

**Cambridge
Colour
Collection**

ZX SPECTRUM PROGRAMS

Richard Francis Altwasser

20 Programs for the

ZX Spectrum

Richard Francis Altwasser

Cambridge Colour Collection

Cambridge Colour Collection, 20 programs for the ZX Spectrum

Written and published by
Richard Francis Altwasser
22 Foxhollow
Bar Hill
Cambridge
CB3 8EP

ISBN No. 0 9507658 2 1

No part of this publication may be reproduced without the written consent of the publisher. Copyright 1982 Richard Francis Altwasser.

CASSETTE

The programs in this book are also available on a quality cassette for automatic loading. Please send cheque or P.O. for £2.95 to the publisher.

Printed by The Leagrave Press Ltd, Luton and London

20 Programs for the ZX Spectrum

Richard Francis Altwasser

Contents

Introduction	4
Calendar	5
Maze	7
Lunar Landing	9
Android Nim	12
Morse	15
Maths	20
Biorhythms	22
Telephone	25
Home Accounts	29
Mastermind	34
Breakout	38
Defender	40
Othello	42
Kingdom	46
Fighter	51
Millionaire	53
Rotate	57
Red Cross	59
Graphics	61
Patterns	64

Introduction

Your ZX Spectrum personal computer is probably the most versatile piece of equipment that you possess. This flexibility or programmability is linked, however, with the less desirable quality of stupidity. Your computer is the ultimate personal servant, obedient to the last but also dumb, because without the correct – that is detailed and precisely correct – instructions, it never does anything. Without programs, your ZX Spectrum can be a bore, it won't play games with you, it can't teach you anything and it's useless.

That is why I wrote this collection of programs, to breathe life into your ZX Spectrum.

You can put your brain in combat with the Spectrum with games like 'Othello', 'Mastermind' and 'Red Cross', or you can rely upon your quick reactions and skill with games like 'Breakout', 'Defender' and 'Fighter'.

You can improve your arithmetic with 'Maths' and train yourself to send and receive telegraphy code with 'Morse'.

You can play the stockmarket with 'Millionaire', following trends and investing shrewdly or you can play the role of a nation's leader for 5 years, managing your country's economy with 'Kingdom'.

You can also make real use of your computer by storing your accounts or your telephone numbers, names and addresses and having instant access to pages of information.

Lastly, I am a firm believer that no computer program is perfect, that is not to say that they need have errors, and I don't expect you to come across any bugs in this collection, but I do believe that a program is a solution to a set of objectives, some of which often oppose each other, thus making the program a compromise. If, therefore, you dream up interesting alterations to any of these programs, I will be very pleased to hear from you.

Technical Notes

Throughout the listings in this book the symbol ▲ is used to indicate that an extra space should be inserted. Many of the games in this program rely upon the generation of Random Numbers. If you are loading them from cassette tape you are advised to execute the command **RANDOMIZE** before executing **RUN**.

Calendar

In the mid 1st century B.C. Julius Caesar invited an Alexandrian astronomer, Sosigenes to advise him on reforming the Lunar calendar. Sosigenes suggested that the old system should be abandoned, and a new "Julian Calendar" was started based around a 365½ day year. This was achieved by adding an extra day to February in every fourth year and calling it a leap year. By the mid 16th century, however, the vernal equinox had moved by ten days from its proper position because the Julian basis of 365½ days was 11 minutes and 14 seconds longer than the true tropical year. Pope Gregory XIII was elected in 1572 and sought the advice of a Jesuit astronomer Christopher Clavius to correct the errors. This resulted in losing ten days in October 1572 (the 15th following immediately after the 4th) and a new system, the "Gregorian" calendar in which 3 out of 4 centenary years are not leap years.

The introduction of the Gregorian Calendar into Roman Catholic countries in 1582 was not matched by its simultaneous adoption by protestant nations. This presented a rather startling difficulty for the international traveller. Journeys between England and France, for example, involved flipping the calendar forward by ten days on the outward leg, and back again when homeward bound. It wasn't until 18th March 1751 that the "New Style" Calendar was introduced by act of parliament into Great Britain. In 1752, the 2nd September was followed by the 14th, a loss of 11 days. Before the reforms, the 25th March, the date of the Annunciation of the Blessed Virgin Mary was regarded as New Year's Day in England and Wales. Many financial transactions were traditionally completed on this day, and after the reform this was moved to 5th April, the day we still retain as the end of Britain's tax year.

(Sources: *Encyclopaedia Britannica*, *New Scientist* 25:3:82)

This calendar program uses the Gregorian basis and displays three months at a time.

```
100 BORDER 0: PAPER 0: INK 7: CLS: GO SUB 1000
110 INPUT AT 0,0: FLASH 1: "Year▲▲": LINE Y$: BEEP .5,30: IF LEN y$=4
    THEN FOR y=1 TO 4: IF (y$(y)>="0") AND (y$(y)<="9") THEN NEXT
    y: LET year=VAL y$: IF year>=1572 THEN GO TO 114
111 GO TO 110
114 INPUT AT 1,0: FLASH 1: "Month▲▲": LINE o$: BEEP .5,30: IF
    (o$<>"") AND (LEN o$<=2) THEN FOR o=1 TO LEN o$: IF
    (o$(o)>="0") AND (o$(o)<="9") THEN NEXT o: LET month=VAL o$:
    IF (month>=1) AND (month<=12) THEN GO TO 120
116 GO TO 114
120 CLS: GO SUB 2000
130 GO TO 100
```

N.B. ▲ denotes space, see introduction.

```

1000 DIM m$(12,9): DATA "January","February","March","April","May",
    "June","July","August","September","October","November",
    "December": RESTORE 1000: FOR n=1 TO 12: READ m$(n): NEXT n
1010 DIM d$(7,3): DATA "Sun","Mon","Tue","Wed","Thu","Fri","Sat":
    RESTORE 1010: FOR n=1 TO 7: READ d$(n): NEXT n
1020 DIM L(12): DATA 31,28,31,30,31,30,31,31,30,31,30,31: RESTORE
    1020: FOR n=1 TO 12: READ L(n): NEXT n
1030 DIM c(12): LET c=0: FOR n=1 TO 12: LET c(n)=c: LET c=c+L(n): NEXT n
1040 DEF FN a(y,m)=y+INT(y/4)-INT(y/100)+INT(y/400)+c(m)-(m<3)
    AND (FN b(y))
1050 DEF FN b(y)=(y/4=INT(y/4)) AND (y/100<>INT(y/100)) OR (y/400=INT
    (y/400))
1060 PRINT AT 0,12: INK 4;"CALENDAR" " " "-----
    -----" INK 5;"▲▲This Program will display
    a " " "calendar for any month since " " "October 1572 when the
    Gregorian " " "calendar was introduced"
1070 PRINT " INK 6;"Type in a year as four digits " " "e.g. 1982 followed by
    ENTER " " "Type in a month as a number in " " "the range 1 to 12
    followed by " " "ENTER"
1080 RETURN
2000 LET y=year-(month=1): LET m=month-1+12*(month=1): LET
    draw=0: BRIGHT 0: CLS: GO SUB 3000: BEEP .5,27
2010 LET y=year: LET m=month: LET draw=7: BRIGHT 1: GO SUB 3000:
    BEEP .5,29
2020 LET y=year+(month=12): LET m=month+1-12*(month=12): LET
    draw=14: BRIGHT 0: GO SUB 3000: BEEP .5,30
2030 INK 2: PLOT 38,0: DRAW 0,175: PLOT 39,0: DRAW 0,175: PLOT 64,0:
    DRAW 0,175: PLOT 65,0: DRAW 0,175: INK 7: INPUT AT 0,0: INK
    6;"f=Forwards b=Backwards r=Return Press a key then press
    enter";a$: BEEP .5,30
2040 IF a$="f" THEN LET year=year+(month=12): LET month=month+1
    -12*(month=12)
2050 IF a$="b" THEN LET year=year-(month=1): LET month=month-1
    +12*(month=1)
2060 IF a$="r" THEN RETURN
2070 GO TO 2000
3000 PRINT AT draw,0: INK 4;y: PRINT AT draw+3,0: INK 4;m$(m,1 TO
    3) " " "----"
3010 FOR n=1 TO 7: PRINT AT draw,4*n+1: INK 5;d$(n): NEXT n
3020 LET row=1: LET day=FN a(y,m): LET day=day-7*INT(day/7)
3030 FOR n=1 TO L(m)+((FN b(y)) AND (m=2))
3040 PRINT AT row+draw,1+4*(day+1);n: LET day=day+1
3050 IF day=7 THEN LET day=0: LET row=row+1
3060 NEXT n
3070 RETURN

