REM DATE IS ENTERED AS DAY=DD%, MONTH=MM% AND YEAR=YY%
1940 REM NUMBER OF DAYS SINCE 0 IS
1950 REM RETURNED AS DYS
1960 REM THE DATA HELD IN THE DATA
1970 REM STATEMENT ARE THE NUMBER OF
1980 REM DAYS FROM THE BEGINNING OF THE
1990 REM YEAR TO THE 1ST OF THE MONTH
2000 RESTORE
2010 IF MM>12 OR MM<1 THEN PRINT "ERROR IN MONTH":STOP
2020 IF DD>31 OR DD<1 THEN PRINT "ERROR IN DAY":STOP
2030 FOR I=1 TO MM
2040 READ TYS
2050 NEXT I
2060 DATA 0,31,59,90,120,151,181,212,243,273,304,334
2070 TYS=TYS+DD
2080 DYS=TYS+YY*365+INT(YY/4)+1-INT(YY/100)+INT(YY/400)
2090 IF YY=INT(YY/4)*4 AND MM<3 THEN DYS=DYS-1
2100 RETURN
```

P72 Encoder

This program can be used to code a secret message. The system used for the encoding is very simple. You are asked for a key word, the characters of which are used to offset the characters of the secret message. The program uses all the normal printable characters of the Commodore 16, so that characters such as space, full stop and comma etc are also encoded.

COMMANDS

Press shift and Commodore keys together. This allows you to enter the program in upper and lower case characters.
Key in the program and RUN.
Follow instructions.

```
10 rem program - encoder
20 print "[cs cd cd cd cd cd cd cd cd cd ]"
30 print "               encoder"
40 a=ti
50 if ti<a+100 then goto 50
60 print "[cs ]"
70 print chr$(14):rem switch to allow lower case
80 print "This program can be used to produce a"
90 print "coded version of a secret message."
100 print
110 print "The program bases the encoding on a key"
120 print "word which you can specify."
130 print:print:print
140 print "Press any key to continue"
150 get a$:if a$="" then goto 150
160 print:print:print
170 print "Note that in this program we use the "
180 print "normal uppercase-lowercase character set"
190 print:print:print
200 print "Enter the secret message"
210 m$=""
220 get a$:if a$="" then goto 220
230 if asc(a$)<>13 then m$=m$+a$:goto 220
240 print "[cs cd cd cd cd cd ]"
```

```
250 print "Do you wish to see the secret message"
260 print "to check it before encoding";
270 input "(y/n)";r$
280 if left$(r$,1)="n" then goto 370
290 print:print:print
300 print "Press space bar to see message"
310 get a$:if a$="" then goto 310
320 print:print:print
330 print m$
340 print:print:print
350 print "Press space bar to hide message"
360 get a$:if a$="" then goto 360
370 print "[cs ]"
380 print "[cs cd cd cd cd cd ]"
390 print "Enter the key word"
400 k$=""
410 get a$:if a$="" then goto 410
420 if asc(a$)<>13 then k$=k$+a$:goto 410
430 print "[cs cd cd cd cd cd ]"
440 print "Do you wish to see the key word to"
450 print "check it before encoding";
460 input "(y/n)";r$
470 if left$(r$,1)="n" then goto 560
480 print:print:print
490 print "Press space bar to see key word"
500 get a$:if a$="" then goto 500
510 print:print:print
520 print k$
530 print:print:print
540 print "Press space bar to hide key word"
550 get a$:if a$="" then goto 550
560 print "[cs cd cd cd cd cd ]"
570 print "Are both the message and the key word"
580 print "all right";:input r$
590 if r$="n" then goto 10
600 print:print:print
610 rem encoding algorithm
620 dim p(25),c(255),d(255)
630 for i=1 to len(k$)
635 p=asc(mid$(k$,i,1)):if p>192 then p=p-96
640 p(i)=p-32
650 next i
660 for i=1 to len(m$)
665 c=asc(mid$(m$,i,1)):if c>192 then c=c-96
670 c(i)=c-32
680 next i
690 c$=""
```

```
700 j=0
710 for i=1 to len(m$)
720 d(i)=(c(i)+p(j+1))
730 d(i)=d(i)-int(d(i)/93)*93
740 c$=c$+chr$(d(i)+32)
750 j=j+1
760 j=j-int(j/len(k$))*len(k$)
770 next i
780 print:print:print
790 print "Your coded message is....":print:print
800 print c$
810 print:print:print
820 print "Press space bar to hide message"
830 get a$:if a$="" then goto 830
840 print "[cs ]"
850 end
```

P73 Decoder

This program can be used to decode the secret message found using the encoder program. You need to have the secret message plus the key word to find the original message.

COMMANDS

Press the Commodore and shift keys together. This allows you to enter the program as shown below.
Key in program and RUN.
Follow instructions

```
10 rem program - decoder
20 print "[cs cd cd cd cd cd cd cd cd cd cd ]"
30 print "              decoder"
40 a=ti
50 if ti<a+100 then goto 50
60 print "[cs ]"
70 print chr$(14):rem switch to allow lower case
80 print "This program can be used to produce a"
90 print "decoded version of a secret message."
100 print
110 print "The program bases the decoding on a key"
120 print "word which you can specify."
130 print:print:print
140 print "Press any key to continue"
150 get a$:if a$="" then goto 150
160 print:print:print
170 print "Note that in this program we use the "
180 print "normal uppercase-lowercase character set"
190 print:print:print
200 print "Enter the coded message"
210 c$=""
220 get a$:if a$="" then goto 220
230 if asc(a$)<>13 then c$=c$+a$:goto 220
240 print "[cs cd cd cd cd cd ]"
250 print "Do you wish to see the coded message"
260 print "to check it before decoding";
270 input "(y/n)";r$
280 if left$(r$,1)="n" then goto 370
290 print:print:print
```

```
300 print "Press space bar to see message"
310 get a$:if a$="" then goto 310
320 print:print:print
330 print c$
340 print:print:print
350 print "Press space bar to hide message"
360 get a$:if a$="" then goto 360
370 print "[cs ]"
380 print "[cs cd cd cd cd cd ]"
390 print "Enter the key word"
400 k$=""
410 get a$:if a$="" then goto 410
420 if asc(a$)<>13 then k$=k$+a$:goto 410
430 print "[cs cd cd cd cd cd ]"
440 print "Do you wish to see the key word to"
450 print "check it before encoding";
460 input "(y/n)";r$
470 if left$(r$,1)="n" then goto 560
480 print:print:print
490 print "Press space bar to see key word"
500 get a$:if a$="" then goto 500
510 print:print:print
520 print k$
530 print:print:print
540 print "Press space bar to hide key word"
550 get a$:if a$="" then goto 550
560 print "[cs cd cd cd cd cd ]"
570 print "Are both the message and the key word"
580 print "all right";:input r$
590 if r$="n" then goto 10
600 print:print:print
610 rem decoding algorithm
620 dim p(25),c(255),d(255)
630 for i=1 to len(k$)
635 p=asc(mid$(k$,i,1)):if p>192 then p=p-96
640 p(i)=p-32
650 next i
660 for i=1 to len(c$)
665 c=asc(mid$(c$,i,1)):if c>192 then c=c-96
670 c(i)=c-32
680 next i
690 m$=""
700 j=0
710 for i=1 to len(c$)
720 d(i)=(c(i)-p(j+1))
725 d(i)=d(i)-int(d(i)/93)*93
730 if d(i)<0 then d(i)=d(i)+93
```

```
740 m$=m$+chr$(d(i)+32)
750 j=j+1
760 j=j-int(j/len(k$))*len(k$)
770 next i
780 print:print:print
790 print "Your coded message is....":print:print
800 print m$
810 print:print:print
820 print "Press space bar to hide message"
830 get a$:if a$="" then goto 830
840 print "[cs ]"
850 end
```

P74 Dog race

In this program we introduce you to the Commodore dog track. It can be profitable to play the part of the bookie when running the program.

COMMANDS

Key in the program and RUN.
Follow instructions.

```
10 REM PROGRAM - DOG RACE
20 PRINT CHR$(147)
30 PRINT:PRINT:PRINT:PRINT:PRINT
40 PRINT "                DOG RACE"
50 A=TI
60 IF TI<A+150 THEN GOTO 60
70 PRINT "[CS ]":PRINT:PRINT
80 PRINT "              RACE NIGHT                "
90 PRINT:PRINT:PRINT
100 PRINT "        GOOD EVENING FOLKS"
110 PRINT "          WELCOME TO THE "
120 PRINT "        COMMODORE DOG TRACK"
130 PRINT:PRINT:PRINT
140 PRINT "PRESS ANY KEY TO CONTINUE"
150 GET A$:IF A$="" THEN GOTO 150
160 PRINT "[CS ]"
170 DIM C(10):REM COLOR ARRAY FOR DOGS
180 FOR I=1 TO 10
190 READ C(I)
200 NEXT I
210 REM PLACE THE TRAPS
220 FOR I=1 TO 10
230 I$=MID$(STR$(I),2):IF LEN(I$)=1 THEN I$=" "+I$
240 PRINT:PRINT CHR$(C(I));I$+" ";CHR$(182);SPC(34);"."
250 NEXT I
260 PRINT:PRINT "PRESS RETURN TO START"
270 GET A$:IF A$="" THEN GOTO 270
280 IF ASC(A$)<>13 THEN GOTO 270
290 PRINT "[CU ]                        ";
300 PRINT "[CH ]"
310 FOR I=1 TO 10
```

```
320 PRINT:PRINT "[CR CR ] ";CHR$(C(I));CHR$(207);"*"
330 NEXT I
340 DIM D(10),P(10)
350 FOR I=1 TO 10:D(I)=5:NEXT I
360 FIN=0
370 P=INT(RND(0)*10)+1
380 G=0
390 FOR I=1 TO 10
400 IF P=P(I) THEN G=1
410 NEXT I
420 IF G=1 THEN GOTO 370
430 X=2*P-1:Y=D(P):GOSUB 670
440 PRINT " ";CHR$(C(P));"*";
450 D(P)=D(P)+1
460 IF D(P)=38 THEN P(FIN+1)=P:FIN=FIN+1
470 IF FIN<10 THEN GOTO 370
480 X=21:Y=0:GOSUB 670
490 PRINT "PRESS ANY KEY FOR PLACINGS"
500 GET A$:IF A$="" THEN GOTO 500
510 PRINT "[CS ]"
520 PRINT:PRINT:PRINT "THE RESULTS WERE"
530 PRINT:PRINT:PRINT
540 PRINT "          1ST DOG - ";P(1)
550 PRINT "          2ND DOG - ";P(2)
560 PRINT "          3RD DOG - ";P(3)
570 :PRINT:PRINT
580 PRINT "IN ORDER THE REST WERE":PRINT:PRINT
590 FOR I=4 TO 10
600 PRINT "          DOG ";P(I)
610 NEXT I
620 END
630 END
640 REM COLOR DATA
650 DATA 28,30,144,149,150,153,154,156,158,159
660 REM PRINT AT SUBROUTINE
670 PRINT "[CH ]"
680 IF X=0 THEN GOTO 710
690 FOR I=1 TO X:PRINT:NEXT I
700 IF Y=0 THEN GOTO 720
710 FOR I=1 TO Y-1:PRINT"[CR ]";:NEXT I
720 RETURN
```

P75 Magic matrix

This program is based on an interesting idea presented in Martin Gardener's book "Mathematical Puzzles and Diversions", (Bell, 1964).

The program generates a matrix which is not a magic square but which has some interesting properties.

We have found that this is an amusing party trick, even if the matrix is simply drawn out on a piece of paper. The basic idea is rather simple; see if you can work it out.

COMMANDS

Press the shift and Commodore key together to go into lower case mode.
Key in the program and RUN.
Follow instructions.

```
10 rem program - magic matrix
20 cl$=chr$(147):home$=chr$(19):down$=chr$(17):acros$=
chr$(29)
30 print cl$;
40 x=10:y=13:gosub 930
50 print "magic matrix"
60 a=ti
70 if ti<a+150 then goto 70
80 rem set up screen display
90 brder=4:screen=0
100 purple=5:yellow=8:red$=chr$(28):black$=chr$(144)
110 case$=chr$(142):uncase$=chr$(14)
120 color brder,yellow
130 color screen,purple
140 print cl$+black$+uncase$
150 x=3:y=3:gosub 930
160 print "This program produces a square array"
170 print "   with an interesting property."
180 x=7:y=3:gosub 930
190 print "You are asked to choose any number"
200 print "   in the square. You indicate "
```

```
210 print "   your choice by a row and column "
220 print "   number."
230 x=13:y=3:gosub 930
240 print "The computer will then block out"
250 print "   all other numbers in that row"
260 print "   and column. This continues until"
270 print "   all the numbers are either chosen"
280 print "   or deleted."
290 x=20:y=3:gosub 930
300 print "Press any key to continue";
310 get a$:if a$="" then goto 310
320 print cl$;
330 x=3:y=3:gosub 930
340 print "Whatever numbers are chosen, the "
350 print "   sum of the chosen numbers will be"
360 print "   equal to the number the computer"
370 print "   will print at the bottom of the "
380 print "   screen before you start."
390 x=20:y=3:gosub 930
400 print "Press any key to continue";
410 get a$:if a$="" then goto 410
420 dim x(5,2)
430 sum=0
440 for i=1 to 5
450 x(i,1)=int(rnd(0)*31)
460 x(i,2)=int(rnd(0)*31)
470 sum=sum+x(i,1)+x(i,2)
480 next i
490 dim a(5,5),row(5),col(5)
500 print cl$+red$+"[ron ]              [rof ]"+black$
510 for n=1 to 5
520 for k=1 to 5
530 a(n,k)=x(n,2)+x(k,1)
540 p=a(n,k):gosub 870
550 x=3+2*n:y=4+4*k:gosub 930
560 print p$;
570 next k:next n
580 s$=""
590 x=20:y=1:gosub 930
600 print "Sum = "sum
610 for j=1 to 5
620 x=21:y=1:gosub 930
630 input "Row = ";rw
640 input "Column = ";cl
650 if row(rw)>0 or col(cl)>0 then 620
660 row(rw)=1:col(cl)=1
670 print red$
```

```
680 :
690 for i=0 to 4
700 x=5+i*2:y=5+4*cl:gosub 930
710 print " . ";
720 x=3+2*rw:y=9+4*i:gosub 930
730 print " . "
740 next i
750 :
760 print black$;
770 s$=s$+"+"+str$(a(rw,cl))
780 x=3+2*rw:y=35:gosub 930
790 print a(rw,cl);
800 next j
810 x=24:y=1:gosub 930
820 print mid$(s$,2)+"=";sum;
830 get a$:if a$="" then goto 830
840 print cl$+case$
850 end
860 rem format subroutine
870 p$=str$(p)
880 if len(p$)>=3 then goto 910
890 p$=" "+p$
900 goto 880
910 return
920 rem subroutine to place cursor at position x,y
930 print home$;
940 if x=1 then 960
950 for r=1 to x:print down$;:next r
960 if y=1 then 980
970 for c=1 to y:print acros$;:next c
980 return
```

P76 Shuffle

This program shuffles a deck of cards on the screen. The program uses the Commodore card characters.

You could consider this program to be the kernel of any card based game.

COMMANDS

Key in the program and RUN.

```
10 REM PROGRAM  -  SHUFFLE
20 CL$=CHR$(147):HOME$=CHR$(19):
   DOWN$=CHR$(17):ACROS$=CHR$(29)
30 H$=CHR$(115):C$=CHR$(120):
   D$=CHR$(122):S$=CHR$(97)
40 PRINT CL$;
50 X=10:Y=16:GOSUB 490
60 PRINT "SHUFFLE"
70 A=TI
80 IF TI<A+150 THEN GOTO 80
90 REM SET UP SCREEN DISPLAY
100 BRDER=4:SCREEN=0
110 PURPLE=5:YELLOW=8:RED$=CHR$(28):
    BLACK$=CHR$(144)
120 CASE$=CHR$(142):UNCASE$=CHR$(14)
130 COLOR BRDER,YELLOW
140 COLOR SCREEN,PURPLE
150 FOR I=2 TO 9
160 HR$=HR$+MID$(STR$(I),2)+H$
170 CB$=CB$+MID$(STR$(I),2)+C$
180 DI$=DI$+MID$(STR$(I),2)+D$
190 SP$=SP$+MID$(STR$(I),2)+S$
200 NEXT I
210 HR$="A"+H$+HR$+"T"+H$+"J"+H$+"Q"+H$+"K"+H$
220 CB$="A"+C$+CB$+"T"+C$+"J"+C$+"Q"+C$+"K"+C$
230 DI$="A"+D$+DI$+"T"+D$+"J"+D$+"Q"+D$+"K"+D$
240 SP$="A"+S$+SP$+"T"+S$+"J"+S$+"Q"+S$+"K"+S$
250 PACK$=HR$+CB$+DI$+SP$
260 PRINT CL$+BLACK$
270 PRINT "[RON ]              SHUFFLING                  [ROF
]"
```

```
280 FOR Z=1 TO 10:PRINT DOWN$:NEXT Z
290 PRINT "[RON ]                                                    [ROF
]";
300 SHUFFLED$=""
310 FOR I=1 TO 50
320 P=(INT(RND(0)*(53-I))+1)*2-1
330 SHUFFLED$=SHUFFLED$+MID$(PACK$,P,2)
340 PACK$=LEFT$(PACK$,P-1)+MID$(PACK$,P+2)
350 T$=SHUFFLED$+PACK$
360 PRINT HOME$+DOWN$+DOWN$+DOWN$+DOWN$+DOWN$
370 FOR J=1 TO 4
380 HN$=MID$(T$,(J-1)*26+1,26)
390 FOR K=0 TO 12
400 CD$=MID$(HN$,K*2+1,2)
410 S$=RIGHT$(CD$,1)
420 IF S$=H$ OR S$=D$ THEN PRINT RED$;CD$;" ";:GOTO 440
430 IF S$=C$ OR S$=S$ THEN PRINT BLACK$;CD$;" ";
440 NEXT K
450 PRINT:PRINT
460 NEXT J
470 NEXT I
480 END
490 PRINT HOME$;
500 IF X=1 THEN 520
510 FOR R=1 TO X:PRINT DOWN$;:NEXT R
520 IF Y=1 THEN 540
530 FOR C=1 TO Y:PRINT ACROS$;:NEXT C
540 RETURN
```

P77 Recipes

In this program, we have recorded some of our favorite recipes. These are based on recipes in "Favourite Family Cookbook" by Norma Macmillan, Octopus (1978).

Although we have included our own recipes, it would perhaps be better if you replaced our data statements with your own.

COMMANDS

Key in program and type RUN.
Follow the menus as presented.