```

6

Maze

This program generates a square maze of any size up to 50x50. You are placed in the centre and your position is marked by a red spot. You can move up, down, left or right. Only a 9x9 portion of the maze is displayed at any one time, but a grid reference is given down one side and along the top to indicate where you are in the maze. When you move out of the section on display a new section is drawn with your new position at the centre. Not all mazes have solutions, so beware, you may be permanently trapped.

Note: The letter A in quotes in lines 1040 and 1120 is graphics A.

```

1000 GO SUB 4000
1010 DIM x$(n+1,n+1): DIM y$(n+1,n+1)
1020 RANDOMIZE: FOR m=1 TO n+1: FOR p=1 TO n+1: LET x$(m,p)=
    CHR$(RND<.5): LET y$(m,p)=CHR$(RND>.5): NEXT p: NEXT m
1030 PRINT " " "Press any key": PAUSE 0: LET cx=INT((n+1)/2): LET
    cy=cx: LET cx1=cx: LET cy1=cy
1040 GO SUB 3000: GO SUB 2000: PRINT AT 12,9: INK 2;"A"
1050 LET a$=INKEY$: IF a$<"5" OR a$>"9" THEN GO TO 1050
1060 BEEP .4,30: IF a$="9" THEN RUN
1070 IF (a$="5") AND CODE y$(cx,cy) THEN LET cx1=cx-1
1080 IF (a$="6") AND CODE x$(cx,cy) THEN LET cy1=cy-1
1090 IF (a$="7") AND CODE x$(cx,cy+1) THEN LET cy1=cy+1
1100 IF (a$="8") AND CODE y$(cx+1,cy) THEN LET cx1=cx+1
1110 IF (cx1<x) OR (cx1>(x+8)) OR (cy1<y) OR (cy1>(y+8)) THEN LET
    cx=cx1: LET cy=cy1: GO SUB 2000
1120 PRINT AT 20-2*(cy-y),2*(cx-x)+1;"▲"; AT 20-2*(cy1-y),2*
    (cx1-x)+1: INK 2;"A"
1130 LET cx=cx1: LET cy=cy1: IF (cx>n) OR (cx<1) OR (cy>n) OR (cy<1)
    THEN PAUSE 30: INPUT AT 0,0: INK 6: PAPER 2: FLASH 1;"**▲▲You
    have escaped▲▲" " "Press enter for another game": LINE a$: RUN
1140 GO TO 1050
2000 LET x=cx-4: LET y=cy-4: LET a1=((1-x)>0)*(1-x): LET a2=
    ((n-x-8)>0)*9+((n-x-8)<0)*(n-x+1): LET b1=((1-y)>0)*(1-y):
    LET b2=((n-y-8)>0)*9+((n-y-8)<0)*(n-y+1)
2005 BORDER 0: PAPER 0: CLS: INK 4: BRIGHT 1: FOR a=a1 TO a2: FOR
    b=b1 TO b2-1: PRINT AT 20-2*b,20: INK 5;y+b: FOR c=0 TO 2:
    PLOT 16*a+c,16*b: DRAW 0,5: DRAW INVERSE CODE y$(x+a,y+b):
    0,7: DRAW 0,6: NEXT c: NEXT b: NEXT a
2010 FOR a=a1 TO a2-1: PRINT AT 2-(a-2*INT(a/2)),2*a: INK 5;x+a: FOR
    b=b1 TO b2: FOR c=0 TO 2: PLOT 16*a,16*b+c: DRAW 5,0: DRAW
    INVERSE CODE x$(x+a,y+b): 7,0: DRAW 6,0: NEXT c: NEXT b:
    NEXT a
2020 PRINT AT 0,0: INK 6;"Use the cursor keys 5-8 to move "; AT 5,23;

```

N.B. ▲ denotes space, see introduction.

7

```

4010 PRINT AT 5,4; INK 5;"This program will generate a random square maze
and place you in the centre of it" INK 4;"▲▲▲▲A 9x9 section is
displayed▲in which you can move left,▲▲▲▲right, up or down.▲A new
▲▲▲▲section will be displayed when▲you move off the grid"
4020 PRINT " INK 5;"You may select the size of the▲maze by specifying the
length of one side": INPUT AT 0,0;"Type in a number 1 to 50▲▲▲▲
▲▲▲▲followed by enter.▲▲"; FLASH 1;"SIZE▲"; LINE a$
4030 IF a$<>" " THEN FOR n=1 TO LEN a$: IF (a$(n)>="0") AND (a$(n)<=
"9") THEN NEXT n: LET n=VAL a$: IF n<=50 THEN GO TO 4050
4040 GO TO 4000
4050 CLS: PRINT "▲▲▲Your Spectrum computer will▲now take about▲"; INT
(n ↑ 2/16)+4;"▲secs" "to generate a▲"; n;"▲by▲"; n;"▲maze":
RETURN

```

Lunar Landing

This program presents you with the task of safely landing a lunar module on the moon's surface. You have two variables that you can control independently, the magnitude and direction of the jet thrust. You can set the magnitude from 0 to 9 by holding down the keys 0 to 9 and you can aim the module in one of three directions, left, vertical or right. The angle can be rotated anticlockwise by pressing Q and clockwise with P. The keys don't react immediately and so you may have to hold them down. The keyboard is scanned about once every second and to indicate that a key has been recognised a yellow square lights up in the top left of the screen.

The vertical velocity is positive upwards, the height is positive above the moon's surface. The horizontal position is positive when the space vehicle is to the right of its target, and the horizontal velocity is positive when the vehicle is moving from left to right with respect to the lunar surface. Therefore, to make the vehicle move to the left, a negative horizontal velocity is needed, and this may be generated by turning the craft anticlockwise until it is pointing left.

A safe landing is defined as landing with a velocity of less than 10. You must, however, also try to land in the right place, that is reduce your horizontal position to between +10 and -10. Your horizontal velocity must also be in the same range. Be careful that your fuel doesn't run out, or you will fall freely and land with a crash.

```

100 GO SUB 4000: GO SUB 5000: INPUT "Press enter to play▲"; LINE z$:
GO SUB 2000: GO SUB 1000: GO SUB 3000
110 LET z$=" "; PAUSE 1: LET z$=INKEY$
115 PRINT AT 2,0;"▲": IF z$<>" " THEN PRINT AT 2,0; PAPER 6;"▲"
120 IF (z$="p") AND (d<3) THEN LET d=d+1
130 IF (z$="q") AND (d>1) THEN LET d=d-1
140 IF (z$="0") AND (z$<="9") THEN LET z=VAL z$: LET j=z
145 LET j=j*(f>0)
150 LET v=INT (.5+v+ (.7+.3*(d=2))*j-5+(ABS hv ↑ 2)/5000)
160 LET h=INT (h+v+.5): LET h=h*(h>0)
170 LET hv=INT (.5+hv+.7*j*((d=3)-(d=1)))
180 LET hp=((5*hp+hv)/5)
190 LET f=(f>0)*INT (.5+f-j)
200 GO SUB 3000
210 IF f=0 THEN PRINT AT 1,16; PAPER 2; FLASH 1;"▲NO FUEL▲"
220 IF (h<10) AND (ABS v<10) AND (ABS hp<10) AND (ABS hv<10) THEN
PRINT AT 1,0; PAPER 4; INK 0; FLASH 1;"▲SAFE LANDING▲": INPUT
"Press enter to play again"; LINE z$: RUN
230 IF (h<10) AND (ABS v<10) AND (ABS hv<10) THEN PRINT AT 1,0; INK
0; PAPER 4; FLASH 1;"▲SAFE LANDING▲", PAPER 2; FLASH 1;
"▲WRONG PLACE▲": INPUT "Press enter to play again"; LINE z$: RUN

```

N.B. ▲ denotes space, see introduction.

```

240 IF h<10 THEN PRINT AT 1,0: PAPER 2: FLASH 1:"▲CRASH
LANDING▲": INPUT "Press enter to play again": LINE z$: RUN
250 GO TO 110
1000 BORDER 0: INK 7: PAPER 0: CLS
1010 PRINT PAPER 1:"▲▲▲▲▲LUNAR LANDING▲▲▲▲▲"
1020 PAPER 7: INK 2: PRINT "▲▲▲▲▲JETS:"
1030 PRINT "▲▲▲▲▲DIRECTION:"
1040 INK 1: PRINT "▲▲▲VERTICAL VEL.:"
1050 PRINT "▲▲▲▲▲HEIGHT:"
1060 PRINT "▲HORIZONTAL VEL.:"
1070 PRINT "▲HORIZONTAL POS.:"
1080 PRINT "▲▲▲▲▲FUEL:"
1090 PAPER 0: INK 7: PRINT PAPER 4: AT 19,0:"▲▲▲▲▲
▲▲▲▲▲"
1100 RETURN
2000 DATA 0,2,10000,0,INT(1000*RND),0,2000,"LEFT▲▲▲","VERTICAL",
"RIGHT▲▲"
2010 RESTORE 2000: DIM d$(3,6): READ j,d,h,v,hp,hv,f,d$(1),d$(2),d$(3)
2020 LET b$="▲▲▲▲▲": LET c$=b$+b$: LET b=1: RETURN
3000 DIM r$(8): DATA STR$ j,d$(d),STR$ v,STR$ h,STR$ hv,STR$ hp,
STR$ f
3010 RESTORE 3000: FOR n=1 TO 7: READ r$: PRINT AT 1+2*n,16: INK 7:
BRIGHT 1: PAPER 2-(n>2): FLASH ((n=3)*(ABS v>200)+(n=4)*
(h<500)+(n=5)*(ABS hv>100)+(n=6)*(ABS hp>1000)+(n=7)*
(f<100)):r$: NEXT n
3020 LET b=INT(hp/10+.5): LET b=b-32*INT((b-1)/32): PRINT AT 18,0:
INK 4:c$(b TO b+31)
3030 LET h1=INT(16-h/1000): LET h1=h1+(h1<1)*(1-h1): PRINT AT
h1,27:"▲▲": AT h1+3,27:"▲▲": AT h1+1,27: INK 5:a$(d,1): AT
h1+2,27:a$(d,2)
3040 FOR m=0 TO 1: FOR n=0 TO 8 STEP 4: FOR o=0 TO 1: PLOT
INVERSE m: INK 2: BRIGHT 1:231+o+n-6*d,151-h1*8: DRAW INK
2: INVERSE m:2*j*((d=1)-(d=3))-2*j: NEXT n: NEXT m
3050 RETURN
4000 BORDER 0: INK 7: PAPER 0: CLS :
4010 BRIGHT 1: PRINT PAPER 1:"▲▲▲▲▲LUNAR▲LANDING▲▲▲▲▲
▲▲▲▲"
4020 PRINT "▲▲▲▲Your task is to land the▲▲▲lunar module safely on
the▲▲▲moon's surface at the correct▲▲place."
4030 PRINT "▲▲▲You can alter direction▲▲(LEFT RIGHT OR VERTICAL)
by▲▲pressing keys Q(anticlockwise)▲▲and▲P(clockwise).▲"
4040 PRINT "▲▲You can set the jet thrust by pressing keys 1 to 9."

```

```

4050 PRINT "▲▲▲The vertical and horizontal speeds at touchdown must be
less than 10 for safety."
4060 PRINT "▲▲▲The correct landing place▲▲▲has a horizontal
position▲▲▲between +10 and -10."
4070 PRINT "▲▲▲Keys don't react immediately they are scanned every
second.▲▲Hold them down until a yellow▲▲square appears in the top
left▲of the screen."
4080 RETURN
5000 DATA 15,127,123,119,239,223,255,255
5010 DATA 0,128,192,192,192,192,192,128
5020 DATA 127,62,0,0,0,0,0
5030 DATA 255,255,198,196,192,240,224,192
5040 DATA 3,31,56,63,63,63,31,7
5050 DATA 192,248,60,252,252,252,248,224
5060 DATA 3,7,14,28,56,124,0,0
5070 DATA 192,224,112,56,28,62,0,0
5080 DATA 0,1,3,3,3,3,1
5090 DATA 240,254,222,238,247,251,255,255
5100 DATA 255,255,99,35,3,15,7,3
5110 DATA 254,124,0,0,0,0,0,0
5120 DIM a$(3,2,2): FOR a=0 TO 11: RESTORE 5000+10*a: FOR n=USR
CHR$(97+a) TO USR CHR$(97+a)+7: READ b: POKE n,b: NEXT n:
NEXT a
5130 FOR a=1 TO 3: FOR b=1 TO 2: FOR c=1 TO 2: LET a$(a,b,c)=CHR$(
137+4*a+2*b+c): NEXT c: NEXT b: NEXT a
5140 RETURN

```

Android Nim

Nim is a game usually played with matchsticks, here it is played with colourful androids. The rules and object of the game are simple. The table is set with six rows, each containing a random number of androids.

You take it in turns with the Spectrum to remove any number of androids from one of the rows. You must remove at least one android each time and you cannot remove them from more than one row in any move.

The winner is the person to take the last android. This game is not rigged in favour of the computer, although the person to start often has an advantage.