```
10 REM PROGRAM FOR RECIPES
20 PRINT "[CS ]";
30 PRINT "[RON ]                RECIPES [ROF ]"
40 PRINT:PRINT:PRINT:PRINT
50 PRINT "      1. [RON ]HORS D'OEUVRES[ROF ]"
60 PRINT "      2. [RON ]SOUPS         [ROF ]"
70 PRINT "      3. [RON ]FISH          [ROF ]"
80 PRINT "      4. [RON ]MEAT          [ROF ]"
90 GOSUB 1000
100 IF CH<1 OR CH>4 THEN GOTO 10
110 RESTORE
120 REM SELECT SUB MENU
125 IF CH=1 THEN GOTO 185
130 FOR M=1 TO CH-1
140 READ T$
150 FOR I=1 TO 4
160 READ T1$
170 NEXT I
180 NEXT M
185 READ T$
190 PRINT "[CS ]":PRINT:PRINT:PRINT
200 PRINT TAB(20-LEN(T$)/2);CHR$(18);T$;CHR$(146)
210 PRINT:PRINT:PRINT
220 FOR M=1 TO 4
230 READ M$
240 PRINT TAB(5);M;" ";M$
250 NEXT M
260 IF CH=4 THEN GOTO 330
```

```
270 FOR M=CH+1 TO 4
280 READ S$
290 FOR N=1 TO 4
300 READ S$
310 NEXT N
320 NEXT M
330 REM WE HAVE NOW READ PAST ALL TITLE PAGES
340 C=CH
350 GOSUB 1000
360 REM NOW WE CALCULATE WHERE TO READ FROM
370 REM C=RECIPE TYPE:CH=CHOICE WITHIN TYPE
380 REM THEREFORE RECIPE NUMBER IS (C-1)*4+CH
390 REM EACH RECIPE IS HELD IN 21 DATA ITEMS
400 REM THUS WE HAVE TO READ PAST
410 REM ((C-1)*4+CH-1)*21 ITEMS TO GET
420 REM TO RECIPE
430 E=((C-1)*4+CH-1)*21
440 FOR I=1 TO E
450 READ S$
460 NEXT I
470 REM NOW WE ARE AT THE CORRECT RECIPE
480 READ T$
490 PRINT "[CS ]"; TAB(20-LEN(T$)/2); CHR$(18); T$;
CHR$(146): PRINT: PRINT
500 FOR I=1 TO 19
510 READ L$
520 PRINT L$
530 NEXT I
540 PRINT "PRESS ANY KEY TO RETURN TO MENU"
550 GET A$:IF A$="" THEN GOTO 550
560 RUN
999 END
1000 PRINT:PRINT:PRINT
1010 PRINT "ENTER THE APPROPRIATE NUMBER"
1020 PRINT "FOR THE RECIPE OF YOUR CHOICE ";
1030 INPUT CH
1040 RETURN
2000 DATA H O R S  D' O E V R E S,KIPPER PATE,EGG MAYONNAISE
2010 DATA MUSHROOMS A LA GRECQUE,GRILLED GRAPEFRUIT
2020 DATA S O U P S,BORSCH,PEA SOUP,OXTAIL SOUP,FRENCH ONION
SOUP
2030 DATA F I S H,TROUT WITH ALMONDS,SALMON STEAKS,FINNAN
HADDIE
2040 DATA SCALLOPS AND BACON
2050 DATA M E A T,NEAPOLITAN STEAK,WEINER SCHNITZEL,LAMB
CURRY,PORK'N'ORANGE
2999 REM HORS D'OEVRES
```

```
3000 REM K I P P E R  P A T E
3001 DATA ,K I P P E R  P A T E
3002 DATA 2 TBLESPOON UNSALTED BUTTER
3003 DATA 1/2 LB KIPPER FILLETS
3004 DATA 1 CUP CREAM CHEESE
3005 DATA 1/2 GARLIC CLOVE - CRUSHED
3006 DATA JUICE OF 1/2 LEMON
3007 DATA BLACK PEPPER,,,
3010 DATA MELT BUTTER IN FRYING PAN
3012 DATA ADD THE KIPPERS AND COOK TILL SOFT
3013 DATA REMOVE FISH AND COOL THEN FLAKE FISH
3014 DATA BLEND FISH CHEESE ETC UNTIL SMOOTH
3015 DATA SERVE CHILLED WITH PEPPER,,,,
3100 DATA E G G  M A Y O N N A I S E
3101 DATA 4 LARGE LETTUCE LEAVES
3102 DATA 8 SPRIGS WATERCRESS
3103 DATA 4 EGGS - HARD BOILED AND HALVED
3104 DATA 1 AND 1/4 CUPS OF MAYONNAISE
3105 DATA PINCH OF PAPRIKA,,,
3112 DATA PUT A LETTUCE LEAF ON EACH PLATE
3113 DATA ADD 2 SPRIGS OF WATERCRESS TO EACH
3114 DATA PLACE EGGS IN CENTER FLAT SIDE DOWN
3115 DATA POUR OVER MAYONNAISE AND ADD PAPRIKA,,,,,,,,
3200 DATA MUSHROOMS A LA GRECQUE
3201 DATA 1 AND 1/4 CUPS WATER
3202 DATA SMALL ONION CHOPPED
3203 DATA 1 TEASPOON TOMATO PASTE
3204 DATA SALT AND PEPPER TO TASTE
3205 DATA BOUQUET GARNI
3206 DATA LEMON JUICE TO TASTE
3207 DATA 3 CUPS OF SMALL BUTTON MUSHROOMS
3208 DATA PARSLEY TO GARNISH,,,
3209 DATA PUT ALL EXCEPT MUSHROOMS AND GARNISH
3210 DATA INTO SAUCEPAN AND SIMMER FOR 5 MINUTES
3211 DATA ADD MUSHROOMS AND SIMMER FOR 10 MINS
3212 DATA REMOVE MUSHROOMS
3213 DATA BOIL LIQUID UNTIL 3-4 TABLESPOONS LEFT
3214 DATA POUR OVER MUSHROOOMS AND SERVE WITH
3215 DATA PARSLEY,,
3300 DATA GRILLED GRAPEFRUIT
3301 DATA 2 LARGE GRAPEFRUIT - HALVED
3302 DATA 4 TEASPOONS OF MEDIUM SHERRY
3303 DATA 4 TABLESPOONS OF BROWN SUGAR
3304 DATA 1 TABLESPOON OF BUTTER - CUT INTO FOUR,,,,
3305 DATA LOOSEN SEGMENTS OF GRAPEFRUIT AND
3306 DATA REMOVE SEEDS. DRAIN HALVES ONTO KITCHEN
3307 DATA PAPER FOR A FEW MINUTES. PLACE
```

```
3308 DATA GRAPEFRUIT HALVES IN GRILL PAN CUT
3309 DATA SIDES UP. SPRINKLE EACH WITH SUGAR AND
3310 DATA SHERRY. TOP EACH WITH BUTTER AND GRILL
3311 DATA UNTIL BUBBLING.,,,,,
4000 DATA B O R S C H
4001 DATA 2 TABLESPOONS OF BUTTER
4002 DATA LARGE UNION - PEELED AND SLICED
4003 DATA LARGE CARROT - PEELED AND GRATED
4004 DATA 2 LARGE COOKED BEETROOTS - SLICED
4005 DATA 1/2 SMALL HEAD RED CABBAGE - SHREDDED
4006 DATA 1 TABLESPOON OF TOMATO PASTE
4007 DATA 1 TABLESPOON OF VINEGAR AND 1 OF SUGAR
4008 DATA SALT AND PEPPER TO TASTE
4009 DATA 5 CUPS OF BEEF STOCK
4010 DATA 1 AND 1/4 CUPS OF SOUR CREAM TO SERVE,,,
4011 DATA MELT BUTTER IN PAN AND COOK VEGETABLES
4012 DATA GENTLY FOR 5 MINUTES. STIR IN OTHER
4013 DATA INGREDIENTS AND SIMMER FOR 20-30 MINS
4014 DATA UNTIL VEG ARE TENDER. SERVE WITH CREAM.,,,
4100 DATA P E A  S O U P
4101 DATA 2 TABLESPOONS OF BUTTER AND 1 OF OIL
4102 DATA 1 ONION AND 2 CELERY STALKS - CHOPPED
4103 DATA 2 CUPS OF SHELLED PEAS
4104 DATA 1 SMALL HAM BONE
4105 DATA 2 CUPS OF CHICKEN STOCK
4106 DATA 1 TABLESPOON OF CORNFLOUR
4107 DATA 1 AND 1/4 CUPS OF MILK
4108 DATA SALT AND PEPPER TO TASTE,,,
4109 DATA MELT BUTTER AND OIL IN PAN - FRY CELERY
4110 DATA AND ONION. COOK PEAS FOR 5 MINS. ADD
4111 DATA BONE AND STOCK THEN BOIL FOR 40 MINS.
4112 DATA REMOVE BONE AND PUREE SOUP. DISSOLVE
4113 DATA FLOUR IN MILK AND ADD TO SOUP. SIMMER
4114 DATA UNTIL THICK.,,,
4200 DATA O X T A I L    S O U P
4201 DATA 4 TABLESPOONS OF OLIVE OIL
4202 DATA 4 AND 1/2 PINTS OF WATER
4203 DATA 1 OXTAIL - CUT INTO PIECES
4204 DATA 2 ONIONS AND 2 CARROTS - CHOPPED
4205 DATA 2 CELERY STALKS AND 1 TURNIP - CHOPPED
4206 DATA SALT AND PEPPER
4207 DATA ONE 14 OZ CAN OF TOMATOES
4208 DATA 2 TABLESPOONS OF FLOUR AND 4 OF SHERRY.,,,
4209 DATA BROWN OXTAIL AND FRY VEG TILL SOFT. ADD
4210 DATA TO WATER WITH SALT AND PEPPER. SIMMER
4211 DATA FOR 4 HOURS. REMOVE OXTAILS AND CUT
4212 DATA MEAT FROM BONES. REMOVE FAT FROM SOUP
```

```
4213 DATA AND RETURN MEAT. THICKEN WITH SHERRY
4214 DATA AND FLOUR. SERVE HOT.,,,
4300 DATA F R E N C H  O N I O N  S O U P
4301 DATA 2 TABLESPOONS BUTTER
4302 DATA 2 TABLESPOONS OF OLIVE OIL
4303 DATA 1 GARLIC CLOVE - CRUSHED
4304 DATA 3 LARGE SLICED ONIONS
4305 DATA 5 CUPS OF BEEF STOCK
4306 DATA SALT AND PEPPER
4307 DATA 8-12 SLICES OF FRENCH BREAD
4308 DATA 3/4 CUP OF CHEESE.,,,
4309 DATA FRY GARLIC AND ONION FOR 20 MINS - DO
4310 DATA NOT BROWN TOO MUCH. ADD TO STOCK WITH
4311 DATA SALT AND PEPPER. SIMMER FOR 30 MINS.
4312 DATA TOAST BREAD. SERVE SOUP HOT WITH TOAST
4313 DATA AND CHEESE.,,,,
5000 DATA T R O U T  W I T H  A L M O N D S
5001 DATA 1/4 CUP PLAIN FLOUR
5002 DATA SALT AND PEPPER
5003 DATA 4 LARGE TROUT - CLEANED
5004 DATA 1/3 CUP BUTTER
5005 DATA 1/2 CUP ALMOND FLAKES
5006 DATA 2 TEASPOONS OF LEMON JUICE
5007 DATA LEMON AND PARSLEY TO GARNISH,,,
5008 DATA MIX FLOUR WITH SALT AND PEPPER AND COAT
5009 DATA THE TROUT. MELT BUTTER IN PAN AND BROWN
5010 DATA TROUT QUICKLY ON BOTH SIDES. COOK TROUT
5011 DATA UNTIL TENDER. REMOVE FISH AND KEEP HOT.
5012 DATA COOK ALMONDS IN PAN UNTIL GOLDEN. STIR
5013 DATA IN LEMON JUICE. POUR LIQUID AND
5014 DATA ALMONDS OVER FISH AND GARNISH.,,,
5100 DATA S A L M O N  S T E A K S
5101 DATA 4 SALMON STEAKS
5102 DATA 1/4 CUP OF BUTTER CUT INTO FOUR
5103 DATA 4 HALF BAY LEAVES AND 4 ONION SLICES
5104 DATA 4 SLIVERS LEMON RIND AND 4 SPRIGS THYME
5105 DATA 4 PARLSEY SPRIGS
5106 DATA SALT AND PEPPER TO TASTE,,,,
5107 DATA DIVIDE INGREDIENTS INTO INDIVIDUAL
5108 DATA PORTIONS. WRAP EACH IN FOIL AND BAKE
5109 DATA IN A MODERATE OVEN (180 C:350 F GAS
5110 DATA MARK 4). FOR 15-20 MINUTES. SERVE
5111 DATA STEAKS HOT WITHOUT HERBS.,,,,,
5200 DATA F I N N A N  H A D D I E
5201 DATA 1 AND 1/2 LB SMOKED HADDOCK - CHOPPED
5202 DATA 1 AND 1/4 CUPS OF MILK
5203 DATA 2/3 CUP OF SINGLE CREAM
```

```
5204 DATA 2 TABLESPOONS BUTTER
5205 DATA PEPPER
5206 DATA 6 EGGS,,,
5207 DATA PLACE FISH IN BAKING DISH. PUT MILK AND
5208 DATA CREAM AND BUTTER IN PAN AND HEAT UNTIL
5209 DATA BUTTER IS MELTED. POUR OVER FISH. BAKE
5210 DATA IN MODERATE OVEN (180C : 350F : GAS 4)
5211 DATA FOR 20 MINS. POACH EGGS 5 MINS BEFORE
5212 DATA FISH IS READY. TOP FISH MIXTURE WITH
5213 DATA EGGS AND SERVE.,,,,
5300 DATA S C A L L O P S  A N D  B A C O N
5301 DATA 16-20 SHELLED SCALLOPS
5302 DATA SALT AND PEPPER AND LEMON JUICE
5303 DATA 16-20 SLICES OF BACON,,,,
5304 DATA SPRINKLE SCALLOPS WITH SALT AND PEPPER
5305 DATA AND LEMON JUICE. STRETCH BACON WITH
5306 DATA FLAT KNIFE. WRAP RASHER ROUND EACH
5307 DATA SCALLOP AND SECURE WITH COCKTAIL STICK.
5308 DATA GRILL SLOWLY TILL COOOKED - 5 MINS
5309 DATA AND SERVE WITH TARTARE SAUCE.,,,,,,,
6000 DATA N E A P O L I T A N  S T E A K
6001 DATA 4 SIRLOIN STEAKS
6002 DATA 3 CUPS PEELED AND CHOPPED TOMATOES
6003 DATA 2 TABLESPOONS OLIVE OIL
6004 DATA 2 CHOPPED GARLIC CLOVES
6005 DATA 1 TABLESPOON CHOPPED PARSLEY
6006 DATA 1/2 TEASPOON OREGANO
6007 DATA 1 TEASPOON OF SUGAR
6008 DATA SALT AND PEPPER.,,,
6009 DATA BOIL TOMATOES OIL GARLIC PARSLEY SUGAR
6010 DATA SALT AND PEPPER. SIMMER FOR 5 MINS.
6011 DATA GRILL STEAKS AND POUR SAUCE OVER.
6012 DATA SERVE HOT.,,,,,
6100 DATA W I E N E R  S C H N I T Z E L
6101 DATA 4 VEAL ESCALOPES POUNDED THIN
6102 DATA JUICE OF 2 LEMONS
6103 DATA SALT AND PEPPER AND LARGE BEATEN EGG
6104 DATA 1 CUP OF DRY BREADCRUMBS
6105 DATA 3 TABLESPOONS BUTTER
6106 DATA LEMON WEDGES TO GARNISH,,,
6107 DATA MARINATE VEAL IN LEMON WITH SALT AND
6108 DATA PEPPER FOR 1 HOUR. DIP VEAL IN EGG AND
6109 DATA COAT WITH BREADCRUMBS. FRY VEAL IN
6110 DATA BUTTER TILL GOLDEN BROWN.,,,,,,,
6200 DATA L A M B  C U R R Y
6201 DATA 2LB LEAN LAMB CUBED
6202 DATA 2/3 CUP YOGHURT
```

```
6203 DATA 2 TEASPOONS OF GARAM MASALA
6204 DATA 1 TABLESPOON CURRY POWDER
6205 DATA 2 TABLESPOONS OF BUTTER
6206 DATA 2 TABLESPOONS OF OLIVE OIL
6207 DATA 2 CHOPPED ONIONS
6208 DATA 1 CRUSHED GARLIC CLOVE
6209 DATA SALT PEPPER AND LEMON JUICE TO TASTE
6210 DATA 2/3 CUP DRIED FRUIT
6211 DATA 2 TABLESPOONS OF ALMONDS,
6212 DATA MIX YOGHURT WITH GARAM MASALA AND CURRY
6213 DATA ADD LAMB CUBES AND MARINATE FOR 4 HOURS
6214 DATA FRY ONIONS AND GARLIC TILL SOFT. STIR
6215 DATA IN LANB AND YOGHURT WITH SALT:PEPPER &
6216 DATA LEMON JUICE. COOK FOR 5 MINS THEN ADD
6217 DATA FRUIT AND ALMONDS. SIMMER TILL TENDER.,,
6300 DATA P O R K  'N'  O R A N G E
6301 DATA 1/4 CUP PLAIN FLOUR
6302 DATA 1 AND 1/2LB PORK FILLET CUT INTO CUBES
6303 DATA 2 TABLESPOONS OF BUTTER
6304 DATA 1 SMALL CHOPPED ONION
6305 DATA 1 CHOPPED GREEN PEPPER- NO PITH OR SEED
6306 DATA GRATED RIND AND JUICE OF 2 ORANGES
6307 DATA 1 TABLESPOON OF WORCESTERSHIRE SAUCE
6308 2/3 CUPS OF BEEF STOCK
6309 DATA 1 PEELED SEGMENTED ORANGE,,
6310 DATA MIX FLOUR SALT AND PEPPER IN BAG. ADD
6311 DATA PORK CUBES AND SHAKE TO COAT. FRY ONION
6312 DATA AND PEPPER IN BUTTER TILL SOFT. BROWN
6313 DATA PORK. STIR IN ORANGE RIND AND JUICE ADD
6314 DATA STOCK AND SAUCE. SIMMER FOR 10 MINS.
6315 DATA ADD ORANGE SEGS AND COOK FOR 2 MINS.
6316 DATA SERVE HOT,,,
6317 REM END OF PROGRAM DATA
```

P78 Kitchen timer

A useful program for the kitchen here - it lets you know how long it will be before a meal is ready. At the end of the period specified an alarm sounds.

COMMANDS

Key in the program and RUN.
Enter the delay required when prompted.
Press any key to stop the alarm.

```
10 REM KITCHEN TIMER
20 REM *************
30 REM
40 UP$=CHR$(145):REM UP CURSOR
50 S$=CHR$(32)
60 FOR N=0 TO 2:S$=S$+S$+S$:NEXT:REM S$ CONTAINS 27 SPACES
70 COLOR 0,1:REM BLACK SCREEN
80 COLOR 1,6,6:REM GREEN INK
90 COLOR 4,1:REM BLACK BORDER
100 REM**************************
110 SCNCLR:PRINT:PRINT:PRINT
120 PRINT TAB(14)"KITCHEN TIMER"
130 PRINT TAB(14)"++++++++++++"
140 PRINT:PRINT
150 PRINT TAB(5)"WHAT SETTING DO YOU REQUIRE?"
160 PRINT
170 DO
180 :  PRINT TAB(11):INPUT"HOURS";H%
190 :  IF H%<0 OR H%>23 THEN PRINT UP$+S$+UP$:F=1:ELSE F=0
200 LOOP UNTIL F=0
210 PRINT
220 DO
230 :  PRINT TAB(11):INPUT"MINUTES";M%
240 :  IF M%<0 OR M%>59 THEN PRINT UP$+S$+UP$:F=1:ELSE F=0
250 LOOP UNTIL F=0
260 PRINT
270 DO
280 :  PRINT TAB(11):INPUT"SECONDS";S%
290 :  IF S%<0 OR S%>59 THEN PRINT UP$+S$+UP$:F=1:ELSE F=0
300 LOOP UNTIL F=0
```

```
310 TI$="000000":REM RESET CLOCK
320 DL=H%*3600+M%*60+S%
330 SCNCLR
340 CHAR,12,9,"TIME REMAINING"
350 REM**************************
360 REM PRINT TIME REMAINING
370 DO
380 :  A$=STR$(H%):IF LEN(A$)=2 THEN A$="0"+RIGHT$(A$,1)
390 :  IF LEN(STR$(H%))=3 THEN A$=RIGHT$(STR$(H%),2)
400 :  B$=STR$(M%):IF LEN(B$)=2 THEN B$="0"+RIGHT$(B$,1)
410 :  IF LEN(STR$(M%))=3 THEN B$=RIGHT$(STR$(M%),2)
420 :  C$=STR$(S%):IF LEN(C$)=2 THEN C$="0"+RIGHT$(C$,1)
430 :  IF LEN(STR$(S%))=3 THEN C$=RIGHT$(STR$(S%),2)
440 :  D$=A$+":"+B$+":"+C$
450 :  CHAR,15,11,D$
460 :  REM**************************
470 :  REM CALCULATE DELAY
480 :  T=DL-INT(TI/60)
490 :  H%=T/3600:TM=T-H%*3600
500 :  M%=TM/60:
510 :  S%=TM-M%*60
520 LOOP UNTIL T<=0
530 REM**************************
540 REM TIME UP
550 SCNCLR
560 CHAR,15,9,"TIME UP"
570 CHAR,8,14,"PRESS ANY KEY TO STOP"
580 REM**************************
590 REM ALARM
600 VOL 8
610 DO
620 :  SOUND 1,917,8
630 :  FOR N=0 TO 1
640 :     SOUND 2,881,2
650 :     FOR D=0 TO 45:NEXT
660 :     SOUND 2,953,2
670 :     FOR D=0 TO 45:NEXT
680 :  NEXT
690 :  GET W$
700 LOOP UNTIL LEN(W$)>0
710 END
720 REM**************************
730 REM**************************
```

P79 Counting

This program could be useful for the very young schoolchild. It displays up to nine monsters on the screen. The user is required to count the monsters and press the appropriate numeric key. The RETURN key is not used.

The program could be expanded so that several groups of items (say monsters, flowers and automobiles) appear on the screen at the same time and the user is asked to count the numbers in one particular group.

COMMANDS

Key in the program and RUN.
Stop the program by pressing the space bar as instructed.

```
10 REM COUNTING
20 REM ********
30 REM
40 POKE 52,47:POKE 56,47:CLR:REM RESERVE RAM
50 MC=12032:REM START OF MACHINE CODE ROUTINE
60 RA=12288:REM START OF CHARACTER RAM
70 CC=65298:REM CONTROLS CHARACTER LOCATION
80 SC=3072:REM START OF SCREEN MEMORY
90 COLOR 0,1:REM BLACK SCREEN
100 COLOR 1,8,7:REM YELLOW INK
110 COLOR 4,7,3:REM BLUE BORDER
120 REM**************************
130 SCNCLR:PRINT:PRINT
140 PRINT TAB(18)"WAIT"
150 FOR N=0 TO 33
160 :  READ A:POKE MC+N,A
170 NEXT:REM LOAD MACHINE CODE ROUTINE
180 SYS MC:REM TRANSFER CHARACTER SET
190 R1=PEEK(CC)
200 R2=PEEK(CC+1)
210 POKE CC,R1 AND 251:REM CHARACTER CONTROL TO RAM
220 POKE CC+1,(R2 AND 3) OR 48:REM POINT TO CHARACTERS
230 REM**************************
240 REM CHANGE CHARACTER SHAPES
250 FOR N=33 TO 36
260 :  FOR K=0 TO 7
```

```
270 :      READ A
280 :      POKE RA+K+N*8,A
290 :   NEXT
300 NEXT
310 REM*****************************
320 REM PRINT MONSTERS IN BACKGROUND COLOR
330 DO
340 :   SCNCLR
350 :   MN%=9*RND(1):REM NUMBER OF MONSTERS WILL BE MN%+1
360 :   FOR K=0 TO MN%
370 :      DO
380 :         F=0
390 :         X(K)=1+INT(37*RND(1))
400 :         Y(K)=1+INT(18*RND(1))
410 :         L=SC+40*Y(K)+X(K)
420 :         M=SC+40*(Y(K)+1)+X(K)
430 :         IF PEEK(L)<>32 OR PEEK(L+1)<>32 THEN F=1
440 :         IF PEEK(M)<>32 OR PEEK(M+1)<>32 THEN F=1
450 :      LOOP UNTIL F=0
460 :      CHAR 0,X(K),Y(K),CHR$(33)
470 :      CHAR 0,X(K)+1,Y(K),CHR$(34)
480 :      CHAR 0,X(K),Y(K)+1,CHR$(35)
490 :      CHAR 0,X(K)+1,Y(K)+1,CHR$(36)
500 :   NEXT
510 :   REM************************
520 :   REM COLOR THEM IN
530 :   FOR K=0 TO MN%
540 :      CHAR 1,X(K),Y(K),CHR$(33)
550 :      CHAR 1,X(K)+1,Y(K),CHR$(34)
560 :      CHAR 1,X(K),Y(K)+1,CHR$(35)
570 :      CHAR 1,X(K)+1,Y(K)+1,CHR$(36)
580 :   NEXT
590 :   REM************************
600 :   CHAR,10,23,"HOW MANY MONSTERS?"
610 :   VOL 8
620 :   M$=RIGHT$(STR$(MN%+1),1)
630 :   DO
640 :      DO
650 :         GETKEY A$
660 :      LOOP UNTIL ASC(A$)>48 AND ASC(A$)<58
670 :      IF A$<>M$ THEN SOUND 3,1000,50
680 :   LOOP UNTIL A$=M$
690 :   SOUND 1,810,20
700 :   SOUND 2,596,20
710 :   SOUND 1,810,20
720 :   SOUND 1,917,20
730 :   FOR N=0 TO 100:GET B$:IF B$=" " THEN N=100:NEXT
```

```
740 LOOP UNTIL B$=" ":REM ONE SPACE
750 REM TO STOP PROGRAM AND RESTORE CHARACTERS
760 REM HOLD THE SPACE BAR DOWN AS THE "CORRECT"
770 REM SOUND ENDS.
780 POKE CC,R1
790 POKE CC+1,R2
800 END
810 REM*****************************
820 REM*****************************
830 REM
840 REM          ********
850 REM          *      *
860 REM          *      *
870 REM          * DATA *
880 REM          *      *
890 REM          *      *
900 REM          ********
910 REM
920 REM*****************************
930 REM*****************************
940 REM DATA FOR MACHINE CODE ROUTINE
950 DATA 169,0,133,3,133,5,169,48,133,4,169,208
960 DATA 133,6,160,0,177,5,145,3,200,208,249
970 DATA 230,4,230,6,169,216,197,6,208,239,96
980 REM*****************************
990 REM*****************************
1000 REM DATA FOR CHARACTERS
1010 DATA 1,1,13,13,34,34,42,42
1020 DATA 64,64,80,80,136,136,168,168
1030 DATA 4,4,49,48,192,192,192,192
1040 DATA 16,16,76,12,3,3,3,3
1050 REM*************************
1060 REM*************************
```

P80 School report

This program prepares a school report for a student. The program could be developed to store data on a tape or disk file and handle more than one student.

COMMANDS

Key in the program and RUN.
Enter details as requested.

```
10 REM PROGRAM - SCHOOL REPORT
20 PRINT "[CS CD CD CD CD CD CD CD CD CD CD ]"
30 PRINT "            SCHOOL REPORT"
40 A=TI
50 IF TI<A+100 THEN GOTO 50
60 GOSUB 1000
70 PRINT "    PLEASE ENSURE THAT YOUR PRINTER"
80 PRINT "    HAS BEEN SET UP CORRECTLY."
90 PRINT:PRINT:PRINT
100 PRINT "    PRESS ANY KEY WHEN READY"
110 GET A$:IF A$="" THEN 110
120 GOSUB 1000
130 PRINT "WHAT IS NAME OF SCHOOL ";
140 INPUT SC$
150 INPUT "ENTER STUDENT'S NAME ";N$
160 OPEN 1,4:REM OPEN PRINTER
170 PRINT#1,TAB((60-LEN(SC$))/2);SC$
180 PRINT#1:PRINT#1:PRINT#1
190 INPUT "SESSION";S$
200 PRINT#1,"SESSION - ";S$
210 PRINT#1,"NAME - ";N$
220 PRINT#1:PRINT#1:PRINT#1
230 PRINT#1,"    SUBJECT   !ATTEND!GRADE!POSITION!
COMMENT"
240 PRINT#1,SPC(14);"! P  A !     !IN CLASS!"
250
PRINT#1,"-------------!------!-----!--------!--------------
-------"
260 REM LOOP BACK POINT
265 GOSUB 1000
270 INPUT "SUBJECT (999  TO END)";SB$
```

```
280 IF SB$="999" THEN 480
290 INPUT "POSSIBLE ATTENDANCE ";P$
300 INPUT "ACTUAL ATTENDANCE ";A$
310 INPUT "GRADE ";GR$
320 INPUT "POSITION IN CLASS ";PO$
330 INPUT "TEACHER'S COMMENT";CO$
340 IF LEN(SB$)>14 THEN SB$=LEFT$(SB$,14):GOTO 360
350 FOR Z=LEN(SB$) TO 13:SB$=SB$+" ":NEXT Z
360 IF LEN(P$)>3 THEN P$=LEFT$(P$,3):GOTO 380
370 FOR Z=LEN(P$) TO 2:P$=P$+" ":NEXT Z
380 IF LEN(A$)>3 THEN A$=LEFT$(A$,3):GOTO 400
390 FOR Z=LEN(A$) TO 2:A$=A$+" ":NEXT Z
400 IF LEN(GR$)>5 THEN GR$=LEFT$(GR$,5):GOTO 420
410 FOR Z=LEN(GR$) TO 4:GR$=GR$+" ":NEXT Z
420 IF LEN(PO$)>8 THEN PO$=LEFT$(PO$,8):GOTO 440
430 FOR Z=LEN(PO$) TO 7:PO$=PO$+" ":NEXT Z
440 IF LEN(CO$)>20 THEN CO$=LEFT$(CO$,20):GOTO 460
450 FOR Z=LEN(CO$) TO 19:CO$=CO$+" ":NEXT Z
460 PRINT#1,SB$+"!"+P$+"!"+A$+"!"+GR$+"!"+PO$+"!"+CO$
470 GOTO 265
480 GOSUB 1000
490 PRINT#1:PRINT#1:PRINT#1
500 INPUT "OVERALL ASSESSMENT ";OV$
510 PRINT#1,"OVERALL ASSESSMENT"
520 PRINT#1,"=================="
530 PRINT#1
540 PRINT#1,OV$
550 PRINT#1:PRINT#1:PRINT#1
560 PRINT#1,"SIGNATURE OF PARENT OR GARDIAN"
570 PRINT#1,"----------------------------"
580 GOSUB 1000
590 INPUT "ANOTHER (Y/N) ";R$
600 IF R$="Y" THEN GOTO 10
610 END
1000 PRINT "[CS ]";
1010 PRINT "[RON ]            SCHOOL REPORT
[ROF ]"
1020 PRINT:PRINT:PRINT
1030 RETURN
```

P81 French tutorial

The next two programs give the French and German implementations of a language vocabulary tutorial. In both cases the data are in the form of word pairs which may be inserted by the teacher as data statements. The last item of data must be EOF.

The programs give the student up to three attempts at each word. After the tutorial is finished statistics are returned.

The programs could be extended so that several alternative answers may be accepted for some words.

COMMANDS

Key in the program(s) and RUN.
Follow the instructions.

```
10 REM FRENCH TUTORIAL
20 REM ***************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,3:REM BLUE INK
60 COLOR 4,10,5:REM BROWN BORDER
70 SCNCLR:PRINT:PRINT
80 PRINT TAB(12)"FRENCH TUTORIAL"
90 PRINT TAB(12)"================"
100 PRINT:PRINT
110 PRINT TAB(4)"THIS PROGRAM TESTS YOUR KNOWLEDGE"
120 PRINT TAB(4)"OF FRENCH VOCABULARY. ENGLISH"
130 PRINT TAB(4)"WORDS ARE PUT ON THE SCREEN ONE AT"
140 PRINT TAB(4)"A TIME AND YOU ARE ASKED FOR THE"
150 PRINT TAB(4)"FRENCH EQUIVALENTS. YOU ARE"
160 PRINT TAB(4)"ALLOWED THREE ATTEMPTS AT EACH"
170 PRINT TAB(4)"WORD."
180 PRINT:PRINT
190 PRINT TAB(8)CHR$(18)" PRESS ANY KEY TO START "
200 GETKEY A$
210 REM*******************************
220 DO
230 :  RESTORE
```

```
240 :  FOR N=1 TO 4:R(N)=0:NEXT:T$="":Y$=""
250 :  REM*******************************
260 :     DO
270 :     READ E$
280 :     FL=0
290 :     DO WHILE E$<>"EOF" AND FL=0
300 :        READ F$
310 :        F=2
320 :        FOR N=1 TO 3
330 :           SCNCLR:PRINT:PRINT
340 :           PRINT TAB(14)"ATTEMPT";N
350 :           PRINT TAB(14)"---------"
360 :           PRINT:PRINT
370 :           PRINT TAB(7)"TYPE IN THE FRENCH WORD"
380 :           PRINT TAB(7)"THEN PRESS THE RETURN KEY."
390 :           PRINT:PRINT
400 :           PRINT TAB(7)"ENGLISH WORD IS - "E$
410 :           PRINT:PRINT
420 :           PRINT TAB(8):INPUT"FRENCH WORD IS ";T$
430 :           IF T$=F$ THEN R(N)=R(N)+1:F=1:N=3
440 :        NEXT
450 :        ON F GOSUB 800,860
460 :        FOR DE=0 TO 2000:NEXT:REM DELAY
470 :        FL=1
480 :     LOOP
490 :  LOOP UNTIL E$="EOF"
500 :  REM*******************************
510 :  REM FINISH
520 :  SCNCLR:PRINT:PRINT
530 :  PRINT TAB(8)"NUMBER CORRECT AT FIRST"
540 :  PRINT TAB(8)"ATTEMPT WAS";R(1)
550 :  PRINT
560 :  PRINT TAB(8)"NUMBER CORRECT AT SECOND"
570 :  PRINT TAB(8)"ATTEMPT WAS";R(2)
580 :  PRINT
590 :  PRINT TAB(8)"NUMBER CORRECT AT THIRD"
600 :  PRINT TAB(8)"ATTEMPT WAS";R(3)
610 :  PRINT
620 :  PRINT TAB(8)"NUMBER OF UNKNOWN"
630 :  PRINT TAB(8)"ANSWERS WAS";R(4)
640 :  PRINT:PRINT
650 :  PRINT TAB(10):INPUT"ANOTHER GO (Y/N)";Y$
660 LOOP UNTIL ASC(Y$)<>89
670 END
680 REM*******************************
690 REM*******************************
700 REM
```

```
710 REM        ***************
720 REM        *             *
730 REM        * SUBROUTINES *
740 REM        *             *
750 REM        ***************
760 REM
770 REM********************************
780 REM********************************
790 REM CORRECT ANSWER
800 PRINT:PRINT
810 PRINT TAB(15)CHR$(18)" CORRECT "
820 RETURN
830 REM********************************
840 REM********************************
850 REM WRONG ANSWER
860 R(4)=R(4)+1
870 PRINT:PRINT
880 PRINT TAB(7)"CORRECT ANSWER IS - "F$
890 FOR K=1 TO LEN(F$):PRINT TAB(26+K)"-";:NEXT
900 RETURN
910 REM********************************
920 REM********************************
930 REM
940 REM  PUT IN AS MANY ENGLISH WORDS AS
950 REM  YOU WISH (EACH FOLLOWED BY ITS
960 REM  FRENCH EQUIVALENT) 1N LINES
970 REM  2000 ONWARDS. TERMINATE THE
980 REM  DATA WITH EOF AS SHOWN. A FEW
990 REM  SIMPLE WORDS ARE GIVEN AS
1000 REM EXAMPLES.
1010 REM
1020 REM********************************
1030 REM********************************
1040 REM
1050 REM           ********
1060 REM           *      *
1070 REM           * DATA *
1080 REM           *      *
1090 REM           ********
1100 REM
1110 REM********************************
2000 DATA YES,OUI,NO,NON,END,FIN,EOF
```

P82 German tutorial

```
10 REM GERMAN TUTORIAL
20 REM ***************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,3:REM BLUE INK
60 COLOR 4,10,5:REM BROWN BORDER
70 SCNCLR:PRINT:PRINT
80 PRINT TAB(12)"GERMAN TUTORIAL"
90 PRINT TAB(12)"================"
100 PRINT:PRINT
110 PRINT TAB(4)"THIS PROGRAM TESTS YOUR KNOWLEDGE"
120 PRINT TAB(4)"OF GERMAN VOCABULARY. ENGLISH"
130 PRINT TAB(4)"WORDS ARE PUT ON THE SCREEN ONE AT"
140 PRINT TAB(4)"A TIME AND YOU ARE ASKED FOR THE"
150 PRINT TAB(4)"GERMAN EQUIVALENTS. YOU ARE"
160 PRINT TAB(4)"ALLOWED THREE ATTEMPTS AT EACH"
170 PRINT TAB(4)"WORD."
180 PRINT:PRINT
190 PRINT TAB(8)CHR$(18)" PRESS ANY KEY TO START "
200 GETKEY A$
210 REM*******************************
220 DO
230 :  RESTORE
240 :  FOR N=1 TO 4:R(N)=0:NEXT:T$="":Y$=""
250 :  REM*******************************
260 :     DO
270 :     READ E$
280 :     FL=0
290 :     DO WHILE E$<>"EOF" AND FL=0
300 :        READ G$
310 :        F=2
320 :        FOR N=1 TO 3
330 :           SCNCLR:PRINT:PRINT
340 :           PRINT TAB(14)"ATTEMPT";N
350 :           PRINT TAB(14)"---------"
360 :           PRINT:PRINT
370 :           PRINT TAB(7)"TYPE IN THE GERMAN WORD"
380 :           PRINT TAB(7)"THEN PRESS THE RETURN KEY."
390 :           PRINT:PRINT
400 :           PRINT TAB(7)"ENGLISH WORD IS - "E$
410 :           PRINT:PRINT
420 :           PRINT TAB(8):INPUT"GERMAN WORD IS ";T$
430 :           IF T$=G$ THEN R(N)=R(N)+1:F=1:N=3
```

```
440 :         NEXT
450 :         ON F GOSUB 800,860
460 :         FOR DE=0 TO 2000:NEXT:REM DELAY
470 :         FL=1
480 :      LOOP
490 :   LOOP UNTIL E$="EOF"
500 :   REM*******************************
510 :   REM FINISH
520 :   SCNCLR:PRINT:PRINT
530 :   PRINT TAB(8)"NUMBER CORRECT AT FIRST"
540 :   PRINT TAB(8)"ATTEMPT WAS";R(1)
550 :   PRINT
560 :   PRINT TAB(8)"NUMBER CORRECT AT SECOND"
570 :   PRINT TAB(8)"ATTEMPT WAS";R(2)
580 :   PRINT
590 :   PRINT TAB(8)"NUMBER CORRECT AT THIRD"
600 :   PRINT TAB(8)"ATTEMPT WAS";R(3)
610 :   PRINT
620 :   PRINT TAB(8)"NUMBER OF UNKNOWN"
630 :   PRINT TAB(8)"ANSWERS WAS";R(4)
640 :   PRINT:PRINT
650 :   PRINT TAB(10):INPUT"ANOTHER GO (Y/N)";Y$
660 LOOP UNTIL ASC(Y$)<>89
670 END
680 REM*******************************
690 REM*******************************
700 REM
710 REM        ***************
720 REM        *             *
730 REM        * SUBROUTINES *
740 REM        *             *
750 REM        ***************
760 REM
770 REM*******************************
780 REM*******************************
790 REM CORRECT ANSWER
800 PRINT:PRINT
810 PRINT TAB(15)CHR$(18)" CORRECT "
820 RETURN
830 REM*******************************
840 REM*******************************
850 REM WRONG ANSWER
860 R(4)=R(4)+1
870 PRINT:PRINT
880 PRINT TAB(7)"CORRECT ANSWER IS - "G$
890 FOR K=1 TO LEN(G$):PRINT TAB(26+K)"-";:NEXT
900 RETURN
```

```
910 REM******************************
920 REM******************************
930 REM
940 REM  PUT IN AS MANY ENGLISH WORDS AS
950 REM  YOU WISH (EACH FOLLOWED BY ITS
960 REM  GERMAN EQUIVALENT) 1N LINES
970 REM  2000 ONWARDS. TERMINATE THE
980 REM  DATA WITH EOF AS SHOWN. A FEW
990 REM  SIMPLE WORDS ARE GIVEN AS
1000 REM EXAMPLES.
1010 REM
1020 REM******************************
1030 REM******************************
1040 REM
1050 REM          ********
1060 REM          *      *
1070 REM          * DATA *
1080 REM          *      *
1090 REM          ********
1100 REM
1110 REM******************************
2000 DATA BLACK,SCHWARTZ,THERE,DORT,RED,ROT,EOF
```

P83 History tutorial

This program demonstrates the use of multiple choice questions to test history knowledge. Three questions are given as examples; one on American history, one on British history and one on ancient history. Up to two attempts at each question are allowed and statistics are compiled for the teacher's use.

The teacher may enter any number of questions (to the limit allowed by machine memory) using the pattern given in the examples. The final data item must be EOF.

This program may be adapted for any subject to which objective testing is applicable.

COMMANDS

Key in the program and RUN.
Follow the instructions.
The program uses keys 1 to 4 to select answers. The RETURN key is not used.