Note: The letters A through to N in quotes in line 5150 are graphics letters.

```

100 GO SUB 2000: GO SUB 5000: PRINT "Press any key to play": PAUSE 0
110 GO SUB 6000
120 IF RND <= 5 THEN PRINT AT 21,0; PAPER 2; FLASH 1;"YOU TO
    START~~~~~": PAUSE 100: GO TO 190
130 PRINT AT 21,0; PAPER 1; FLASH 1;"SPECTRUM TO START~~~~~
    ~~~~~": PAUSE 100
140 PRINT AT 21,0; PAPER 1; FLASH 1;"SPECTRUM'S MOVE~~~~~
    ~~~~~": LET d=1: LET e=a(d): GO SUB 7000: LET a=a+INT
    (6*RND): IF NOT c AND a(e) THEN LET b=1: GO TO 180
150 IF NOT c THEN GO TO 140
160 FOR d=1 TO 6: FOR e=a(d)-1 TO 0 STEP -1: GO SUB 7000: IF c
    THEN NEXT e: NEXT d
170 LET a=d: LET b=a(a)-e
180 GO SUB 8000: GO SUB 9000: LET a=0: FOR n=1 TO 6: LET a=a+(n):
    NEXT n: IF a=0 THEN PRINT AT 21,0; PAPER 1; FLASH 1;
    "SPECTRUM IS THE WINNER~~~~~": INPUT "Press enter to play
    again~": LINE z$: GO TO 110
190 PRINT AT 21,0; PAPER 2; FLASH 1;"YOUR MOVE~~~~~
    ~~~~~": INPUT AT 0,0; INK 2; PAPER 7;"Which row will you
    attack? Type a number followed by enter,▲": LINE z$: IF z$<>" "
    THEN IF (LEN z$=1) AND (z$<"0") AND (z$<="6") THEN LET a=VAL
    z$: IF a(a)>0 THEN GO SUB 8000: GO TO 210
200 GO TO 190
210 INPUT AT 0,0; PAPER 2;"How many will you remove,▲? Type a number
    followed by enter.~": LINE z$: IF z$<>" " THEN IF (LEN z$=1) AND
    (z$>"0") AND (z$<="7") THEN LET b=VAL z$: IF b<=a(a) THEN GO
    TO 230
220 GO TO 210
230 GO SUB 9000: LET a=0: FOR n=1 TO 6: LET a=a+(n): NEXT n: IF
    a=0 THEN PRINT AT 21,0; PAPER 2; FLASH 1;"YOU WIN~~~~~
    ~~~~~": INPUT AT 0,0;"Press enter to play again~":
    LINE z$: GO TO 110

```

N.B. \blacktriangle denotes space, see introduction.

```

240 GO TO 140
2000 REM explanation page
2010 INK 7: PAPER 0: BORDER 0: CLS: PRINT PAPER 2;"
      ~~~~~
      ~~~~~ANDROID NIM~~~~~
      ~~~~~" " PAPER 1;" ~~~~Nim
is a game for two, your ~~~~~
partner is the spectrum.~~~~~"
2015 PRINT " PAPER 1;" ~~~~You take turns at
      removing ~~~~~any number of
      androids from one ~~~~~of six
      rows~~~~~"
2020 PRINT " PAPER 6; INK 0;" ~~~~The Winner removes the last~~~~~
      ~~~~~Android~~~~~
      ~~~~~"
2030 PRINT "": RETURN

5000 DATA 7,31,21,17,17,15,7,1
5010 DATA 7,31,17,25,17,15,7,1
5020 DATA 7,31,17,19,17,15,7,1
5030 DATA 7,31,17,17,21,15,7,1
5040 DATA 224,248,168,136,136,240,224,128
5050 DATA 224,248,136,200,136,240,224,128
5060 DATA 224,248,136,152,136,240,224,128
5070 DATA 224,248,136,136,168,240,224,128
5080 DATA 31,19,31,3,3,1,1,3
5090 DATA 15,11,27,19,19,2,6,12
5100 DATA 255,131,3,3,31,16,48,0
5110 DATA 248,200,248,192,192,128,128,192
5120 DATA 240,208,216,200,200,64,96,48
5130 DATA 255,193,192,192,248,8,12,0
5140 FOR n=0 TO 13: RESTORE 5000+10*n: FOR m=USR CHR$(97+n) TO
      USR CHR$(97+n)+7: READ o: POKE m,o: NEXT m: NEXT n
5150 DIM a$(4,2): DIM b$(3,2): LET a$(1)="AE": LET a$(2)="BF": LET
      a$(3)="CG": LET a$(4)="DH": LET b$(1)="IL": LET b$(2)="JM": LET
      b$(3)="KN"
5160 RETURN

6000 REM sub to set up board
6010 CLS: PRINT AT 0,0; PAPER 1;"~~~~~ANDROID NIM~~~~~
      ~~~~~": DIM a(6): FOR n=1 TO 6: LET a(n)=1+INT(RND*7): FOR
      m=1 TO a(n): PRINT AT 3*n,3*m; INK 4+INT(RND*3);a$(1); AT
      3*n+1,3*m; INK 3;b$(1); AT 3*n,0; INK 7;n: NEXT m: NEXT n
6020 RETURN

7000 REM sub to determine safe or unsafe (c=0)
7010 DIM b(6,3)
7020 FOR n=1 TO 6: LET a=a(n): IF n=d THEN LET a=e

```

```

7030 LET b(n,3)=INT (a/4): LET a=a-4*INT (a/4): LET b(n,2)=INT (a/2): LET
a=a-2*INT (a/2): LET b(n,1)=INT a: NEXT n
7040 DIM c(3): FOR n=1 TO 3: LET c(n)=0: FOR m=1 TO 6:
LET c(n)=(c(n)<>b(m,n)): NEXT m: NEXT n
7050 LET c=c(1)+c(2)+c(3): RETURN
8000 REM sub to point androids in right direction
8010 PRINT AT 3*a,0: FLASH 1; a
8020 FOR n=1 TO a-1: FOR m=1 TO a(n): LET s=INT (3*RND): FOR p=2
TO 1 STEP -1: FOR q=3 TO 4: PRINT AT 3*n,3*m: INK 4+s;a$(q): AT
3*n+1,3*m: INK 3;b$(p): BEEP .02,2*(q+m+n+p): NEXT q: NEXT p:
NEXT m: NEXT n
8030 FOR m=1 TO a(a): LET s=INT (RND*3): FOR p=1 TO 3: FOR q=2 TO
3: PRINT AT 3*a,3*m: INK 4+s; a$(q): AT 3*a+1,3*m: INK 3;b$(p):
BEEP .02,10+2*(q+p+m): NEXT q: NEXT p: NEXT m
8040 FOR n=a+1 TO 6: FOR m=1 TO a(n): LET s=INT (RND*3): FOR p=1
TO 2: FOR q=2 TO 1 STEP -1: PRINT AT 3*n,3*m: INK 4+s;a$(q): AT
3*n+1,3*m: INK 3;b$(p): BEEP .02,20+2*(q+p+m+n): NEXT q: NEXT
p: NEXT m: NEXT n
8050 RETURN
9000 REM sub to remove androids a=row, b=no. to be removed
9010 FOR n=a(a) TO a(a)-b+1 STEP -1: FOR p=2 TO 3: FOR q=1 TO 3:
FOR r=1 TO 4: PRINT AT 3*a,3*n: INK 7;a$(r): AT 3*a+1,3*n: INK
2;b$(q): BEEP .1-.03*p,40+r+p+q: NEXT r: NEXT q: NEXT p
9020 PRINT AT 3*a,3*n: INK 5: FLASH 1;a$(4): AT 3*a+1,3*n: INK 2:
FLASH 1;60: BEEP 1,60: PRINT AT 3*a,3*n;"▲": AT
3*a+1,3*n;"▲": NEXT n
9030 LET a(a)=a(a)-b
9040 PRINT AT 3*a,0;a: RETURN

```

Morse

This program is a complete morse code training kit. It can transmit morse at any speed for you to translate into English and it can listen to your morse and check if it is correct.