```
10 REM HISTORY TUTORIAL
20 REM ****************
30 REM
40 COLOR 0,7,3:REM BLUE SCREEN
50 COLOR 1,2:REM WHITE INK
60 COLOR 4,7,3:REM BLUE BORDER
70 REM**************************
80 SCNCLR
90 PRINT TAB(10)CHR$(18)" HISTORY TUTORIAL "
100 CHAR,0,5,""
110 PRINT TAB(4)"THIS TUTORIAL USES MULTIPLE"
120 PRINT TAB(4)"CHOICE QUESTIONS.  AFTER EACH"
130 PRINT TAB(4)"QUESTION YOU WILL BE GIVEN"
140 PRINT TAB(4)"FOUR POSSIBLE ANSWERS, ONLY"
150 PRINT TAB(4)"ONE OF WHICH IS CORRECT."
160 PRINT
170 PRINT TAB(4)"CHOOSE YOUR ANSWER BY PRESSING"
180 PRINT TAB(4)"ONE OF THE KEYS 1 TO 4."
190 CHAR,0,20,""
```

```
200 PRINT TAB(6)"PRESS ANY KEY TO CONTINUE"
210 GETKEY P$
220 REM*****************************
230 DO
240 :  RESTORE
250 :  DO
260 :     SCNCLR
270 :     READ A$
280 :     R=1
290 :     DO WHILE A$<>"EOF" AND R=1
300 :        READ B$,C$,D$,E$,F$
310 :        CHAR,0,6,""
320 :        PRINT A$
330 :        PRINT:PRINT
340 :        PRINT TAB(4)"1. "B$
350 :        PRINT TAB(4)"2. "C$
360 :        PRINT TAB(4)"3. "D$
370 :        PRINT TAB(4)"4. "E$
380 :        FOR N=1 TO 2
390 :           CHAR,10,2,"ATTEMPT "+STR$(N)
400 :           DO
410 :              GETKEY G$
420 :           LOOP UNTIL ASC(G$)>48 AND ASC(G$)<53
430 :           IF G$=F$ THEN CHAR,13,20,"CORRECT"
440 :           CA=0
450 :           IF G$=F$ THEN R(N)=R(N)+1:N=2:CA=1
460 :        NEXT
470 :        F=VAL(F$)
480 :        IF CA=0 THEN R(3)=R(3)+1
490 :        IF CA=0 THEN ON F GOSUB 710,750,790,830
500 :        R=0
510 :     LOOP
520 :     FOR N=0 TO 1500:NEXT
530 :  LOOP UNTIL A$="EOF"
540 :  REM**************************
550 :  SCNCLR
560 :  PRINT:PRINT
570 :  PRINT"NUMBER CORRECT AT FIRST ATTEMPT WAS";R(1)
580 :  PRINT
590 :  PRINT"NUMBER CORRECT AT SECOND ATTEMPT WAS";R(2)
600 :  PRINT
610 :  PRINT"NUMBER OF UNKNOWN ANSWERS WAS";R(3)
620 :  CHAR,8,15,""
630 :  FOR N=1 TO 3:R(N)=0:NEXT
640 :  Y$="N"
650 :  INPUT "WANT ANOTHER GO";Y$
660 LOOP UNTIL ASC(Y$)<>89
```

```
670 END
680 REM******************************
690 REM******************************
700 REM FIRST ANSWER CORRECT
710 CHAR,7,18,"ANSWER IS "+B$
720 RETURN
730 REM******************************
740 REM SECOND ANSWER CORRECT
750 CHAR,7,18,"ANSWER IS "+C$
760 RETURN
770 REM******************************
780 REM THIRD ANSWER CORRECT
790 CHAR,7,18,"ANSWER IS "+D$
800 RETURN
810 REM******************************
820 REM FOURTH ANSWER CORRECT
830 CHAR,7,18,"ANSWER IS "+E$
840 RETURN
850 REM******************************
860 REM******************************
870 REM PUT AS MANY QUESTIONS AS YOU WISH
880 REM (EACH FOLLOWED BY FOUR ANSWERS AND
890 REM A KEY) IN LINES 960 ONWARDS.
900 REM
910 REM TERMINATE THE DATA BY EOF.
920 REM
930 REM  A FEW SIMPLE EXAMPLES ARE GIVEN.
940 REM******************************
950 REM******************************
960 DATA DATE OF AMERICAN DECLARATION OF
        INDEPENDENCE WAS JULY 4TH:
970 DATA 1766,1767,1776,1777,3
980 REM******************************
990 DATA WHICH BRITISH MONARCH DIED IN 1649?
1000 DATA CHARLES II,JAMES I,JAMES II,CHARLES I,4
1010 REM******************************
1020 DATA WHICH OF THE FOLLOWING BATTLES WAS
      FOUGHT IN 216 BC?
1030 DATA CANNAE,MARATHON,SALAMIS,LAKE TRAMENE,1
1040 REM******************************
1050 DATA EOF
1060 REM******************************
1070 REM******************************
```

P84 Number base conversion

This is a very useful program for work in a computing laboratory. It allows you to convert numbers from one base to another.

COMMANDS

Key in the program and RUN.
Select the conversion you require.

```
10 REM NUMBER BASE CONVERSION
20 REM **********************
30 REM
40 COLOR 0,11,5:REM YELLOW SCREEN
50 COLOR 1,4,7:REM CYAN INK
60 COLOR 4,7,4:REM BLUE BORDER
70 SCNCLR
80 CHAR,9,2,"NUMBER BASE CONVERSION"
90 CHAR,9,3,"**********************"
100 CHAR,4,6,"THIS PROGRAM DEALS WITH POSITIVE"
110 CHAR,4,7,"INTEGER NUMBERS ONLY."
120 PRINT
130 GOSUB 1530:REM ANYKEY
140 REM************************
150 REM DISPLAY MENU
160 DO
170 :  SCNCLR
180 :  CHAR,8,3,"1. HEX TO DECIMAL"
190 :  CHAR,8,5,"2. DECIMAL TO HEX"
200 :  CHAR,8,7,"3. BINARY TO DECIMAL"
210 :  CHAR,8,9,"4. DECIMAL TO BINARY"
220 :  CHAR,8,11,"5. ENDS PROGRAM"
230 :  CHAR,8,14,"ENTER 1,2,3,4 OR 5"
240 :  CHAR,8,16,""
250 :  INPUT"WHAT IS YOUR SELECTION";X$
260 :  A%=ASC(X$)-48
270 :  ON A% GOSUB400,680,1010,1220
280 LOOP UNTIL A%=5
290 END
300 REM************************
310 REM************************
```

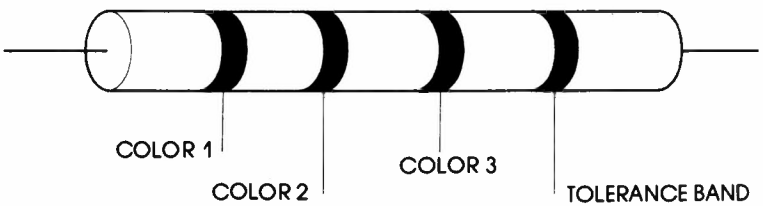
```
320 REM
330 REM       ***************
340 REM       *             *
350 REM       * SUBROUTINES *
360 REM       *             *
370 REM       ***************
380 REM
390 REM HEX TO DECIMAL
400 DO
410 :  SCNCLR
420 :  CHAR,12,3,"HEX TO DECIMAL"
430 :  CHAR,12,4,"**************"
440 :  CHAR,4,7,"PLEASE USE CAPITAL LETTERS FOR"
450 :  CHAR,4,8,"HEX SYMBOLS A TO F."
460 :  CHAR,4,11,""
470 :  INPUT"WHAT IS THE HEX NUMBER";H$
480 :  DC=0
490 :  FOR N=1 TO LEN(H$)
500 :     HX=ASC(MID$(H$,N,1))
510 :     BAD=1
520 :     IF HX>47 AND HX<58 THEN HX=HX-48:BAD=0
530 :     IF HX>64 AND HX<71 THEN HX=HX-55:BAD=0
540 :     IF BAD=1 THEN N=LEN(H$)
550 :     D=HX*16^(LEN(H$)-N)
560 :     DC=DC+D
570 :  NEXT
580 LOOP WHILE BAD=1
590 REM DEC FUNCTION NOT USED SO THAT VALUES
600 REM GREATER THAN $FFFF MAY BE ACCEPTED
610 CHAR,4,14,"DECIMAL NUMBER IS"
620 PRINT DC
630 GOSUB 1530:REM ANYKEY
640 RETURN
650 REM**********************
660 REM**********************
670 REM DECIMAL TO HEX
680 DO
690 :  SCNCLR
700 :  CHAR,12,3,"DECIMAL TO HEX"
710 :  CHAR,12,4,"**************"
720 :  CHAR,4,7,""
730 :  INPUT"WHAT IS THE DECIMAL NUMBER";D$
740 :  H$="":REM NO SPACE
750 :  BAD=0
760 :  FOR N=1 TO LEN(D$)
770 :     DD=ASC(MID$(D$,N,1))
780 :     IF DD<48 OR DD>57 THEN BAD=1:N=LEN(D$)
```

```
790 :  NEXT
800 LOOP UNTIL BAD=0
810 DC=VAL(D$)
820 REM**********************
830 REM CALCULATE LOOP (HEX)
840 DO
850 :  D=16*(DC/16-INT(DC/16))
860 :  IF DC<16 THEN D=DC
870 :  DC=INT(DC/16)
880 :  D=D+48:IF D>57 THEN D=D+7:REM CONVERT D TO ASCII
890 :  H$=CHR$(D)+H$
900 LOOP UNTIL DC=0
910 REM HEX$ FUNCTION NOT USED SO THAT VALUES
920 REM GREATER THAN 65535 CAN BE ACCEPTED.
930 REM**********************
940 CHAR,4,10,"HEXADECIMAL NUMBER IS "
950 PRINT H$
960 GOSUB 1530:REM ANYKEY
970 RETURN
980 REM**********************
990 REM**********************
1000 REM BINARY TO DECIMAL
1010 DO
1020 :  SCNCLR
1030 :  CHAR,10,3,"BINARY TO DECIMAL"
1040 :  CHAR,10,4,"*****************"
1050 :  CHAR,4,7,"WHAT IS THE BINARY NUMBER"
1060 :  CHAR,4,8,"":INPUT B$
1070 :  DC=0:BAD=0
1080 :  FOR N=1 TO LEN(B$)
1090 :     BIN=ASC(MID$(B$,N,1))-48
1100 :     IF BIN<0 OR BIN>1 THEN BAD=1:N=LEN(B$)
1110 :     D=BIN*2^(LEN(B$)-N)
1120 :     DC=DC+D
1130 :  NEXT
1140 LOOP UNTIL BAD=0
1150 CHAR,4,11,"DECIMAL NUMBER IS"
1160 PRINT DC
1170 GOSUB 1530:REM ANYKEY
1180 RETURN
1190 REM**********************
1200 REM**********************
1210 REM DECIMAL TO BINARY
1220 DO
1230 :  SCNCLR
1240 :  CHAR,10,3,"DECIMAL TO BINARY"
1250 :  CHAR,10,4,"*****************"
```

```
1260 :   CHAR,4,7,""
1270 :   INPUT"WHAT IS THE DECIMAL NUMBER";D$
1280 :   B$="":REM NO SPACE
1290 :   BAD=0
1300 :   FOR N=1 TO LEN(D$)
1310 :      DD=ASC(MID$(D$,N,1))
1320 :      IF DD<48 OR DD>57 THEN BAD=1:N=LEN(D$)
1330 :   NEXT
1340 LOOP UNTIL BAD=0
1350 DC=VAL(D$)
1360 REM**********************
1370 REM CALCULATE LOOP (BIN)
1380 DO
1390 :   D=2*(DC/2-INT(DC/2))
1400 :   IF DC<2 THEN D=DC
1410 :   DC=INT(DC/2)
1420 :   D=D+48:REM CONVERT D TO ASCII
1430 :   B$=CHR$(D)+B$
1440 LOOP UNTIL DC=0
1450 REM**********************
1460 CHAR,4,10,"BINARY NUMBER IS"
1470 CHAR,4,11,"":PRINT B$
1480 GOSUB 1530:REM ANYKEY
1490 RETURN
1500 REM**********************
1510 REM**********************
1520 REM ANYKEY
1530 PRINT:PRINT
1540 PRINT TAB(4)"PRESS ANY KEY TO GET MENU"
1550 GETKEY A$
1560 RETURN
1570 REM**********************
1580 REM**********************
```

P85 Color codes for resistors

This program could be useful in an electronics laboratory. It allows you to calculate the value of a resistor from its color code, or to calculate the color code from its value. The program deals only with resistors coded by color bands as shown below:



COMMANDS

Key in the program and RUN.
Follow the instructions.

```
10 REM COLOR CODES FOR RESISTORS
20 REM ************************
30 REM
40 SCNCLR
50 COLOR 0,12,7:REM PINK SCREEN
60 COLOR 1,1:REM BLACK INK
70 COLOR 4,12,7:REM PINK BORDER
80 PRINT TAB(10)"RESISTOR COLOR CODES"
90 PRINT TAB(10)"********************"
100 PRINT:PRINT:PRINT
110 PRINT TAB(4)"THIS PROGRAM MAY BE USED TO"
120 PRINT TAB(4)"DETERMINE THE VALUE OF A RESISTOR"
130 PRINT TAB(4)"FROM ITS COLOR CODE, OR TO"
140 PRINT TAB(4)"DETERMINE ITS COLOR CODE FROM ITS"
```

```
150 PRINT TAB(4)"RESISTANCE VALUE."
160 DIM B$(12):REM HOLDS COLORS
170 FOR N=0 TO 12
180 :  READ A$:B$(N)=A$
190 NEXT
200 REM*********************
210 DATA BLACK,BROWN,RED,ORANGE,YELLOW
220 DATA GREEN,BLUE,VIOLET,GRAY,WHITE
230 DATA GOLD,SILVER,NONE
240 REM*********************
250 GOSUB 1660:REM ANY KEY
260 REM
270 REM
280 REM
290 REM
300 REM*********************
310 REM*********************
320 DO
330 :  SCNCLR:PRINT:PRINT:PRINT
340 :  PRINT TAB(4)"DO YOU WISH TO FIND RESISTANCE"
350 :  PRINT TAB(4) "VALUE OR COLOR CODE?"
360 :  PRINT:PRINT
370 :  PRINT TAB(8)"PRESS KEY 1 FOR VALUE"
380 :  PRINT
390 :  PRINT TAB(8)"PRESS KEY 2 FOR CODE"
400 :  PRINT
410 :  PRINT TAB(8)"PRESS KEY 3 TO END"
420 :  DO:GETKEY A$:LOOP UNTIL A$="1" OR A$="2" OR A$="3"
430 :  A=VAL(A$)
440 :  ON A GOSUB 560,1120
450 LOOP UNTIL A=3
460 END
470 REM
480 REM
490 REM        ***************
500 REM        *             *
510 REM        * SUBROUTINES *
520 REM        *             *
530 REM        ***************
540 REM
550 REM VALUE
560 DO
570 :  SCNCLR
580 :  PRINT TAB(12)"COLORS ARE:"
590 :  PRINT
600 :  FOR N=0 TO 9
610 :     PRINT,B$(N),"=";N
```

```
620 :  NEXT
630 :  PRINT:PRINT
640 :  PRINT TAB(3)"ENTER THE THREE COLOR BANDS AS"
650 :  PRINT TAB(3)"THREE NUMBERS FOLLOWED BY THE"
660 :  PRINT TAB(3)"RETURN KEY."
670 :  PRINT:PRINT
680 :  PRINT TAB(3)"FOR EXAMPLE:"
690 :  PRINT TAB(3)"RED-BLACK-ORANGE=203"
700 :  PRINT
710 :  PRINT TAB(3);:INPUT"WHAT ARE THE COLORS";COL$
720 :  REM CHECK ENTRY
730 :  FOR N=1 TO LEN(COL$)
740 :     D(N)=ASC(MID$(COL$,N,1))-48
750 :     IF D(N)<0 OR D(N)>9 THEN F=1:N=3:ELSE F=0
760 :     IF LEN(COL$)<>3 THEN F=1
770 :  NEXT
780 LOOP UNTIL F=0
790 REM**********************
800 REM GET TOLERANCE
810 SCNCLR
820 PRINT:PRINT
830 PRINT TAB(3)"THE THIRD BAND INDICATES"
840 PRINT TAB(3)"TOLERANCE. THE ALTERNATIVES ARE:"
850 PRINT:PRINT
860 FOR N=10 TO 12
870 :  PRINT,N-9,B$(N)
880 NEXT
890 PRINT:PRINT
900 PRINT TAB(3)"ENTER 1,2 OR 3 TO SELECT"
910 PRINT TAB(3)"TOLERANCE COLOR. ANY OTHER ENTRY"
920 PRINT TAB(3)"WILL BE TREATED AS NO FOURTH"
930 PRINT TAB(3)"BAND."
940 PRINT:PRINT
950 PRINT TAB(3);:INPUT"SELECTION";T$
960 IF T$<>"1" AND T$<>"2" THEN T$="3"
970 T=VAL(T$)
980 T=T*5
990 REM**********************
1000 REM CALCULATE RESISTANCE
1010 RES=(D(1)*10+D(2))*10^D(3)
1020 SCNCLR
1030 PRINT:PRINT:PRINT
1040 PRINT TAB(4)"RESISTANCE IS";RES;"OHM"
1050 PRINT:PRINT:PRINT
1060 PRINT TAB(4)"TOLERANCE IS";T;"%"
1070 GOSUB 1660:REM ANY KEY
1080 RETURN
```