Morse Code is a system of signals in which dots and dashes are combined to represent letters of the alphabet. It was invented in 1838 by Samuel Morse of the United States for use in electric telegraphy where it continued to be used until 1920. It is still used today in radiotelegraphy.

The basic code element is the dot or unit pulse. The time duration of a dot and space is that of two unit pulses. A dash is three unit pulses long; the space between letters is three unit spaces; the space between words is seven unit spaces. A speed of one baud is one pulse per second. The code speed in words per minute (w.p.m.) is defined as the number of dots/min divided by 25.

This program understands the following letters, numbers and symbols.

a . -	1 . - - -
b - . . .	2 . - - -
c - . - .	3 . - - -
d . - .	4 . - - -
e .	5
f . . .	6 - . . .
g - -	7 - . . .
h . . .	8 - . . .
i . .	9 - - - .
j - - -	0 - - - -
k - . -	
l . - .	Period . - - - -
m - -	Comma - - - - -
n -	Colon - - - - -
o - - -	Query . - - - -
p - . -	Apostrophe . - - - -
q - - -	Hyphen . - - - -
r . .	Fraction bar . - - .
s . .	Parenthesis - - - - -
t -	Quotation marks . - - . .
u . .	
v . . .	
w . - -	
x - . -	
y - - -	
z - - .	

The program has a small vocabulary of common words to throw at you and it

N.B. ▲ denotes space, see introduction.

can also string them together into sentences. These are not always meaningful but usually grammatically correct.

```

90 GO SUB 1000
100 GO SUB 6000: GO SUB 6000+1000*x: GO TO 100
1000 DATA "a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"
1010 DATA "1","2","3","4","5","6","7","8","9","0","?","/","("
1020 DIM m$(60,7)
1030 RESTORE 1000: FOR n=1 TO 45: READ a$: LET a=CODE a$(1): LET a=a-32*(a>90): LET m$(a-30)=CHR$(LEN a$-1)+a$(2 TO): NEXT n
1040 RETURN
1500 LET b$=""
1510 FOR e=1 TO LEN e$: LET e1=CODE e$(e)-30-32*(e$(e)>="a"): LET b$=b$+m$(e1,2 TO)+CODE m$(e1,1))+ "▲": NEXT e
1520 RETURN
2000 LET sp=1.4/s: FOR b=1 TO LEN b$: LET c$=b$(b): PAUSE 50*sp: BEEP sp*((c$=".")+3*(c$="-")),40: NEXT b
2010 RETURN
3000 DATA 20,"the","one","this","my","your","his","her","our","their","that","another","someone's","anyone's","david's","mary's","everybody's","the king's","last year's","the one and only","the latest"
3010 DATA 30,"red","orange","yellow","green","blue","violet","black","white","big","small","heavy","light","bright","dark","round","square","friendly","good","bad","fast","slow","crazy","lazy","thick","wet","dry","hot","warm","cold","honest"
3020 DATA 30,"man","woman","boy","girl","child","cat","dog","horse","cow","plant","field","factory","house","car","bike","lorry","train","aeroplane","book","pen","inkpot","pencil","chair","table","carpet","computer","television","fish","bird","worm"
3030 DATA 10,"hit","dropped","spoke to","ran to","played with","jumped over","sang with","got married to","killed","shot at"
3040 DATA 10,"quickly","slowly","merrily","happily","sadly","enthusiastically","with all the grace of a queen","drunkenly","repeatedly","with great gusto"
3050 RESTORE 3000: READ e: FOR n=1 TO 1+RND*e: READ f$: NEXT n: LET e$=f$+"▲"
3060 FOR m=1 TO RND*3: RESTORE 3010: READ e: FOR n=1 TO 1+RND*e: READ f$: NEXT n: LET e$=e$+f$+"▲": NEXT m

```

```

3070 RESTORE 3020: READ e: FOR n=1 TO 1+RND*e: READ f$: NEXT n: LET e$=e$+f$+"▲"
3080 RESTORE 3030: READ e: FOR n=1 TO 1+RND*e: READ f$: NEXT n: LET e$=e$+f$+"▲"
3090 RESTORE 3000: READ e: FOR n=1 TO 1+RND*e: READ f$: NEXT n: LET e$=e$+f$+"▲"
3100 FOR m=1 TO RND*3: RESTORE 3010: READ e: FOR n=1 TO 1+RND*e: READ f$: NEXT n: LET e$=e$+f$+"▲": NEXT m
3110 RESTORE 3020: READ e: FOR n=1 TO 1+RND*e: READ f$: NEXT n: LET e$=e$+f$+"▲"
3120 RESTORE 3040: READ e: FOR n=1 TO 1+RND*e: READ f$: NEXT n: LET e$=e$+f$+"▲"
3130 RETURN
4000 RESTORE 3000+10*INT(5*RND): READ e: FOR n=1 TO 1+RND*e: READ e$: NEXT n: RETURN
5000 DATA "0123456789abcdefghijklmnopqrstuvwxyz.-?/('\"
5010 RESTORE 5000: READ e$: LET e$=e$(1+INT(RND*LEN e$))
5020 RETURN
6000 BORDER 0: PAPER 0: INK 7: CLS: BRIGHT 1: PRINT PAPER 1: "▲▲▲▲MORSE TRAINER▲▲▲▲"
6005 BEEP 5,30
6010 PRINT "INK 2: PAPER 7:"▲▲▲YOU CAN BE TESTED ON THREE SUBJECTS▲▲▲▲▲"
6020 PAPER 2: PRINT "'1: LETTERS NUMBERS AND SYMBOLS▲"
6030 PRINT "'2: WORDS AND SHORT PHRASES▲"
6040 PRINT "'3: COMPLETE SENTENCES▲"
6045 PAPER 0: BEEP 5,30
6050 INPUT AT 0,0: PAPER 2:"TYPE THE NUMBER OF YOUR CHOICE FOLLOWED BY ENTER▲": LINE Z$
6060 IF Z$>"0" AND Z$<"4" THEN LET Z=VAL Z$(1): GO TO 6080
6070 GO TO 6050
6080 PRINT AT 5+2*Z,0: FLASH 1: INK 7: PAPER 1:Z;"":
6085 BEEP 5,30
6090 PRINT AT 13,0: INK 2: PAPER 7:"▲▲▲YOU CAN BE TESTED IN TWO WAYS"
6100 PRINT "PAPER 2:"'1: TRANSLATE MORSE INTO ENGLISH▲"
6105 BEEP 5,30
6110 INPUT AT 0,0: PAPER 2:"TYPE THE NUMBER OF YOUR CHOICE FOLLOWED BY ENTER▲": LINE X$
6120 IF X$>"0" AND X$<"3" THEN LET X=VAL X$(1): GO TO 6140
6130 GO TO 6110
6140 PRINT AT 14+2*X,0: FLASH 1: INK 7: PAPER 1:X;"":
6150 BEEP 5,30
6160 RETURN

```

```

6500 LET n=1
6510 IF d$(n)="." THEN LET n=n+1: GO TO 6510
6520 IF LEN d$<n+32 THEN PRINT d$(n TO): RETURN
6530 IF d$(n+32)="." THEN PRINT d$(n TO n+31): LET n=n+32:
GO TO 6510
6540 FOR m=31 TO 1 STEP -1: IF d$(n+m)="." THEN PRINT d$(n TO
n+m): LET n=n+m: GO TO 6510
6550 NEXT m: PRINT d$(n TO n+31): LET n=n+32: GO TO 6510
7000 CLS: PRINT PAPER 1: "TRANSLATE MORSE INTO ENGLISH."
7100 INPUT PAPER 2: "SELECT A CODE SPEED (W.P.M.):"; LINE Z$: IF
Z$<>" " THEN FOR N=1 TO LEN Z$: IF Z$(N)>="0" AND Z$(N)<=
"9" THEN LET S=VAL Z$: GO TO 7120
7110 GO TO 7100
7120 PRINT "PAPER 1: WRITE THE ENGLISH ON PAPER AS IT IS
TRANSMITTED IN MORSE."; GO SUB 6000-1000*Z: GO
SUB 1500: INPUT PAPER 2: "PRESS ENTER TO START"; LINE Z$
7130 GO SUB 2000: INPUT AT 0,0: PAPER 2: "DO YOU WISH TO SEE THE
MORSE PRINTED? (Y=YES)"; LINE Z$: IF Z$="Y" OR Z$="y"
THEN LET d$=b$: INK 6: GO SUB 6500: INK 7
7140 INPUT AT 0,0: PAPER 2: "PRESS ENTER TO SEE THE CORRECT
ENGLISH TRANSLATION."; LINE Z$: PRINT: INK 4: LET D$=E$: GO
SUB 6500: INK 7
7150 INPUT AT 0,0: PAPER 2: "DO YOU WANT ANOTHER GO?";
(Y=YES)"; LINE Z$: IF Z$="Y" OR Z$="y" THEN CLS:
GO TO 7120
7160 RETURN
8000 CLS: PRINT PAPER 1: "TRANSLATE ENGLISH TO MORSE."
8010 GO SUB 6000-1000*Z: GO SUB 1500: PRINT PAPER 1: "1=DOT,
2=DASH"
8020 PRINT "PAPER 1: ENTER THE MORSE USING KEYS 1 & 2 PRESS KEY
5 WHEN YOU HAVE FINISHED. DO NOT PRESS
ENTER."
8030 INK 6: LET D$=E$: GO SUB 6500: INK 7: PRINT "PAPER 2: START
NOW": GO SUB 8500: GO SUB 8600
8040 IF G$=f$ THEN PRINT INK 5: "CORRECT": GO TO 8060
8050 PRINT INK 5: "WRONG: THIS IS THE CORRECT ANSWER"
8060 LET D$=B$: GO SUB 6500
8070 INPUT PAPER 2: "ANOTHER GO? (Y=YES)"; LINE Z$: IF Z$="Y"
OR Z$="y" THEN GO TO 8000
8080 RETURN
8500 LET f$=""
8505 IF INKEY$<>" " THEN GO TO 8500
8510 LET a$=INKEY$: IF a$=" " THEN GO TO 8510
8515 IF a$="1" THEN BEEP .1,30: LET f$=f$+"."
8520 IF a$="2" THEN BEEP .3,30: LET f$=f$+"-"

```

```

8530 IF a$="5" THEN BEEP .7,5: RETURN
8540 GO TO 8505
8600 LET G$=" ": FOR N=1 TO LEN B$: IF B$(N)="." OR B$(N)="-" THEN
LET G$=G$+B$(N)
8610 NEXT N
8620 RETURN

```

Maths

This program tests your powers of arithmetic. It presents series of questions on any of the four fundamental operations; addition, subtraction, multiplication and division. You may also select one of three levels of difficulty; introductory, intermediate and advanced. These levels control the size of the numbers that are used.

When you have made your choice, you are given 5 questions with a running score that is updated after each question. If you get an answer wrong, the correct answer is given.

It is not intended that you should be able to do all of these questions in your head, so have some writing material handy for the more difficult ones. When you have answered 5 questions you are given the chance to be set another 5, otherwise you may move onto another subject or level.

All of the answers are positive integers

```

1000 DATA "▲▲▲ADDITION", "▲▲SUBTRACTION", "▲MULTIPLICATION",
      "▲▲DIVISION": RESTORE 1000: DIM C$(4,16): FOR N=1 TO 4:
      READ C$(N): NEXT N
1010 BORDER 0: INK 7: BRIGHT 1: PAPER 0: CLS: RANDOMIZE
1020 PRINT AT 0,3: PAPER 1: "▲INTEGER ARITHMETIC TESTS▲"
1030 PRINT " PAPER 2: "THERE ARE 4 TYPES OF TEST" " PAPER 7: INK 2'
      "1:▲ADDITION▲" "2:▲SUBTRACTION▲" "3:▲MULTIPLICATION▲"
      "4:▲DIVISION▲"
1040 INPUT PAPER 2: "ENTER 1-4▲": LINE A$: IF A$>="1" AND
      A$<="4" THEN LET C1=VAL A$: PRINT AT 2+2*C1,0: PAPER 2:
      FLASH 1;C1: GO TO 1060
1050 GO TO 1040
1060 PRINT AT 12,0: PAPER 2: "THERE ARE 3 LEVELS" " INK 2: PAPER 7:
      "1:▲INTRODUCTORY" "2:▲INTERMEDIATE" "3:▲ADVANCED"
1070 INPUT PAPER 2: "ENTER 1-3▲": LINE A$: IF A$>="1" AND A$<=
      "3" THEN LET C2=VAL A$: GO TO 1090
1080 GO TO 1070
1090 GO SUB 2000
1100 GO TO 1000
1200 REM RANDOM INTEGER
1210 LET X$=STR$ INT (9*RND+1)
1220 LET Y$=STR$ INT (10*RND): IF Y$=X$ THEN GO TO 1220
1230 LET Z$=STR$ INT (10*RND): IF Z$=X$ OR Z$=Y$ THEN GO TO 1230
1240 LET X$=X$+Y$+Z$: LET X$=X$(1 TO C2): LET Z=VAL X$
1250 RETURN
2000 LET N0=5: LET N1=1: LET N2=0
2005 FOR M=0 TO 4

```

20

N.B. ▲ denotes space, see introduction.

```

2010 CLS: PRINT AT 0,7: PAPER 1;C$(C1)' PAPER 2: INK 6;N0:
      "▲QUESTIONS▲THIS IS NUMBER▲":N1' INK 7' "SCORE SO FAR▲":
      AT 4,13;N2: AT 4,16:"▲OUT OF▲": AT 4,24;N1-1
2020 GO SUB 1200: LET NN1=Z: GO SUB 1200: GO SUB 2190+10*C1
2030 INPUT "ENTER ANSWER▲": LINE Z$: IF Z$<>" " THEN FOR N=1 TO
      LEN Z$: IF Z$(N)>="0" AND Z$(N)<="9" THEN NEXT N: LET NN2=
      VAL Z$: PRINT NN2: GO TO 2045
2040 GO TO 2030
2050 IF NN2=NN3 THEN PRINT INK 0: PAPER 4: FLASH 1' "CORRECT":
      LET N2=N2+1: BEEP .5,30: GO TO 2070
2060 PRINT INK 0: PAPER 5: FLASH 1' "WRONG": PRINT "CORRECT
      ANSWER IS▲":NN3: BEEP .5,0
2070 PRINT PAPER 2: AT 4,13;N2: AT 4,24;N1: INPUT "PRESS ENTER FOR
      NEXT QUESTION▲": LINE Z$: LET N1=N1+1
2080 NEXT M
2090 INPUT AT 0,0: "ANOTHER 5 QUESTIONS▲?▲▲▲▲▲Y=▲YES
      N=▲NO▲": LINE Z$: IF Z$="Y" OR Z$="y" THEN LET N0=N0+5:
      GO TO 2005
2100 RETURN
2200 PRINT AT 8,0;NN1;"▲+▲":Z;"▲=▲": LET NN3=NN1+Z: RETURN
2210 PRINT AT 8,0;NN1+Z;"▲-▲":NN1;"▲=▲": LET NN3=Z: RETURN
2220 PRINT AT 8,0;NN1;"▲x▲":Z;"▲=▲": LET NN3=Z*NN1: RETURN
2230 PRINT AT 8,0;NN1*Z;"▲/▲":NN1;"▲=▲": LET NN3=Z: RETURN

```

21

Biorhythms

This program plots the cyclic variations of human vitality, emotion and intellect. According to biorhythm theory these vary in cycles of 23, 28 and 33 days respectively from birth.