```
1090 REM**********************
1100 REM**********************
1110 REM GET CODE
1120 DO
1130 :  SCNCLR
1140 :  PRINT:PRINT:PRINT
1150 :  PRINT TAB(4)"PLEASE ENTER RESISTANCE IN OHM."
1160 :  PRINT TAB(4)"PLEASE ENTER NUMBERS ONLY."
1170 :  PRINT TAB(4)"ANY OTHER SYMBOLS, INCLUDING"
1180 :  PRINT TAB(4)"DECIMAL POINTS WILL NOT BE"
1190 :  PRINT TAB(4)"ACCEPTED. PLEASE DO NOT START"
1200 :  PRINT TAB(4)"THE ENTRY WITH A ZERO."
1210 :  PRINT
1220 :  PRINT TAB(4)"RESISTORS WITH VALUES LESS THAN"
1230 :  PRINT TAB(4)"10 OHM OR GREATER THAN"
1240 :  PRINT TAB(4)"99000000000 OHM ARE NOT COVERED"
1250 :  PRINT TAB(4)"BY THIS PROGRAM."
1260 :  PRINT:PRINT
1270 :  PRINT TAB(4);:INPUT"RESISTANCE VALUE IN OHM";RST$
1280 :  FOR N=1 TO LEN(RST$)
1290 :     T=ASC(MID$(RST$,N,1))
1300 :     IF T>57 OR T<48-(N=1) THEN F=1:ELSE F=0
1310 :     IF F=1 THEN N=LEN(RST$)
1320 :  NEXT
1330 :  IF LEN(RST$)>11 OR LEN(RST$)<2 THEN F=1
1340 LOOP UNTIL F=0
1350 REM**********************
1360 REM GET TOLERANCE
1370 SCNCLR
1380 PRINT:PRINT:PRINT
1390 PRINT TAB(4)"ENTER 1,2 OR 3 DEPENDING ON"
1400 PRINT TAB(4)"WHETHER THE TOLERANCE IS 5%,10%"
1410 PRINT TAB(4)"OR 20% RESPECTIVELY."
1420 PRINT
1430 PRINT TAB(4)"ALL ENTRIES OTHER THAN 1 OR 2"
1440 PRINT TAB(4)"WILL BE ASSUMED TO MEAN 20%"
1450 PRINT TAB(4)"TOLERANCE."
1460 PRINT:PRINT
1470 PRINT TAB(4);:INPUT"TOLERANCE (1,2 OR 3)";TL$
1480 IF TL$<>"1" AND TL$<>"2" THEN TL$="3"
1490 REM**********************
1500 REM DISPLAY BANDS
1510 SCNCLR
1520 PRINT:PRINT:PRINT
1530 A(1)=VAL(LEFT$(RST$,1))
1540 A(2)=VAL(MID$(RST$,2,1))
1550 A(3)=LEN(RST$)-2
```

```
1560 A(4)=ASC(TL$)-39
1570 FOR N=1 TO 4
1580 :  PRINT TAB(4)"BAND";N;"IS",B$(A(N))
1590 :  PRINT
1600 NEXT
1610 GOSUB 1660:REM ANY KEY
1620 RETURN
1630 REM**********************
1640 REM**********************
1650 REM ANY KEY
1660 PRINT:PRINT:PRINT
1670 PRINT TAB(4)"PRESS ANY KEY FOR INSTRUCTIONS"
1680 GETKEY A$
1690 RETURN
1700 REM**********************
1710 REM**********************
```

P86 Volumes of solids

This program calculates the volumes of spheres, cylinders and cones or pyramids.

It could be extended to calculate the volumes of other regular solids.

COMMANDS

Key in the program and RUN.
Follow the instructions.

```
10 REM VOLUME OF SOLIDS
20 REM ****************
30 REM
40 SCNCLR
50 COLOR 0,15,1:REM DARK BLUE SCREEN
60 COLOR 1,4:REM CYAN PRINT
70 COLOR 4,15,1:REM DARK BLUE BORDER
80 PRINT TAB(10)"VOLUMES OF SOLIDS"
90 PRINT TAB(10)"******************"
100 PRINT:PRINT:PRINT
110 PRINT TAB(5)"PRESS KEY 1 FOR SPHERE"
120 PRINT
130 PRINT TAB(5)"PRESS KEY 2 FOR CYLINDER"
140 PRINT
150 PRINT TAB(5)"PRESS KEY 3 FOR CONE OR PYRAMID"
160 PRINT
170 PRINT TAB(5)"PRESS KEY 4 TO FINISH"
180 GETKEY A
190 IF A=4 THEN END
200 SCNCLR
210 ON A GOSUB 310,440,610
220 RUN
230 REM
240 REM       ***************
250 REM       *             *
260 REM       * SUBROUTINES *
270 REM       *             *
280 REM       ***************
290 REM
```

```
300 REM SPHERE
310 PRINT TAB(10)"VOLUME OF A SPHERE"
320 PRINT TAB(10)"******************"
330 PRINT:PRINT:PRINT
340 INPUT"WHAT IS THE RADIUS";R
350 PRINT:PRINT:PRINT
360 VOL=4*[PI ]*R*R*R/3
370 PRINT"THE VOLUME OF A SPHERE OF RADIUS";R
380 PRINT:PRINT"IS";VOL
390 GOSUB 990
400 RETURN
410 REM*************************
420 REM*************************
430 REM CYLINDER
440 PRINT TAB(9)"VOLUME OF A CYLINDER"
450 PRINT TAB(9)"********************"
460 PRINT:PRINT:PRINT
470 INPUT"WHAT IS THE HEIGHT";H
480 PRINT:PRINT
490 INPUT"WHAT IS THE BASE RADIUS";R
500 PRINT:PRINT:PRINT
510 VOL=[PI ]*R*R*H
520 PRINT"THE VOLUME OF A CYLINDER OF"
530 PRINT
540 PRINT"BASE RADIUS";R;"AND HEIGHT";H
550 PRINT:PRINT"IS";VOL
560 GOSUB 990
570 RETURN
580 REM*************************
590 REM*************************
600 REM CONE
610 PRINT TAB(11)"VOLUME OF A CONE"
620 PRINT TAB(11)"****************"
630 PRINT:PRINT:PRINT
640 INPUT"DO YOU KNOW THE BASE AREA (Y/N)";Y$
650 IF ASC(Y$)=89 THEN GOSUB 1060:GOSUB 830:ELSE GOSUB 690
660 RETURN
670 REM*************************
680 REM*************************
690 PRINT:PRINT
700 PRINT"SELECT BASE SHAPE BY PRESSING:"
710 PRINT
720 PRINT"KEY 1 FOR EQUILATERAL TRIANGLE"
730 PRINT TAB(4)"2 FOR SQUARE"
740 PRINT TAB(4)"3 FOR CIRCLE"
750 PRINT
760 DO:GETKEY B:LOOP UNTIL B<4 AND B>0
```

```
770 ON B GOSUB 1120,1180,1240
780 GOSUB 830
790 RETURN
800 REM************************
810 REM************************
820 REM CONE VOLUME
830 SCNCLR
840 PRINT TAB(11)"VOLUME OF A CONE"
850 PRINT TAB(11)"***************"
860 PRINT:PRINT:PRINT
870 INPUT"WHAT IS THE HEIGHT";H
880 PRINT:PRINT:PRINT
890 PRINT"THE VOLUME OF A CONE OF BASE"
900 PRINT
910 PRINT"AREA";AR;"AND HEIGHT";H
920 PRINT
930 PRINT"IS";AR*H/3
940 GOSUB 990
950 RETURN
960 REM************************
970 REM************************
980 REM ANY KEY
990 PRINT:PRINT:PRINT
1000 PRINT"PRESS ANY KEY TO RETURN"
1010 GETKEY A$
1020 RETURN
1030 REM************************
1040 REM************************
1050 REM GET BASE AREA
1060 PRINT:PRINT
1070 INPUT"WHAT IS THE BASE AREA";AR
1080 RETURN
1090 REM************************
1100 REM************************
1110 REM TRIANGLE
1120 INPUT"LENGTH OF SIDE OF TRIANGLE";S
1130 AR=.5*S*S*SIN([PI ]/3)
1140 RETURN
1150 REM************************
1160 REM************************
1170 REM SQUARE
1180 INPUT"LENGTH OF SIDE OF SQUARE";S
1190 AR=S*S
1200 RETURN
1210 REM************************
1220 REM************************
```

```
1230 REM CIRCLE
1240 INPUT"RADIUS OF CIRCLE";R
1250 AR=[PI ]*R*R
1260 RETURN
1270 REM************************
1280 REM************************
```

P87 Physics experiment 1 – moment of inertia

The next two programs are an attempt to show how a micro could be used within a physics laboratory to take some of the drudgery out of experimentation.

The programs are based on two experiments in F. Tyler's "A Laboratory Manual of Physics", Edward Arnold, 1966.

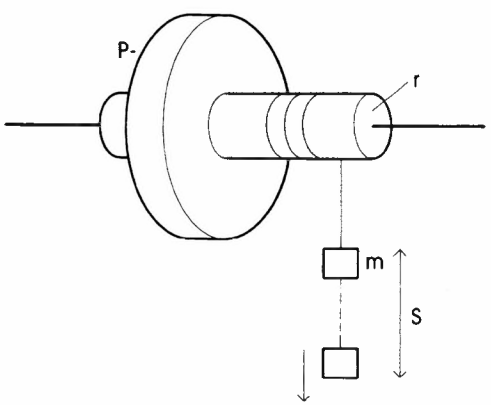
COMMANDS

Use WORK SHEETS and programs to perform the experiments.

EXPERIMENT 1 - MOMENT OF INERTIA

Work Sheet 1

Determination of the moment of inertia of a flywheel



Apparatus

Wall supported flywheel of standard pattern: a weight is attached to a length of fine cord which is wrapped round the axle, the free end being passed through a hole in the axle. The length of the cord is adjusted so that the cord detaches itself from the axle when the weight reaches the ground. Callipers, stop watch, meter rule.

Method

The value of m is obtained by weighing; the radius r of the axle is found by using callipers.

The weight (m) is allowed to fall through a measured distance (s) to the ground, and the time of descent (t) is taken by a stop-watch. The number of revolutions (n) of the wheel during this time is taken by observing a mark made on the circumference of the wheel at P. The further revolutions (p) made by the wheel before coming to rest after m is detached are also counted by reference to the mark P. The experiment is repeated three times for the same distance (s).

Perform the experiment as follows:

Power on microcomputer.
Load program.
Take measurements m, r and s.
RUN the program.
Perform the experiment as directed, and enter values as prompted.

```
10 REM MOMENT OF INERTIA
20 REM *****************
30 REM
40 COLOR 0,8,5:REM YELLOW SCREEN
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,3,2:REM RED BORDER
70 SCNCLR
80 PRINT:PRINT
90 PRINT TAB(10)CHR$(18)"--------------------"
100 PRINT TAB(10)CHR$(18)"  MOMENT OF INERTIA "
110 PRINT TAB(10)CHR$(18)"        OF A        "
120 PRINT TAB(10)CHR$(18)"      FLYWHEEL      "
```

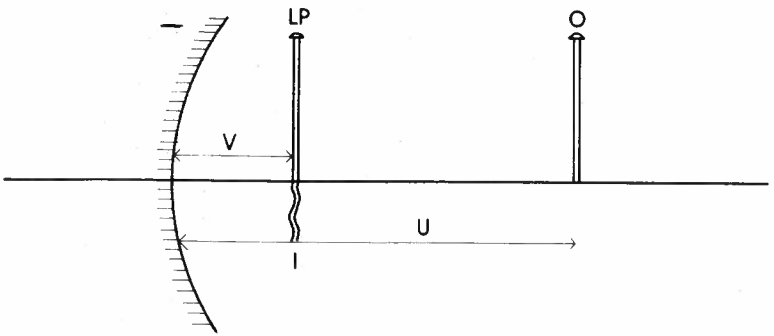
```
130 PRINT TAB(10)CHR$(18)"--------------------"
140 PRINT
150 PRINT TAB(10)"PHYSICS EXPERIMENT 1"
160 PRINT TAB(10)"--------------------"
170 PRINT
180 PRINT TAB(10)CHR$(18)"--------------------"
190 PRINT TAB(10)CHR$(18)"   SEE WORK SHEET   "
200 PRINT TAB(10)CHR$(18)"--------------------"
210 PRINT:PRINT:PRINT
220 PRINT TAB(6)CHR$(18)" PRESS ANY KEY TO CONTINUE "
230 GETKEY A$
240 REM************************
250 DO
260 :  R=0:M=0:S=0
270 :  SCNCLR:PRINT:PRINT
280 :  DO
290 :     PRINT TAB(4);
300 :     INPUT"RADIUS OF AXLE (METERS)";R
310 :  LOOP UNTIL R>0
320 :  PRINT
330 :  DO
340 :     PRINT TAB(4);
350 :     INPUT"MASS OF WEIGHT (KG)";M
360 :  LOOP UNTIL M>0
370 :  PRINT
380 :  DO
390 :     PRINT TAB(4);
400 :     INPUT"DISTANCE TO GROUND (METERS)";S
410 :  LOOP UNTIL S>0
420 :  REM**********************
430 :  TT=0:NN=0:PP=0
440 :  FOR K=1 TO 3
450 :     T=0:N=0:P=0
460 :     SCNCLR:PRINT:PRINT
470 :     PRINT TAB(10)CHR$(18)" PERFORM EXPERIMENT "
480 :     PRINT:PRINT
490 :     PRINT TAB(17)"RUN";K
500 :     PRINT TAB(17)"====="
510 :     PRINT
520 :     DO
530 :        PRINT TAB(10);
540 :        INPUT"TIME (SECS)";T
550 :     LOOP UNTIL T>0
560 :     PRINT
570 :     DO
580 :        PRINT TAB(10);
590 :        INPUT"N (REVS)";N
```

```
600 :      LOOP UNTIL N>0
610 :      PRINT
620 :      DO
630 :         PRINT TAB(10);
640 :         INPUT"P (REVS)";P
650 :      LOOP UNTIL P>0
660 :      TT=TT+T:NN=NN+N:PP=PP+P
670 :   NEXT
680 :   REM**********************
690 :   T=TT/3:N=NN/3:P=PP/3
700 :   G=9.81:REM ACCELERATION DUE TO GRAVITY
710 :   IT=M*R*R*(G*T*T/2/S-1)*(P/(P+N))
720 :   IT=INT(IT*100+.5)/100
730 :   IT$=MID$(STR$(IT),2):IF IT<1 THEN IT$="0"+IT$
740 :   SCNCLR:PRINT:PRINT:PRINT:PRINT
750 :   PRINT TAB(10)"MOMENT OF INERTIA ="
760 :   PRINT
770 :   PRINT TAB(10)IT$" KG-(METER)^2"
780 :   PRINT TAB(10)"-------------------"
790 :   REM**********************
800 :   PRINT:PRINT
810 :   Y$="N"
820 :   PRINT TAB(7);
830 :   INPUT"ANOTHER EXPERIMENT (Y/N)";Y$
840 LOOP UNTIL ASC(Y$)<>89
850 END
860 REM**********************
870 REM**********************
```

P88 Physics experiment 2 – focal length

Work Sheet 2

Determination of the focal length of a concave mirror



Apparatus

Concave mirror in stand, two retort stands with clamps and pins, meter rule.

Method

The object pin O is placed a given distance (u) from the concave mirror. The position of the image I formed by reflection in the mirror is located by the method of non-parallax using the second pin (locating pin LP). The distance (v) of the locating pin from the mirror is measured. O and I are said to be conjugate points. A series of values of v for a given range of values of u are obtained.

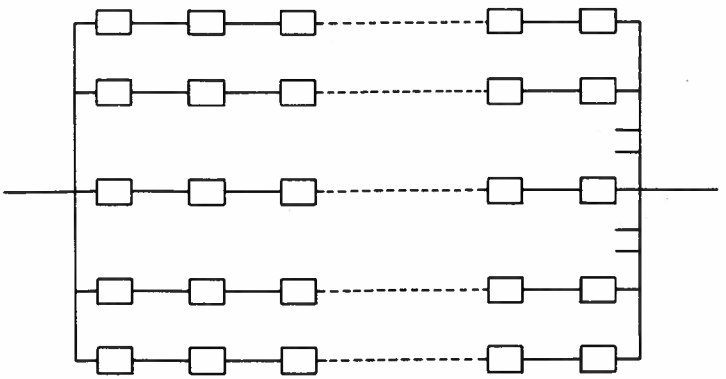
The computer program is used to calculate the focal length of the mirror from each measurement. The average of these values, and their standard deviation, are calculated and presented on the screen.

```
10 REM FOCAL LENGTH
20 REM ************
30 REM
40 COLOR 0,8,5:REM YELLOW SCREEN
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,3,2:REM RED BORDER
70 SCNCLR:PRINT
80 PRINT TAB(12)CHR$(18)"----------------"
90 PRINT TAB(12)CHR$(18)"  FOCAL LENGTH  "
100 PRINT TAB(12)CHR$(18)"      OF A      "
110 PRINT TAB(12)CHR$(18)" CONCAVE MIRROR "
120 PRINT TAB(12)CHR$(18)"----------------"
130 PRINT
140 PRINT TAB(10)"PHYSICS EXPERIMENT 2"
150 PRINT TAB(10)"--------------------"
160 PRINT
170 PRINT TAB(12)CHR$(18)"----------------"
180 PRINT TAB(12)CHR$(18)" SEE WORK SHEET "
190 PRINT TAB(12)CHR$(18)"----------------"
200 PRINT:PRINT
210 REM**********************
220 DO
230 :  PRINT TAB(9);
240 :  INPUT"HOW MANY MEASUREMENTS";MEAS%
250 LOOP UNTIL MEAS%>0
260 REM**********************
270 FOR N=1 TO MEAS%
280 :  SCNCLR:PRINT:PRINT
290 :  PRINT TAB(13)"MEASUREMENT";N
300 :  PRINT:PRINT
310 :  DO
320 :     PRINT TAB(16);:INPUT"U = ";U
330 :  LOOP UNTIL U>0
340 :  PRINT
350 :  DO
360 :     PRINT TAB(16):INPUT"V = ";V
370 :  LOOP UNTIL V>0
380 :  T=1/(1/V+1/U)
390 :  SM=SM+T
400 :  SQ=SQ+T*T
410 NEXT
420 REM**********************
430 SCNCLR:PRINT:PRINT
440 MN=SM/MEAS%
450 DV=SQR(SQ/MEAS%-MN*MN)
460 PRINT TAB(4)"AVERAGE FOCAL LENGTH =";MN
470 PRINT
```

```
480 PRINT TAB(4)"STANDARD DEVIATION   =";DV
490 PRINT:PRINT
500 PRINT TAB(4):INPUT"WANT TO REPEAT THE EXPERIMENT";Y$
510 IF ASC(Y$)=89 THEN RUN
520 REM**********************
530 REM**********************
```

P89 Resistors

This program computes the resultant resistance of an electric circuit of the following type:



COMMANDS

Key in the program and RUN.
Follow the instructions, entering the resistance values as required.

```
10 REM RESISTORS
20 REM *********
30 REM
40 SCNCLR
50 COLOR 0,1:REM BLACK SCREEN
60 COLOR 1,8:REM YELLOW INK
70 COLOR 4,1:REM BLACK BORDER
80 PRINT
90 PRINT TAB(15)"RESISTORS"
100 PRINT:PRINT:PRINT
110 PRINT TAB(5)"THIS PROGRAM COMPUTES THE TOTAL"
120 PRINT TAB(5)"RESISTANCE OF A CIRCUIT"
130 PRINT TAB(5)"CONSISTING OF SEVERAL BRANCHES"
```

```
140 PRINT TAB(5)"CONNECTED IN PARALLEL, EACH"
150 PRINT TAB(5)"BRANCH CONSISTING OF ONE OR"
160 PRINT TAB(5)"MORE SERIES RESISTORS."
170 PRINT:PRINT:PRINT
180 PRINT TAB(5)"PRESS ANY KEY TO CONTINUE"
190 GETKEY A$
200 REM********************
210 REM********************
220 REM ANOTHER
230 CLR
240 SCNCLR
250 DO:INPUT"NUMBER OF BRANCHES";BR%:LOOP UNTIL BR%>0
260 DIM R(BR%)
270 REM********************
280 REM********************
290 FOR N=1 TO BR%
300 GOSUB 670:REM PRINT BRANCH NUMBER
310 DO:INPUT"NUMBER OF RESISTORS";RS%:LOOP UNTIL RS%>0
320 REM********************
330 REM********************
340 GOSUB 670:REM PRINT BRANCH NUMBER
350 FOR K=1 TO RS%
360 DO:PRINT"VALUE OF RESISTOR";K;:INPUT VL:LOOP WHILE VL<0
370 REM********************
380 GOSUB 670:REM PRINT BRANCH NUMBER
390 R(N)=R(N)+VL
400 NEXT
410 IF R(N)=0 THEN ZER=1:N=BR%:ELSE CN=CN+1/R(N)
420 REM ADD CONDUCTANCES UNLESS ZERO RESISTANCE BRANCH
430 NEXT
440 REM********************
450 REM********************
460 IF ZER=1 THEN TTL=0:ELSE TTL=1/CN:REM TOTAL RESISTANCE
470 REM DISPLAY TOTAL RESISTANCE
480 SCNCLR
490 CHAR,9,9,"TOTAL RESISTANCE IS"
500 PRINT
510 PRINT TAB(9) TTL;"OHM"
520 PRINT TAB(9)"=================="
530 CHAR,9,18,""
540 INPUT"ANOTHER CIRCUIT";Y$
550 IF ASC(Y$)=89 THEN 230:REM ANOTHER
560 REM CANNOT USE DO LOOP CONTAINING CLR INSTRUCTION
570 END
```

```
580 REM         *****************
590 REM         *               *
600 REM         *               *
610 REM         *  SUBROUTINES  *
620 REM         *               *
630 REM         *               *
640 REM         *****************
650 REM
660 REM PRINT BRANCH NUMBER
670 SCNCLR
680 PRINT:PRINT:PRINT
690 PRINT TAB(15)"BRANCH";N
700 PRINT:PRINT
710 RETURN
720 REM*********************
730 REM*********************
```

P90 Calculator

There are many occasions when you need the capability of a simple calculator rather than a complex computer. This program simulates a simple four function (+,-,*,/) calculator.

The program could be expanded to provide memory-add, memory-subtract and memory-read features.

COMMANDS

Key in the program and RUN.
Use the numeric keys and +, -, * and / to perform arithmetic. Use '.' for decimal point and '=' to get final answer. Use the C, A, and S keys as instructed.

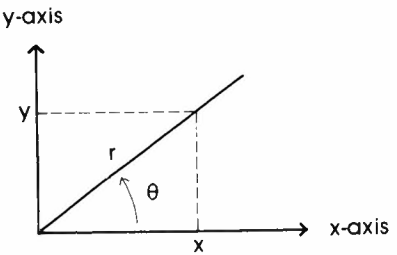
```
10 REM CALCULATOR
20 REM **********
30 REM
40 COLOR 0,8,5:REM YELLOW SCREEN
50 COLOR 1,7,1:REM BLUE INK
60 COLOR 4,9,3:REM ORANGE BORDER
70 UP$=CHR$(145):REM UP CURSOR
80 REM ***********************
90 SCNCLR
100 PRINT:PRINT:PRINT
110 PRINT TAB(14)CHR$(18)"CALCULATOR"
120 PRINT:PRINT:PRINT
130 PRINT TAB(7)"KEY C CLEARS CURRENT ENTRY"
140 PRINT
150 PRINT TAB(7)"KEY A CLEARS ALL ENTRIES"
160 PRINT
170 PRINT TAB(7)"KEY S STOPS PROGRAM"
180 PRINT:PRINT:PRINT:PRINT
190 REM***********************
200 S$=CHR$(32):REM SPACE
210 FOR N=0 TO 4
220 :  S$=S$+S$
230 NEXT:
240 REM S$ CONTAINS 32 SPACES
250 E$="":REM NO SPACE
```

```
260 REM**********************
270 REM CLEAR ALL
280 A$=E$:F$=E$:R=0
290 REM**********************
300 REM REPEAT
310 DO
320 :  GETKEY B$
330 :  REM*********************
340 :  REM DECODE
350 :  IF ASC(B$)>47 AND ASC(B$)<58THEN GOSUB 550
360 :  IF B$="." THEN GOSUB 550
370 :  IF B$="+" OR B$="-" OR B$="*" THEN GOSUB 640
380 :  IF B$="/" OR B$="=" THEN GOSUB 640
390 :  IF B$="A" OR B$="C" THEN GOSUB 790
400 :  IF B$="A" THEN A$=E$:F$=E$:R=0
410 LOOP UNTIL B$="S"
420 END
430 REM
440 REM**********************
450 REM**********************
460 REM        ***************
470 REM        *             *
480 REM        * SUBROUTINES *
490 REM        *             *
500 REM        ***************
510 REM
520 REM**********************
530 REM**********************
540 REM NUMBER OR DECIMAL POINT
550 IF F$="=" THEN R=0:A$=E$:F$=E$
560 A$=A$+B$
570 IF LEN(A$)>8 THEN A$=LEFT$(A$,8)
580 PRINT UP$+S$:PRINTUP$+UP$
590 PRINT TAB(21-LEN(A$)) A$
600 RETURN
610 REM**********************
620 REM**********************
630 REM OPERATOR
640 BAD=0:REM USED TO TEST FOR DIVISION BY ZERO
650 IF F$="+" OR F$=E$ THEN R=R+VAL(A$)
660 IF F$="-" THEN R=R-VAL(A$)
670 IF F$="*" THEN R=R*VAL(A$)
680 IF F$="/" AND VAL(A$)<>0 THEN R=R/VAL(A$)
690 PRINT UP$+S$:PRINT UP$+UP$
700 IF F$="/" AND VAL(A$)=0 THEN BAD=1
710 IF BAD=0 THEN F$=B$:P$=F$
720 IF BAD=0 AND F$="=" THEN P$=CHR$(146)+CHR$(32)
```

```
730 PRINT TAB(21-LEN(STR$(R)))R;TAB(33) CHR$(18)+P$
740 A$=E$
750 RETURN
760 REM**********************
770 REM**********************
780 REM CLEAR
790 PRINT UP$+S$
800 IF B$="A" THEN PRINT TAB(33)UP$+CHR$(32)
810 A$=E$
820 RETURN
830 REM**********************
840 REM**********************
```

P91 Coordinate conversion

It can happen quite often that you have points plotted on a graph in the rectangular (x,y) format and wish to convert the co-ordinates of these points to the polar (r,θ) format, or vice versa, as in the following figure:



The conversion formulae are:

```
x=r*COS(θ)
y=r*SIN(θ)
r=x^2+y^2
θ=ARC TAN(y/x)
```

This is a demonstration program and the formulae used are accurate only for the first quadrant. The special case when x=0 has not been dealt with, nor has any 'trap' been set for entry of negative values of r. The program, extended to incorporate all these features, could be a valuable tool for mathematicians.

COMMANDS

Key in the program and RUN.
Follow the instructions.

```
10 REM COORDINATE CONVERSION
20 REM **********************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,7,1:REM BLUE INK
60 COLOR 4,9,1:REM BROWN BORDER
70 SCNCLR
80 CHAR,12,5,CHR$(18)+"------------"
90 CHAR,12,6," COORDINATE "
100 CHAR,12,7,"             "
110 CHAR,12,8," CONVERSION "
120 CHAR,12,9,"------------"
130 CHAR,6,18," PRESS ANY KEY TO CONTINUE "
140 GETKEY A$
150 REM************************
160 PRINT CHR$(146):REM REVERSE OFF
170 GOSUB 750:REM DEGREES OR RADIANS
180 REM************************
190 REM MENU
200 DO
210 :  DO
220 :     SCNCLR
230 :     CHAR,4,6,"KEY 1. RECTANGULAR TO POLAR"
240 :     CHAR,4,8,"KEY 2. POLAR TO RECTANGULAR"
250 :     CHAR,4,10,"KEY 3. STOPS THE PROGRAM"
260 :     CHAR,4,13,""
270 :     INPUT"ENTER YOUR CHOICE (1,2 OR 3)";C$
280 :     C%=ASC(C$)-48
290 :  LOOP UNTIL C%>0 AND C%<4
300 :  ON C% GOSUB 410,580
310 LOOP UNTIL C%=3
320 END
330 REM
340 REM       ***************
350 REM       *             *
360 REM       * SUBROUTINES *
370 REM       *             *
380 REM       ***************
390 REM
400 REM RECTANGULAR TO POLAR
410 SCNCLR
420 CHAR,8,6,"":INPUT"WHAT IS X-COORDINATE";X
430 CHAR,8,8,"":INPUT"WHAT IS Y-COORDINATE";Y
440 R=SQR(X*X+Y*Y)
450 A=ATN(Y/X)
```

```
460 CHAR,0,13,""
470 PRINT TAB(8)"RADIAL VALUE =";R
480 PRINT
490 PRINT TAB(8)"ANGLE =";FNA1(A);T$
500 PRINT:PRINT
510 PRINT TAB(12)CHR$(18)"PRESS ANY KEY";
520 GETKEY A$
530 GOSUB 900
540 RETURN
550 REM************************
560 REM************************
570 REM POLAR TO RECTANGULAR
580 SCNCLR
590 CHAR,8,6,"":INPUT"WHAT IS RADIAL VALUE";R
600 CHAR,8,8,"":INPUT"WHAT IS ANGLE";A
610 A=FNA2(A)
620 X=R*COS(A):Y=R*SIN(A)
630 CHAR,0,13,""
640 PRINT TAB(8)"X COORDINATE =";X
650 PRINT
660 PRINT TAB(8)"Y COORDINATE =";Y
670 PRINT:PRINT
680 PRINT TAB(12)CHR$(18)"PRESS ANY KEY";
690 GETKEY A$
700 GOSUB 900
710 RETURN
720 REM************************
730 REM************************
740 REM SELECT DEGREES OR RADIANS
750 DO
760 :  SCNCLR
770 :  CHAR,4,6,"DO YOU WISH ANGLES TO BE IN:"
780 :  CHAR,12,8,"1. RADIANS?"
790 :  CHAR,12,10,"2. DEGREES?"
800 :  CHAR,4,13,""
810 :  INPUT"WHAT IS YOUR CHOICE (1 OR 2)";B$
820 :   B=ASC(B$)-48
830 LOOP UNTIL B=1 OR B=2
840 IF B=1 THEN DEF FNA1(A)=A:DEF FNA2(A)=A:T$="RADIANS"
850  IF  B=2  THEN  DEF  FNA1(A)=A*180/[PI   ]:DEF   FNA2(A)=
A*[PI ]/180:T$="DEGREES"
860 RETURN
870 REM************************
880 REM************************
890 REM ALTER ANGLE UNITS
900 SCNCLR
910 PRINT CHR$(146)
```

```
920 CHAR,4,10,""
930 INPUT"WANT TO ALTER ANGLE UNITS";Y$
940 IF ASC(Y$)=89 THEN GOSUB 750
950 RETURN
960 REM************************
970 REM************************
```

P92 Vector multiplication

This is a rather simple program which can be used to find the dot and cross products of vectors. The vectors used have only three components.

The program would have been more complex if it had been written for general vectors. We believe, however, that this is a useful routine to have in any program library.

COMMANDS

Key in the program and RUN.
Enter vectors when prompted.

```
10 REM VECTOR MULTIPLICATION
20 REM **********************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,15,1:REM BLUE BORDER
70 SCNCLR
80 CHAR,9,2,"VECTOR MULTIPLICATION"
90 CHAR,9,3,"**********************"
100 CHAR,4,6,"THIS PROGRAM COMPUTES BOTH THE"
110 CHAR,4,7,"DOT AND THE CROSS PRODUCTS OF"
120 CHAR,4,8,"TWO VECTORS."
130 CHAR,4,10,"THE VECTORS ARE ENTERED IN"
140 CHAR,4,11,"COMPONENT FORM AND THEIR PRODUCTS"
150 CHAR,4,12,"ARE THEN GIVEN."
160 CHAR,4,14,"THE VECTORS MUST BE IN 3D SPACE."
170 CHAR,4,17,"PRESS ANY KEY TO CONTINUE."
180 GETKEY A$
190 REM*****************************
200 DO
210 FOR N=0 TO 2:U(N)=0:V(N)=0:NEXT
220 SCNCLR
230 CHAR,4,4,"ENTER COMPONENTS OF FIRST VECTOR"
240 CHAR,0,7,""
250 FOR N=0 TO 2
260 PRINT TAB(8)"COMPONENT";N+1;:INPUT U(N)
270 PRINT
```

```
280 NEXT
290 REM*******************************
300 SCNCLR
310 CHAR,4,4,"ENTER COMPONENTS OF SECOND VECTOR"
320 CHAR,0,7,""
330 FOR N=0 TO 2
340 PRINT TAB(8)"COMPONENT";N+1;:INPUT V(N)
350 PRINT
360 NEXT
370 REM*******************************
380 REM DOT PRODUCT
390 D=0
400 FOR N=0 TO 2
410 D=D+U(N)*V(N)
420 NEXT
430 REM*******************************
440 REM CROSS PRODUCT
450 W(0)=U(1)*V(2)-U(2)*V(1)
460 W(1)=U(2)*V(0)-U(0)*V(2)
470 W(2)=U(0)*V(1)-U(1)*V(0)
480 REM*******************************
490 DEF FNA(X)=INT(X*100+.5)/100
500 SCNCLR:PRINT:PRINT
510 PRINT TAB(13)"CROSS PRODUCT"
520 PRINT TAB(13)"*************"
530 PRINT
540 PRINT TAB(4)"VECTOR 1"SPC(4)"VECTOR 2"SPC(4)"PRODUCT"
550 PRINT TAB(4)"--------"SPC(4)"--------"SPC(4)"-------"
560 FOR N=0 TO 2
570 U(N)=FNA(U(N))
580 V(N)=FNA(V(N))
590 W(N)=FNA(W(N))
600 PRINT TAB(4);U(N);TAB(16);V(N);TAB(28);W(N)
610 NEXT
620 PRINT:PRINT:PRINT
630 PRINT TAB(14)"DOT PRODUCT"
640 PRINT TAB(14)"***********"
650 PRINT
660 PRINT TAB(8)"DOT PRODUCT IS ";FNA(D)
670 PRINT:PRINT:PRINT
680 PRINT TAB(8):INPUT"ANOTHER PRODUCT (Y/N)";Y$
690 LOOP UNTIL ASC(Y$)<>89
700 END
710 REM*******************************
720 REM*******************************
```

P93 Quadratic equations

In this program we have to solve:

$$Ax^2 + Bx + C = 0 \qquad (I)$$

To do this we use the formula:

$$x = (-B \pm SQR(B*B - 4*A*C))/2/A \qquad (II)$$

This gives the two roots of (I). There are, however, some problems:

1. If A=0 we have division by zero in equation (II).
   In this case the solution is x = C/B.

2. If B*B - 4*A*C = 0 we have only one root.
   In this case the solution is x = -B/(2*A).

3. If B*B - 4*A*C < 0 we have complex roots.

COMMANDS

Key in the program and RUN.
Enter the coefficients in the correct order when prompted.

```
10 REM QUADRATIC EQUATIONS
20 REM ********************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,7,1:REM BLUE BORDER
70 SCNCLR
80 PRINT:PRINT:PRINT
90 PRINT TAB(11)"QUADRATIC EQUATIONS"
100 PRINT TAB(11)"====================="
110 PRINT:PRINT
120 PRINT TAB(4)"THIS PROGRAM SOLVES EQUATIONS OF"
130 PRINT TAB(4)"THE FORM:"
```

```
140 PRINT
150 PRINT TAB(13)"A*X^2+B*X+C=0"
160 PRINT
170 PRINT TAB(4)"ENTER THE THREE PARAMETERS IN THE"
180 PRINT TAB(4)"CORRECT ORDER."
190 PRINT
200 PRINT TAB(4):INPUT"COEFFICIENT OF X^2 (A)=";A
210 PRINT TAB(4):INPUT"COEFFICIENT OF X (B)  =";B
220 PRINT TAB(4):INPUT"CONSTANT TERM (C)     =";C
230 IF A=0 AND B=0 THEN RUN:REM NOT A SENSIBLE CONDITION
240 REM*****************************
250 IF A<>0 THEN A$=STR$(A)+"*X^2"
260 IF B>0 THEN B$=MID$(STR$(B),2)+"*X"
270 IF A<>0 AND B>0 THEN B$="+"+B$
280 IF B<0 THEN B$=STR$(B)+"*X"
290 IF C>0 THEN C$="+"+MID$(STR$(C),2)
300 IF C<0 THEN C$=STR$(C)
310 E$=A$+B$+C$+"=0"
320 PRINT
330 PRINT TAB(4)"EQUATION IS "E$
340 PRINT
350 PRINT TAB(4):INPUT"IS THIS CORRECT (Y/N)";Y$
360 IF ASC(Y$)<>89 THEN RUN
370 REM*****************************
380 SCNCLR:PRINT:PRINT:PRINT:PRINT
390 PRINT TAB(4)"EQUATION IS "E$
400 PRINT:PRINT
410 F=0
420 IF A=0 THEN PRINT TAB(4)"SOLUTION IS: X =";-C/B:F=1
430 REM*****************************
440 DO WHILE F=0
450 :  D=B*B-4*A*C:REM DISCRIMINANT
460 :  IF D=0 THEN GOSUB 670
470 :  IF D>0 THEN GOSUB 730
480 :  IF D<0 THEN GOSUB 810
490 :  F=1
500 LOOP
510 REM*****************************
520 PRINT:PRINT TAB(9)"********************"
530 PRINT:PRINT:PRINT
540 PRINT TAB(4):INPUT"WANT TO SOLVE ANOTHER (Y/N)";Y$
550 IF ASC(Y$)=89 THEN RUN
560 END
570 REM*****************************
580 REM*****************************
```

```
590 REM
600 REM       ***************
610 REM       *             *
620 REM       * SUBROUTINES *
630 REM       *             *
640 REM       ***************
650 REM
660 REM EQUAL ROOTS
670 PRINT TAB(4)"WE HAVE EQUAL ROOTS"
680 PRINT:PRINT
690 PRINT TAB(4)"THE SOLUTION IS: X =";-B/2/A
700 RETURN
710 REM*****************************
720 REM REAL ROOTS
730 PRINT TAB(4)"WE HAVE TWO REAL ROOTS"
740 PRINT:PRINT
750 PRINT TAB(4)"ROOT 1 IS: X =";(-B+SQR(D))/2/A
760 PRINT
770 PRINT TAB(4)"ROOT 2 IS: X =";(-B-SQR(D))/2/A
780 RETURN
790 REM*****************************
800 REM COMPLEX ROOTS
810 PRINT TAB(4)"WE HAVE COMPLEX ROOTS"
820 PRINT:PRINT
830 PRINT TAB(4)"ROOT 1 IS:"
840 PRINT TAB(4)"X =";-B/2/A;"+I*(";SQR(-D)/2/A;")"
850 PRINT
860 PRINT TAB(4)"ROOT 2 IS:"
870 PRINT TAB(4)"X =";-B/2/A;"-I*(";SQR(-D)/2/A;")"
880 RETURN
890 REM*****************************
900 REM*****************************
```

P94 Factorization

This program finds the prime factors of positive integers. Any positive integer N may be expressed in terms of prime numbers and indices. The index of a prime number is the power to which that number is raised in the factorization of N.

For example:

180 = 2^2 * 3^2 * 5

We use a method of repeated division to find the set of factors for N.

Let us consider an example - find the prime factors of 180.

The first possible factor of 180 is 2, and we can write:

180 = 2*90 = 2*2*45

Thus 2 is a factor of 180 and of 90, but it is not a factor of 45. That is to say that dividing 45 by two gives a remainder which is not zero.

We can now try 3:

180 = 2*2*3*15 = 2*2*3*3*5

Trying the next prime, 5, gives a result on division of 1. This indicates that all the factors have been found.

Thus:

180 = 2*2*3*3*5*1 = 2^2 * 3^3 * 5

The program presented here uses this algorithm.

COMMANDS

Key in the program and RUN.
Enter the number to be factorized.