The program is largely self-explanatory. When the cycles are positive (above the line) then our level of performance is better. Some people say that when all three cycles bottom out together then we are better off staying in bed. I'm sure you'll find that a useful excuse.

The program will ask for a date of birth and for a date from which to plot the biorhythms. You may select a two week, or two month plot. With the former, the program will plot from the weekend following the date specified; in the latter case it will commence at the start of the month.

```

100 GO SUB 5000: GO SUB 1000: GO TO 3000
1000 REM set up arrays
1010 DIM p$(185): FOR n=0 TO 184: LET p$(n+1)=CHRS ((16*SIN (PI*n/92))
+144): NEXT n
1020 DIM e$(225): FOR n=0 TO 224: LET e$(n+1)=CHRS ((16*SIN
(PI*n/112))+104): NEXT n
1030 DIM i$(265): FOR n=0 TO 264: LET i$(n+1)=CHRS ((16*SIN (PI*n/132))
+64): NEXT n
1040 DIM n(12): DATA 31,28,31,30,31,30,31,31,30,31,30,31:
RESTORE 1040: LET m=0: FOR n=1 TO 12: LET m(n)=m: READ m1:
LET n(n)=m1: LET m=m+m1: NEXT n
1050 DIM m$(12,9): DATA "january","february","march","april","may",
"june","july","august","september","october","november",
"december": FOR n=1 TO 12: READ m$(n): NEXT n
1060 DIM d$(7,2): DATA "su","mo","tu","we","th","fr","sa": FOR n=1 TO 7:
READ d$(n): NEXT n
1070 DEF FN a(y,m)=(m=2)*((y/4=INT (y/4))-(y/100=INT (y/100)))+(y/400=
INT (y/400))
1080 RETURN
1200 LET dn=365*y+INT (y/4)+INT (y/400)-INT (y/100)-(m<3)*((y/4=INT
(y/4))-(y/100=INT (y/100)))+(y/400=INT (y/400))+m(m)+d-1
1210 RETURN
1400 LET dne=1+ne-184*INT (ne/184): LET dne=1+ne-224*INT (ne/
224): LET dnei=1+ne-264*INT (ne/264): INK 6: PLOT n, CODE
p$(dne): DRAW 0,1: INK 4: PLOT n, CODE e$(dnei): DRAW 0,1:
INK 3: PLOT n, CODE i$(dnei): DRAW 0,1: INK 7
1410 RETURN
1600 OVER 1: PRINT AT 0,9: PAPER 1:"▲▲BIORHYTHMS▲▲": PAPER 0:
INK 6: AT 1,0:"PHYSICAL": INK 4: AT 6,0:"EMOTIONAL": INK 3: AT
11,0:"INTELLECTUAL"

```

```

1610 INK 6: PLOT 0,144: DRAW 254,0: INK 4: PLOT 0,104: DRAW 254,0:
INK 3: PLOT 0,64: DRAW 254,0: OVER 0: INK 7
1620 RETURN
2000 REM plot 2 week period
2010 LET y=ye: LET m=me: LET d=7*(we-1)+1: GO SUB 1200: REM
dn=0=sun
2020 LET dne=dn-7*INT (dn/7): LET de=7*(we-1)+1: IF dne<>0 THEN
LET dn=dn+7-dne: LET de=de+7-dne
2030 FOR n=0 TO 2: FOR d=0 TO 6*(n<2) STEP 2: PRINT AT 19,14*n+2*d:
PAPER 1+(d=0):d$(d+1): PRINT INK 5: AT 20,14*n+d*2;de+7*n
+d-n(m)*(de+7*n+d+1>n(m)): NEXT d: NEXT n
2040 INK 6: PRINT AT 21,0;ye: TAB 6;m$(me): IF de>n(me)-13 THEN
PRINT AT 21,22;m$(me+1-12*(me=12)): INK 7
2050 FOR n=0 TO 255: LET ne=8*(dn-dn1)+n/2: GO SUB 1400: NEXT n
2060 GO SUB 1600
2070 RETURN
2500 REM PLOT 2 MONTH PERIOD
2510 LET m=me: LET y=ye: LET d=1: GO SUB 1200
2520 LET dne=dn-7*INT (dn/7): IF dne<>0 THEN LET d=7-dne+1
2530 LET m1=n(me): LET m2=n(1+me-12*INT (me/12)): FOR n=0 TO 9:
LET l=28*n+4*d: IF (l<=254) THEN FOR m=0 TO 1: PLOT INK
2;l+m,24: DRAW INK 2;0,6: NEXT m: NEXT n
2540 FOR n=1 TO m1+m2: IF (n-d)/7=INT ((n-d)/7) THEN PRINT AT
19,n/2-1: PAPER 2;"su": AT 20,n/2-1: PAPER 0: INK 5;n-m1*
(n>m1)
2550 NEXT n
2560 INK 6: PRINT AT 21,0;ye: TAB 6;m$(me): TAB 20;m$(1+me-12*
(me=12)): INK 7
2570 FOR n=0 TO 255: LET ne=8*(dn-dn1)+2*n: GO SUB 1400: NEXT n
2580 GO SUB 1600
2590 RETURN
3000 REM choices
3010 CLS: PRINT AT 0,9: PAPER 1:"▲▲BIORHYTHMS▲▲"
3020 PRINT " INK 5;▲▲▲▲Biorhythms are calculated▲▲▲▲from the date of
birth"
3030 GO SUB 4000: LET dn1=dn: PRINT
3040 PRINT AT 0,9: PAPER 1:"▲▲BIORHYTHMS▲▲": PRINT INK 4: AT
8,0:"Biorhythms may be displayed for any date"
3050 GO SUB 4000
3060 LET ye=y: LET me=m: LET we=1+INT (d/7)-(d>=28)
3070 PRINT INK 6: " Biorhythms may be displayed for a choice of
period" "A=2 weeks" "B=2 months"
3080 INPUT INK 6:"Type in your choice of period▲▲A or B followed by
enter▲": LINE a$: IF a$="a" OR a$="A" THEN CLS: GO SUB 2000:
GO TO 3100