```
10 REM FACTORIZATION
20 REM *************
30 REM
40 COLOR 0,2:REM WHITE PAPER
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,15,1:REM BLUE BORDER
70 SCNCLR
80 PRINT:PRINT:PRINT
90 PRINT TAB(13)"FACTORIZATION"
100 PRINT TAB(13)"*************"
110 PRINT:PRINT
120 PRINT TAB(4)"THIS PROGRAM CAN BE USED TO"
130 PRINT TAB(4)"FACTORIZE A POSTIVE INTEGER INTO"
140 PRINT TAB(4)"PRIME FACTORS."
150 PRINT
160 PRINT TAB(4)"IN ITS PRESENT FORM THE PROGRAM"
170 PRINT TAB(4)"USES ONLY THE PRIMES LESS THAN"
180 PRINT TAB(4)"100."
190 PRINT:PRINT
200 PRINT TAB(4):INPUT"NUMBER TO BE FACTORIZED";N
210 IF N<2 THEN RUN:REM TRIVIAL ENTRIES OF 1 OR 2 IGNORED
220 N=INT(ABS(N)):Q=N
230 D=100:REM CHANGE THIS LINE TO ALTER THE PROGRAM'S RANGE
240 DIM F(D):DIM I(D)
250 REM*******************************
260 REM FIND THE INDICES OF THE FACTORS
270 FOR K=2 TO D
280 :  DO
290 :     T=Q-INT(Q/K)*K
300 :     IF T=0 THEN F(K)=1:I(K)=I(K)+1:Q=INT(Q/K)
310 :  LOOP UNTIL T<>0
320 NEXT
330 REM IF THERE IS A 1 IN THE KTH
340 REM POSITION OF THE FACTOR ARRAY
350 REM THEN K IS A FACTOR AND I(K) IS
360 REM THE INDEX OF THAT FACTOR.
370 REM*******************************
380 REM THE NEXT SECTION OF CODE WRITES
390 REM OUT THE FACTORIZATION. NOTE
400 REM THAT THE FIRST 1 IS WRITTEN
410 REM JUST TO TIDY UP THE DISPLAY,
420 REM EVEN THOUGH IT MIGHT NOT BE
430 REM CONSIDERED A PRIME NUMBER.
440 SCNCLR:PRINT:PRINT:PRINT
450 C=1:A$="1"
460 FOR K=2 TO D
470 :  B$=" *"+STR$(K)+"^"+MID$(STR$(I(K)),2)
```

```
480 :  IF F(K)=1 THEN A$=A$+B$:C=C*K^I(K)
490 NEXT
500 REM*******************************
510 IF C=>N THEN GOSUB 670
520 IF C<N THEN GOSUB 780
530 PRINT:PRINT
540 PRINT TAB(8):INPUT"ANOTHER RUN (Y/N)";Y$
550 IF ASC(Y$)=89 THEN RUN
560 END
570 REM*******************************
580 REM*******************************
590 REM
600 REM       ***************
610 REM       *             *
620 REM       * SUBROUTINES *
630 REM       *             *
640 REM       ***************
650 REM
660 REM FACTORIZATION OK
670 PRINT TAB(4)"THE PRIME FACTORIZATION OF"
680 PRINT
690 PRINT TAB(3)N
700 PRINT
710 PRINT TAB(4)"IS:"
720 PRINT
730 PRINT TAB(4)A$
740 RETURN
750 REM*****************************
760 REM*****************************
770 REM FACTORIZATION INCORRECT
780 PRINT TAB(4)"EITHER THE NUMBER HAS A PRIME"
790 PRINT TAB(4)"FACTOR GREATER THAN 100 OR THE"
800 PRINT TAB(4)"ROUNDING ERRORS OF THE MICRO"
810 PRINT TAB(4)"HAVE MUDDLED THE CALCULATION."
820 RETURN
830 REM*****************************
840 REM*****************************
```

P95 Factorial

In statistics we frequently wish to calculate objects of the form:

N * (N-1) * (N-2) * ..... * 3 * 2 * 1

For example, if we want to know the number of ways of arranging the letters in the word COMPUTER, then:

We have 8 ways of choosing the first letter;
We have 7 ways of choosing the second letter;
We have 6 ways of choosing the third letter;

and so on.

Thus in total we have:

8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 40320

ways of arranging the letters of the word COMPUTER.

Such objects are known as factorials, and are defined as follows:

N! = N * (N-1) * (N-2) * ..... * 3 * 2 * 1

Where ! is the symbol for factorial.

COMMANDS

Key in the program and RUN.
Follow the instructions.

```
10 REM FACTORIAL
20 REM *********
30 REM
40 COLOR 0,2:REM WHITE PAPER
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,15,1:REM BLUE BORDER
70 SCNCLR
80 PRINT
```

```
90 PRINT TAB(13)"!!!!!!!!!!!!!!"
100 PRINT TAB(13)"! FACTORIAL !"
110 PRINT TAB(13)"!!!!!!!!!!!!!!"
120 PRINT:PRINT
130 PRINT TAB(4)"THIS PROGRAM MAY BE USED TO"
140 PRINT TAB(4)"EVALUATE THE THE FACTORIAL OF A"
150 PRINT TAB(4)"POSITIVE INTEGER LESS THAN OR"
160 PRINT TAB(4)"EQUAL TO 33."
170 PRINT
180 PRINT TAB(4)"THE PROGRAM USES THE FORMULA:"
190 PRINT
200 PRINT TAB(4)"N!=N*(N-1)*(N-2)*....*3*2*1."
210 PRINT
220 PRINT TAB(4)"THE LIMITATION OF 33 IS BECAUSE"
230 PRINT TAB(4)"OF THE LIMITED RANGE OF NUMBERS"
240 PRINT TAB(4)"WHICH CAN BE HELD INSIDE A"
250 PRINT TAB(4)"COMPUTER."
260 PRINT:PRINT
270 PRINT TAB(4)"PRESS ANY KEY TO CONTINUE."
280 GETKEY A$
290 REM*****************************
300 DO
310 :   DO
320 :     SCNCLR
330 :     PRINT:PRINT:PRINT:PRINT
340 :     PRINT TAB(4)"PLEASE ENTER A POSITIVE WHOLE"
350 :     PRINT TAB(4)"NUMBER LESS THAN OR EQUAL TO 33."
360 :     PRINT:PRINT
370 :     PRINT TAB(7):INPUT"WHAT IS YOUR NUMBER";N%
380 :   LOOP UNTIL N%<34 AND N%>0
390 :   REM*************************
400 :   F=1
410 :   FOR K=1 TO N%
420 :      F=F*K
430 :   NEXT
440 :   REM*************************
450 :   PRINT:PRINT
460 :   PRINT TAB(7)"THE FACTORIAL OF";N%;"IS:"
470 :   PRINT
480 :   PRINT TAB(6)F
490 :   PRINT TAB(7)"----------------------"
500 :   REM*************************
510 :   PRINT:PRINT:PRINT
520 :   PRINT TAB(7):INPUT"WANT ANOTHER NUMBER (Y/N)";Y$
530 LOOP UNTIL ASC(Y$)<>89
540 END
550 REM*****************************
```

P96 Greatest common divisor

This program uses the Euclidian Algorithm to compute the greatest common divisor of two natural numbers.

COMMANDS

Key in the program and RUN.
Enter numbers as positive integers.

```
10 REM GREATEST COMMON DIVISOR
20 REM ***********************
30 REM
40 COLOR 0,2:REM WHITE PAPER
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,15,1:REM BLUE BORDER
70 SCNCLR
80 PRINT:PRINT
90 PRINT TAB(6)"////////////////////////////"
100 PRINT TAB(6)"/ GREATEST COMMON DIVISOR /"
110 PRINT TAB(6)"////////////////////////////"
120 PRINT:PRINT
130 PRINT TAB(4)"THIS PROGRAM USES THE EUCLIDEAN"
140 PRINT TAB(4)"ALGORITHM TO COMPUTE THE GREATEST"
150 PRINT TAB(4)"COMMON DIVISOR OF TWO NATURAL"
160 PRINT TAB(4)"NUMBERS."
170 PRINT:PRINT
180 PRINT TAB(10):INPUT"FIRST NUMBER";X1
190 PRINT
200 PRINT TAB(10):INPUT"SECOND NUMBER";X2
210 REM*****************************
220 X1=ABS(INT(X1)):A=X1
230 X2=ABS(INT(X2)):B=X2
240 IF A=0 OR B=0 THEN RUN:PREVENT DIVISION BY ZERO
250 IF A<B THEN T=B:B=A:A=T:REM A IS THE LARGER NUMBER
260 REM*****************************
270 REM THE FOLLOWING IS THE EUCLIDIAN ALGORITHM:
280 DO
290 :  R=A-(INT(A/B))*B:REM REMAINDER
300 :  Q=INT(A/B):REM QUOTIENT
310 :  A=B:B=R
320 LOOP UNTIL R=0
```

```
330 REM**************************
340 SCNCLR
350 PRINT:PRINT:PRINT:PRINT
360 PRINT TAB(9)"THE GREATEST COMMON"
370 PRINT TAB(9)"DIVISOR OF"
380 PRINT
390 PRINT TAB(8)X1;"AND"
400 PRINT
410 PRINT TAB(8)X2;"IS:"
420 PRINT
430 PRINT TAB(8)A
440 PRINT:PRINT
450 PRINT TAB(9):INPUT"ANOTHER RUN (Y/N)";Y$
460 IF ASC(Y$)=89 THEN RUN
470 END
480 REM**************************
490 REM**************************
```

P97 Polynomial multiplication

This program allows the user to multiply two polynomials together.

Example

Multiply (2*x*x + 3*x + 2) by (x + 1)

If this has to be done by hand we proceed as follows:

$$\begin{array}{r} 2x^2 + 3x + 2 \\ \underline{\qquad\quad x + 1} \\ 2x^2 + 3x + 2 \\ \underline{2x^3 + 3x^2 + 2x\quad\;\;} \\ 2x^3 + 5x^2 + 5x + 2 \\ \hline\hline \end{array}$$

This can be a rather time consuming exercise when the polynomials become large. This program takes all the work out of it.

The Commodore 16, in common with most other micros, cannot express polynomials in a very satisfying manner, but bear with this and the algorithm can be very useful.

COMMANDS

Key in the program and RUN.
Follow instructions, keying in the parameters as prompted.

```
10 REM POLYNOMIAL MULTIPLICATION
20 REM ************************
30 REM
40 COLOR 0,2:REM WHITE PAPER
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,15,1:REM BLUE BORDER
70 SCNCLR
```

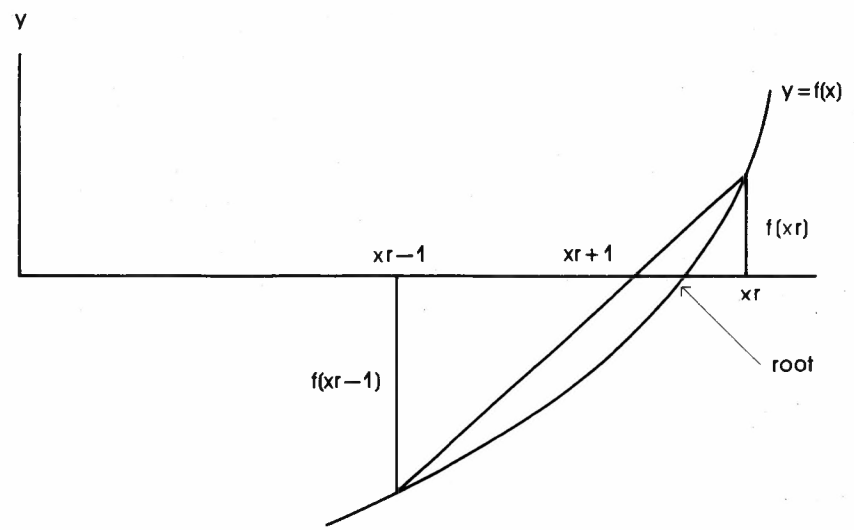
```
80 PRINT:PRINT
90 PRINT TAB(7)"POLYNOMIAL MULTIPLICATION"
100 PRINT TAB(7)"************************"
110 PRINT
120 PRINT TAB(4)"THIS PROGRAM CAN BE USED TO FIND"
130 PRINT TAB(4)"THE RESULT OF MULTIPLYING TWO"
140 PRINT TAB(4)"POLYNOMIALS TOGETHER."
150 PRINT
160 PRINT TAB(4)"THE POLYNOMIALS ARE OF THE FORM:"
170 PRINT
180 PRINT TAB(4)"P(X)=A(0)*X^0+A(1)*X^1+A(2)*X^2"
190 PRINT TAB(9)"+....+A(N)*X^N"
200 PRINT
210 PRINT TAB(4)"Q(X)=B(0)*X^0+B(1)*X^1+B(2)*X^2"
220 PRINT TAB(9)"+....+B(M)*X^M"
230 PRINT
240 PRINT TAB(4)"AND THE RESULT IS:"
250 PRINT
260 PRINT TAB(4)"P(X)*Q(X)=C(0)*X^0+C(1)*X^1+C(2)"
270 PRINT TAB(14)"*X^2+..+C(M+N)*X^(M+N)"
280 PRINT:PRINT
290 PRINT TAB(7)"PRESS ANY KEY TO CONTINUE."
300 GETKEY A$
310 REM*****************************
320 SCNCLR:PRINT:PRINT:PRINT:PRINT
330 PRINT TAB(4)"YOU ARE REQUIRED TO ENTER ONLY"
340 PRINT TAB(4)"THE DEGREE AND THE COEFFICIENTS"
350 PRINT TAB(4)"OF EACH POLYNOMIAL."
360 PRINT:PRINT
370 PRINT TAB(7)"PRESS ANY KEY TO CONTINUE."
380 GETKEY A$
390 REM*****************************
400 DIM A(50):DIM B(50):DIM C(100)
410 REM POLYNOMIALS OF DEGREE > 50 NOT ACCEPTED
420 REM THIS IS ABOVE ANY NORMAL PRACTICAL LIMIT
430 REM*****************************
440 DO
450 :  DO
460 :     SCNCLR:PRINT:PRINT:PRINT:PRINT
470 :     PRINT TAB(9)"WHAT IS THE DEGREE OF"
480 :     PRINT TAB(9):INPUT"POLYNOMIAL 1";N%
490 :  LOOP UNTIL N%>0 AND N%<51
500 :  FOR K=0 TO N%
510 :     A(K)=0
520 :     PRINT
530 :     PRINT TAB(9)"COEFFICIENT";K;"=";
540 :     INPUT A(K)
```

```
550 :  NEXT
560 :  REM*************************
570 :  DO
580 :     SCNCLR:PRINT:PRINT:PRINT:PRINT
590 :     PRINT TAB(9)"WHAT IS THE DEGREE OF"
600 :     PRINT TAB(9):INPUT"POLYNOMIAL Z";M%
610 :  LOOP UNTIL M%>0 AND M%<51
620 :  FOR K=0 TO M%
630 :     B(K)=0
640 :     PRINT
650 :     PRINT TAB(9)"COEFFICIENT";K;"=";
660 :     INPUT B(K)
670 :  NEXT
680 :  REM*************************
685 :  FOR K=1 TO M%+N%:C(K)=0:NEXT
690 :  FOR J=0 TO N%
700 :     FOR K=0 TO M%
710 :        C(K+J)=C(K+J)+B(K)*A(J)
720 :     NEXT
730 :  NEXT
740 :  REM*************************
750 :  SCNCLR:PRINT:PRINT:PRINT:PRINT
760 :  PRINT"THE RESULT OF MULTIPLYING"
770 :  PRINT
780 :  FOR J=N% TO 0 STEP-1
790 :     PRINT A(J);"*X^";J;"+";
800 :  NEXT
810 :  PRINT CHR$(157)CHR$(32):REM DELETE LAST +
820 :  PRINT:PRINT"BY":PRINT
830 :  FOR K=M% TO 0 STEP-1
840 :     PRINT B(K);"*X^";K;"+";
850 :  NEXT
860 :  PRINT CHR$(157)CHR$(32):REM DELETE LAST +
870 :  PRINT:PRINT"IS":PRINT
880 :  FOR L=M%+N% TO 0 STEP-1
890 :     PRINT C(L);"*X^";L;"+";
900 :  NEXT
910 :  PRINT CHR$(157)CHR$(32):REM DELETE LAST +
920 :  REM*************************
930 :  PRINT:PRINT
940 :  PRINT TAB(10):INPUT"ANOTHER (Y/N)";Y$
950 LOOP UNTIL ASC(Y$)<>89
960 END
970 REM*****************************
980 REM*****************************
```

P98 Secant method

This program can be used to find a root of a function of a single variable. The Secant method can be interpreted geometrically as follows:

Consider the diagram:



If xr and xr-1 lie on either side of a root we can draw the secant between the points (xr-1,f(xr-1)) and (xr,f(xr)). The secant cuts the x-axis at the point xr+1. Let xr+1 be the new approximation.

Then by similar triangles:

$$\frac{x_{r+1} - x_{r-1}}{-f(x_{r-1})} = \frac{x_r - x_{r+1}}{f(x_r)}$$

$$\therefore \quad x_{r+1} = x_r - f(x_r) \quad \frac{(x_r - x_{r-1})}{f(x_r) - f(x_{r-1})}$$

The same relationship can be formed even if both approximations lie on the same side of the root.

COMMANDS

Key in the program and RUN.
Enter function and initial approximations when required.
Enter accuracy when prompted.

```
10 REM SECANT METHOD
20 REM *************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,15,1:REM BLUE BORDER
70 SCNCLR
80 CHAR,13,2,"SECANT METHOD"
90 CHAR,13,3,"*************"
100 CHAR,4,6,"THIS PROGRAM USES THE SECANT"
110 CHAR,4,7,"METHOD TO FIND THE ROOT OF AN"
120 CHAR,4,8,"EQUATION."
130 CHAR,4,10,"NOTE THAT THE METHOD WILL ATTEMPT"
140 CHAR,4,11,"TO FIND A ROOT WITHIN AN INTERVAL"
150 CHAR,4,12,"EVEN IF NO ROOT EXISTS."
160 CHAR,7,15,"PRESS ANY KEY TO CONTINUE."
170 GETKEY A$
180 REM******************************
190 SCNCLR
200 CHAR,4,3,"THE METHOD REQUIRES THAT YOU"
210 CHAR,4,4,"INPUT THE FUNCTION, THE RANGE OF"
220 CHAR,4,5,"X VALUES WITHIN WHICH THE ROOT"
230 CHAR,4,6,"LIES AND THE REQUIRED ACCURACY."
240 CHAR,7,10,"PRESS ANY KEY TO CONTINUE."
250 GETKEY A$
260 REM******************************
270 SCNCLR
280 CHAR,4,2,"TO INPUT THE FUNCTION PLEASE"
290 CHAR,4,3,"TYPE IT IN DIRECTLY AFTER"
300 CHAR,4,5,"410 DEF FNA(X)="
310 CHAR,4,7,"THEN PRESS RETURN."
320 CHAR,4,9,"AFTER YOU HAVE DONE THIS TYPE IN"
330 CHAR,4,11,"RUN 410"
340 CHAR,4,13,"THEN PRESS RETURN. USE THE CURSOR"
```

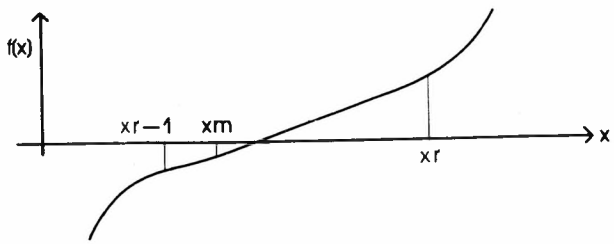
```
350 CHAR,4,14,"CONTROL KEYS TO POSITION THE"
360 CHAR,4,15,"CURSOR DIRECTLY AFTER THE = SIGN"
370 CHAR,4,18,"410 DEF FNA(X)="
380 PRINT:PRINT
390 STOP
400 REM*****************************
410 REM THIS LINE IS REPLACED BY THE FUNCTION ENTERED
420 REM*****************************
430 DO
440 :  SCNCLR
450 :  PRINT:PRINT
460 :  PRINT TAB(8):INPUT"FIRST POINT-   X=";P1
470 :  PRINT
480 :  PRINT TAB(8):INPUT"SECOND POINT-  X=";P2
490 LOOP UNTIL P1<>P2:REM ZERO RANGE NOT ACCEPTED
500 PRINT
510 PRINT TAB(8):INPUT"ACCURACY (+ OR -)";AC
520 PRINT:PRINT
530 REM*****************************
540 FOR R=1 TO 40
550 :  F1=FNA(P1):F2=FNA(P2)
560 :  P3=P2-F2*(P2-P1)/(F2-F1)
570 :  P1=P2:P2=P3
580 :  IF ABS(P2-P1)<AC THEN GOSUB 740:F=1:R=40
590 NEXT
600 IF F=0 THEN GOSUB 820
610 REM*************************
620 PRINT:PRINT
630 PRINT TAB(8):INPUT"ANOTHER RUN (Y/N)";Y$
640 IF ASC(Y$)=89 THEN RUN
650 END
660 REM
670 REM        ***************
680 REM        *             *
690 REM        * SUBROUTINES *
700 REM        *             *
710 REM        ***************
720 REM
730 REM ROOT FOUND WITHIN LIMITS
740 PRINT TAB(8)"ROOT IS";P2
750 PRINT
760 PRINT TAB(8)"AT ITERATION"
770 PRINT TAB(8)"NUMBER";R
780 RETURN
790 REM*****************************
800 REM*****************************
810 REM ROOT NOT FOUND
```

```
820 PRINT TAB(8)"ROOT NOT FOUND AFTER"
830 PRINT TAB(8)"40 ITERATIONS."
840 PRINT
850 PRINT TAB(8)"IF ROOT EXISTS NEAR"
860 PRINT TAB(8)"THE RANGE ENTERED,"
870 PRINT TAB(8)"IT LIES BETWEEN:"
880 PRINT
890 PRINT TAB(7)P1
900 PRINT TAB(8)"AND"
910 PRINT TAB(7)P2
920 RETURN
930 REM*****************************
940 REM*****************************
```

P99 Method of bisections

The method of bisections is based on the use of sign changes to find a root of a function.

Consider the following diagram:



If we have two points, xr and xr-1, such that f(xr) and f(xr-1) have different signs, then there is a root between xr and xr-1. We then evaluate the function at the mid point, xm, between xr and xr-1. If xm=0 then we have a root. If SGN(f(xm)) does not equal SGN(f(xr-1)) then the root lies between xm and xr-1. Otherwise the root lies between xm and xr.

That is the idea behind the method of bisections, which is used in this program.

COMMANDS

Key in the program and RUN.
Enter the function and the end points of the interval straddling the root when prompted.
Enter the accuracy desired.

```
10 REM METHOD OF BISECTIONS
20 REM ********************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,1:REM BLACK INK
60 COLOR 4,15,1:REM BLUE BORDER
70 SCNCLR
80 CHAR,9,2,"METHOD OF BISECTIONS"
90 CHAR,9,3,"********************"
100 CHAR,4,6,"THIS PROGRAM USES THE METHOD OF"
110 CHAR,4,7,"BISECTIONS TO FIND THE ROOT OF"
120 CHAR,4,8,"AN EQUATION."
130 CHAR,4,10,"NOTE THAT THE METHOD WILL ATTEMPT"
140 CHAR,4,11,"TO FIND A ROOT WITHIN AN INTERVAL"
150 CHAR,4,12,"EVEN IF NO ROOT EXISTS."
160 CHAR,7,15,"PRESS ANY KEY TO CONTINUE."
170 GETKEY A$
180 REM******************************
190 SCNCLR
200 CHAR,4,3,"THE METHOD REQUIRES THAT YOU"
210 CHAR,4,4,"INPUT THE FUNCTION, THE RANGE OF"
220 CHAR,4,5,"X VALUES WITHIN WHICH THE ROOT"
230 CHAR,4,6,"LIES AND THE REQUIRED ACCURACY."
240 CHAR,7,10,"PRESS ANY KEY TO CONTINUE."
250 GETKEY A$
260 REM******************************
270 SCNCLR
280 CHAR,4,2,"TO INPUT THE FUNCTION PLEASE"
290 CHAR,4,3,"TYPE IT IN DIRECTLY AFTER"
300 CHAR,4,5,"410 DEF FNA(X)="
310 CHAR,4,7,"THEN PRESS RETURN."
320 CHAR,4,9,"AFTER YOU HAVE DONE THIS TYPE IN"
330 CHAR,4,11,"RUN 410"
340 CHAR,4,13,"THEN PRESS RETURN. USE THE CURSOR"
350 CHAR,4,14,"CONTROL KEYS TO POSITION THE"
360 CHAR,4,15,"CURSOR DIRECTLY AFTER THE = SIGN"
370 CHAR,4,18,"410 DEF FNA(X)="
380 PRINT:PRINT
390 STOP
400 REM******************************
410 REM THIS LINE IS REPLACED BY THE FUNCTION ENTERED
420 REM******************************
430 DO
440 :  SCNCLR
450 :  PRINT:PRINT
460 :  PRINT TAB(8):INPUT"FIRST POINT-   X=";P1
470 :  PRINT
```

```
480 :  PRINT TAB(8):INPUT"SECOND POINT-  X=";P2
490 LOOP UNTIL P1<>P2:REM ZERO RANGE NOT ACCEPTED
500 PRINT
510 PRINT TAB(8):INPUT"ACCURACY (+ OR -)";AC
520 PRINT:PRINT
530 REM*****************************
540 FOR R=1 TO 40
550 :  T=(P1+P2)/2:K=P1
560 :  IF SGN(FNA(T))=SGN(FNA(K)) THEN P1=T:ELSE P2=T
570 :  IF FNA(T)=0 THEN PRINT TAB(8)"SOLUTION IS ";T
580 :  IF FNA(T)=0 THEN R=40:SOL=3:ELSE SOL=0
590 :  DO WHILE SOL=0
600 :     IF SGN(FNA(P1))=SGN(FNA(P2)) THEN FG=1:ELSE FG=0
610 :     IF ABS(P1-P2)<AC AND FG=0 THEN R=40:SOL=1
620 :     IF SOL=0 THEN SOL=2
630 :  LOOP
640 NEXT
650 ON SOL GOSUB 790,880
660 REM*************************
670 PRINT:PRINT
680 PRINT TAB(8):INPUT"ANOTHER RUN (Y/N)";Y$
690 IF ASC(Y$)=89 THEN RUN
700 END
710 REM
720 REM       ***************
730 REM       *             *
740 REM       * SUBROUTINES *
750 REM       *             *
760 REM       ***************
770 REM
780 REM ROOT FOUND WITHIN LIMITS
790 PRINT TAB(8)"ROOT LIES BETWEEN"
800 PRINT
810 PRINT TAB(7)P1
820 PRINT TAB(8)"AND"
830 PRINT TAB(7)P2
840 RETURN
850 REM*****************************
860 REM*****************************
870 REM ROOT NOT FOUND
880 PRINT TAB(8)"ROOT NOT FOUND AFTER"
890 PRINT TAB(8)"40 ITERATIONS."
900 RETURN
910 REM*****************************
920 REM*****************************
```

P100 Trapezoidal rule

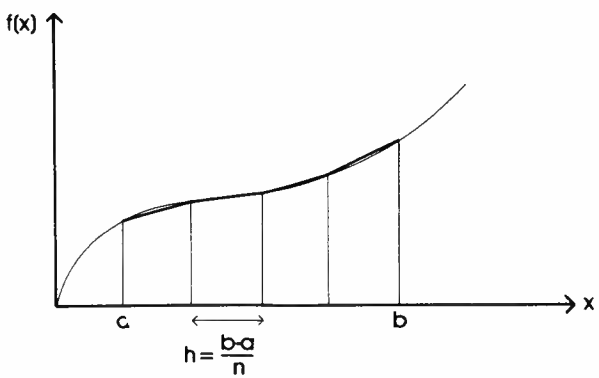
This program uses the trapezoidal rule to evaluate a definite integral of the form:

$$I = \int_a^b f(x)dx$$

Thus the program requires as input:

f(x)
a and b

A definite integral can be considered to be the area under the graph of a function. The trapezium rule approximates this area by a series of trapeziums, as in the following diagram:



COMMANDS

Key in the program and RUN.
Follow the instructions.

```
10 REM TRAPEZOIDAL RULE
20 REM ***************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,1:REM BLACK INK
60 COLOR 1,15,1:REM BLUE BORDER
70 SCNCLR
80 PRINT:PRINT
90 PRINT TAB(11)"TRAPEZOIDAL RULE"
100 PRINT TAB(11)"***************"
110 PRINT
120 PRINT TAB(4)"THIS PROGRAM USES THE TRAPEZOIDAL"
130 PRINT TAB(4)"RULE TO EVALUATE A DEFINITE"
140 PRINT TAB(4)"INTEGRAL."
150 PRINT
160 PRINT TAB(4)"YOU ARE REQUIRED TO ENTER:"
170 PRINT
180 PRINT TAB(8) "1. YOUR FUNCTION OF X."
190 PRINT
200 PRINT TAB(8)"2. THE RANGE OF VALUES"
210 PRINT TAB(11)"OF X OVER WHICH YOU"
220 PRINT TAB(11)"WISH TO EVALUATE THE"
230 PRINT TAB(11)"INTEGRAL."
240 PRINT
250 PRINT TAB(8)"3. THE NUMBER OF SUB-"
260 PRINT TAB(11)"INTERVALS."
270 PRINT
280 PRINT
290 PRINT TAB(7)"PRESS ANY KEY TO CONTINUE."
300 GETKEY A$
310 REM*****************************
320 SCNCLR:PRINT:PRINT
330 PRINT TAB(4)"TO INPUT THE FUNCTION PLEASE"
340 PRINT TAB(4)"TYPE IT IN DIRECTLY AFTER"
350 PRINT
360 PRINT TAB(4)"520 DEF FNA(X)="
370 PRINT
380 PRINT TAB(4)"THEN PRESS RETURN."
390 PRINT
```

```
400 PRINT TAB(4)"AFTER YOU HAVE DONE THIS TYPE IN"
410 PRINT
420 PRINT TAB(4)"RUN 520"
430 PRINT
440 PRINT TAB(4)"THEN PRESS RETURN. USE THE CURSOR"
450 PRINT TAB(4)"CONTROL KEYS TO POSITION THE"
460 PRINT TAB(4)"CURSOR DIRECTLY AFTER THE = SIGN"
470 PRINT:PRINT
480 PRINT TAB(4)"520 DEF FNA(X)="
490 PRINT:PRINT
500 STOP
510 REM*****************************
520 DEF FNA(X)=X
530 REM*****************************
540 DO
550 :  SCNCLR:PRINT:PRINT
560 :  PRINT TAB(6):INPUT"LOWER VALUE OF X-RANGE";A
570 :  PRINT
580 :  PRINT TAB(6):INPUT"HIGHER VALUE OF X-RANGE";B
590 LOOP UNTIL B>A
600 PRINT
610 PRINT TAB(6):INPUT"NUMBER OF POINTS";N%
620 IF N%<2 THEN N%=2:REM GUARD AGAINST SILLY ENTRIES
630 REM*****************************
640 PRINT:PRINT
650 H=(ABS(B-A))/N%
660 LO=FNA(A)/2
670 HI=FNA(B)/2
680 X=A:K=0
690 FOR J=1 TO N%-1
700 :  X=X+H
710 :  K=K+FNA(X)
720 NEXT
730 IT=(LO+HI+K)*H
740 PRINT TAB(8)"INTEGRAL =";IT
750 PRINT TAB(8)"-----------------------"
760 PRINT:PRINT
770 REM*****************************
780 PRINT TAB(8):INPUT"ANOTHER RUN (Y/N)";Y$
790 IF ASC(Y$)=89 THEN RUN
800 END
810 REM*****************************
```

P101 Simpson's rule (bonus program)

Simpson's rule is rather more complicated than the trapezoidal rule. Here we use a quadratic curve, rather than a straight line, between the end points of the interval. This leads to the following rule:

$$\int_a^b f(x)dx \simeq \left[\frac{h}{3} f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \ldots + f(x_n)\right]$$

where $x_0 = a$, $x_n = b$ (n is even), $x_i = a + i * h$.

COMMANDS

Key in the program and RUN.
Enter the function as instructed.
Enter a and b when prompted.
Enter an even number of points as requested. If you enter an odd number, 1 will be added.

```
10 REM SIMPSON'S RULE
20 REM **************
30 REM
40 COLOR 0,2:REM WHITE SCREEN
50 COLOR 1,1:REM BLACK INK
60 COLOR 1,15,1:REM BLUE BORDER
70 SCNCLR
80 PRINT:PRINT
90 PRINT TAB(12)"SIMPSON'S RULE"
100 PRINT TAB(12)"**************"
110 PRINT
120 PRINT TAB(4)"THIS PROGRAM USES SIMPSON'S"
130 PRINT TAB(4)"RULE TO EVALUATE A DEFINITE"
140 PRINT TAB(4)"INTEGRAL."
150 PRINT
```

```
160 PRINT TAB(4)"YOU ARE REQUIRED TO ENTER:"
170 PRINT
180 PRINT TAB(8) "1. YOUR FUNCTION OF X."
190 PRINT
200 PRINT TAB(8)"2. THE RANGE OF VALUES"
210 PRINT TAB(11)"OF X OVER WHICH YOU"
220 PRINT TAB(11)"WISH TO EVALUATE THE"
230 PRINT TAB(11)"INTEGRAL."
240 PRINT
250 PRINT TAB(8)"3. THE NUMBER OF SUB-"
260 PRINT TAB(11)"INTERVALS (THIS MUST"
270 PRINT TAB(11)"BE EVEN)."
280 PRINT:PRINT
290 PRINT TAB(7)"PRESS ANY KEY TO CONTINUE."
300 GETKEY A$
310 REM******************************
320 SCNCLR:PRINT:PRINT
330 PRINT TAB(4)"TO INPUT THE FUNCTION PLEASE"
340 PRINT TAB(4)"TYPE IT IN DIRECTLY AFTER"
350 PRINT
360 PRINT TAB(4)"520 DEF FNA(X)="
370 PRINT
380 PRINT TAB(4)"THEN PRESS RETURN."
390 PRINT
400 PRINT TAB(4)"AFTER YOU HAVE DONE THIS TYPE IN"
410 PRINT
420 PRINT TAB(4)"RUN 520"
430 PRINT
440 PRINT TAB(4)"THEN PRESS RETURN. USE THE CURSOR"
450 PRINT TAB(4)"CONTROL KEYS TO POSITION THE"
460 PRINT TAB(4)"CURSOR DIRECTLY AFTER THE = SIGN"
470 PRINT:PRINT
480 PRINT TAB(4)"520 DEF FNA(X)="
490 PRINT:PRINT
500 STOP
510 REM******************************
520 REM THIS LINE IS REPLACED BY THE FUNCTION ENTERED
530 REM******************************
540 DO
550 :  SCNCLR:PRINT:PRINT
560 :  PRINT TAB(6):INPUT"LOWER VALUE OF X-RANGE";A
570 :  PRINT
580 :  PRINT TAB(6):INPUT"HIGHER VALUE OF X-RANGE";B
590 LOOP UNTIL B>A
600 PRINT
610 PRINT TAB(6):INPUT"NUMBER OF POINTS";N%
620 IF N%<4 THEN N%=4:REM GUARD AGAINST SILLY ENTRIES
```

```
630 IF N%-(INT(N%/2))*2<>0 THEN N%=N%+1:REM ENSURE N% EVEN
640 REM*****************************
650 PRINT:PRINT
660 H=(ABS(B-A))/N%
670 FI=FNA(A)/2
680 LA=FNA(B)/2
690 X=A:EV=0:OD=0
700 FOR J=1 TO N%-3 STEP 2
710 :  X=X+H
720 :  OD=OD+FNA(X)
730 :  X=X+H
740 :  EV=EV+FNA(X)
750 NEXT
760 X=X+H
770 OD=OD+FNA(X)
780 IT=(FI+LA+4*OD+2*EV)*H/3
790 PRINT TAB(8)"INTEGRAL =";IT
800 PRINT TAB(8)"--------------------"
810 PRINT:PRINT
820 REM*****************************
830 PRINT TAB(8):INPUT"ANOTHER RUN (Y/N)";Y$
840 IF ASC(Y$)=89 THEN RUN
850 END
860 REM*****************************
870 REM*****************************
```

P102 The last one (bonus program)

This is a rather interesting program. Once you have run the program, have a look at the listing.

COMMANDS

Key in the program and RUN.

```
10 REM THE LAST ONE
20 PRINT CHR$(147):PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
30 PRINT "THIS IS A SOMEWHAT SILLY PROGRAM"
40 PRINT "ALL IT DOES IS DISAPPEAR !!!!"
50 PRINT:PRINT:PRINT "PRESS ANY KEY TO SEE"
60 GET A$:IF A$="" THEN 60
70 GRAPHIC 1,1
80 BG=INT(RND(1)*16+1)
90 FG=INT(RND(1)*16+1):IF BG=FG THEN 80
100 COLOR0,BG:COLOR 1,FG
110 VOL 8
120 FOR B=0 TO 14
130 X=B*8:Y=B*8
140 BOX1,X,Y,319-X,199-Y
150 FG=FG+1
160 IF FG>15 THEN FG=FG-15
170 COLOR 1,FG
180 SOUND 1,1,0
190 SOUND 2,1,0
200 SOUND 1,ABS(1000*SIN(B)),B*2
210 SOUND 1,ABS(1000*COS(B)),B*2
220 NEXT B
230 NEW
240 END
```

PRENTICE
HALL
INTERNATIONAL
PERSONAL
COMPUTER
BOOK

'This is much more than a games book... There's a lot here for the keen programmer, including programs for graphics, user-defined characters and data handling.

For the businessman there's a large section covering tax calculation, stock control, and a suite of programs for a point-of-sale system.

If that doesn't grab you, there are also sections for educational programs, recreation and domestic use, and science... this book's got a great deal to offer.'

Commodore User

£8.95

ISBN 0-13-634965-X

9 780136 349655

0-13-634965-X

cover design: Diana Allen