```

```

3090 CLS : GO SUB 2500
3100 INPUT "PRESS ENTER"; LINE a$: CLS : INPUT AT 0,0;"1=another
display date 2=another birth date followed by enter^"; LINE a$
3110 IF a$="1" THEN CLS : GO TO 3040
3120 CLS : GO TO 3000
4000 REM input y m d
4030 INPUT INK 5; AT 0,0;"Type in the year as 4 digits^ followed by
enter^"; LINE a$: IF a$<>" " THEN FOR n=1 TO 4: IF
a$(n)>="0" AND a$(n)<="9" THEN NEXT n: LET y=VAL a$: GO TO
4050
4040 GO TO 4030
4050 INPUT AT 0,0; INK 5;"Type the month as a number 1-12 followed by
enter^"; LINE a$: IF a$<>" " THEN FOR n=1 TO LEN a$: IF
a$(n)>="0" AND a$(n)<="9" THEN NEXT n: LET m=VAL a$: IF
m>=1 AND m<=12 THEN GO TO 4070
4060 GO TO 4050
4070 INPUT INK 5; AT 0,0;"Type the day of the month 1-";n(m)+FN a(y,m);
^ followed by enter^"; LINE a$
4080 IF a$<>" " THEN FOR n=1 TO LEN a$: IF a$(n)>="0" AND a$(n)<=
"9" THEN LET d=VAL a$: IF d>=1 AND d<=n(m)+FN a(y,m) THEN
GO TO 4100
4090 GO TO 4070
4100 GO SUB 1200: LET d1=dn-7*INT (dn/7): PRINT PAPER 2;d$(d1+1);
TAB 4;m$(m); TAB 18;d; TAB 24;y
4110 RETURN
5000 REM title page
5100 BORDER 0: INK 7: PAPER 0: BRIGHT 1: CLS
5110 PRINT AT 0,9; PAPER 1;"^BIORHYTHMS^"
5120 PRINT INK 5;"^Biorhythms are the cyclic^variations in our
performance^and abilities physically,^emotionally and
intellectually"
5130 PRINT INK 6;"^These cycles have periods of^23 days, 28 days and
33 days^respectively"
5140 PRINT INK 5;"^The cycles start at birth and^continue to death, and
our^abilities in the three fields^of human behaviour rise and fall
sinusoidally"
5150 RETURN

```

Telephone

This program is a fast data storage and retrieval mechanism. I think that its most popular application will be to store names, addresses and telephone numbers and so I have called it Telephone. The text in line 2010 also calls it "Telephone and Address pad" but there is no reason why you shouldn't change this if you wish to use it for other purposes. You could use this program to store details of collections (stamps etc) or as a memo notepad to remind you of things, you will probably think of other applications as I describe how it works.

The program clears a block of about 6K bytes of memory for data storage (that's enough for about 80 names, addresses and telephone numbers). Entries of any length can be made into this free memory, and are stored sequentially. Entries may contain any number of lines of text although if more than 22 lines are used, they will not all be visible on the screen at once. The text may contain any of the letters, numbers and symbols available from the keyboard, including keywords and colour change codes. It is even possible to have entries with sections picked out in flashing colours with different levels of brightness.

Finding an entry is very simple. All you need to do is enter an identifying key of up to 6 characters and a fast machine code search routine will find the first entry which contains that sequence of letters. You must remember that the computer distinguishes between upper and lower case and so you are advised to adopt some standard for their use when making entries. The identifying key must contain exactly the same characters as the entry. This method of finding entries allows you to remember and hence enter only the minimum amount of information and still find the entry. If however you can only remember that the entry had a letter E in it, then you will probably find that the computer presents you with a lot of entries before it finds the correct one. Here is an example of an entry with some valid and invalid identifying keys.

Mr & Mrs Upandcoming
Excellentcareerprospects View
Ontheright Rd
Milton Keynes
(1234) 56789

Valid identifying keys	Invalid keys
Mr	mr
&	Road
Upand	z
Milton	Rd.
ynes	
34)	
e	

N.B. ^ denotes space, see introduction.

When you first run the program, whether you have loaded it from cassette or typed it in, the screen will go blank for a few seconds and the report code 9 will appear, "STOP Statement". You must now delete lines 1000, 1010, 1020 and 1030 by typing in each of these numbers in turn followed by **ENTER**. You must next type **GO TO 2000** followed by **ENTER** and the program will be ready to go. You have three options.

1. New entry

Type in your details just as you would on a typewriter. To start a fresh line press **ENTER**. At the end of your last line press **ENTER** twice and your details will be stored away automatically.

2. Find change or delete an entry

First you must enter an identifying key. You will then be presented with all of the entries containing that key and you will be asked to indicate which one you are interested in. You can then choose to delete it, change it or leave it alone by entering a capital D, L or C. If you choose to delete it, it will simply be deleted and you will be asked to enter a new copy.

3. Save onto tape

Whenever you change the data, it is necessary to save the new version onto tape. You will, therefore, need at least one blank tape. After you have saved the program and variables, the computer will ask you to rewind the tape to play it back and verify the recording against the contents of memory.

The fast machine code routines are stored in hexadecimal form in a **DATA** statement at the start of the program. You are not expected to be able to understand how these work. Lines 1000 to 1030 contain, also, a short basic routine to clear enough memory for data storage and load the machine code into the bottom of that memory. After this has been done, lines 1000 to 1030 are redundant, and so they are removed to make more space.

One use of this program would be, as mentioned above, as a general notepad. You could make an entry such as "Tuesday 14th visit mum, collect dry cleaning, babysitting at Margarets". Then you might suddenly remember that you have left your clothes to be dry cleaned but cannot remember when they will be ready, so you enter "dry" as your key and up comes the entry, or you may have forgotten what you were talking to Margaret about over the weekend, so you enter "Marg" and up it comes. In the end if you forget where you have left your spectrum this program won't help very much.

```
1000 DATA "210268E5237E21586801A717C5C1EDB12808210000220068
E1C9D1D5C54F1A472B131ABE237920E510F7D1AF2BBE20FC220068
E1C92A0068AF23BE20FCE5EB21FF7FA7ED52E5C1E1ED5B0068ED
B0C9"
```

```
1010 CLEAR 26623: RESTORE 1000: READ A$: FOR N=0 TO 78: LET
B$=A$(N*2+1 TO N*2+2): LET B=16*(CODE B$(1)-48-7*(B$(1)>=
"A"))+CODE B$(2)-48-7*(B$(2)>="A"): POKE 26633+N,B: NEXT N
1020 LET AN=26712: POKE AN,0
1030 STOP
1210 IF AN+LEN A$>=32767 THEN LET B$="FULL": RETURN
1220 POKE AN,0: FOR N=1 TO LEN A$: POKE AN+N, CODE A$(N): NEXT N
1230 LET AN=AN+N: POKE AN,0
1240 RETURN
1410 IF LEN K$>6 THEN LET K$=K$(1 TO 6)
1420 POKE 26626, LEN K$: FOR N=1 TO LEN K$: POKE 26626+N, CODE
K$(N): NEXT N
1430 LET L=USR 26633: LET A1=PEEK 26624+256*PEEK 26625
1440 LET A$=" ": LET NA=1: IF A1=0 THEN RETURN
1450 LET A$=A$+CHR$ PEEK (A1+NA): LET NA=NA+1: IF PEEK (A1+NA)
THEN GO TO 1450
1460 RETURN
1600 REM DELETE
1610 LET L=USR 26686: LET AN=AN-NA
1620 RETURN
2000 INK 7: PAPER 0: BORDER 0
2010 CLS: PRINT AT 0,4: PAPER 1:"TELEPHONE & ADDRESS PAD"
2030 PRINT AT 3,0: PAPER 2:"1=NEW ENTRY""2=FINCH CHANGE OR
DELETE ENTRY""3=SAVE ONTO TAPE"
2040 INPUT "ENTER A NUMBER 1 TO 3": LINE Z$: IF Z$>"0" AND
Z$<"4" THEN LET Z=VAL Z$(1): GO SUB (2+Z)*1000
2050 GO TO 2010
3000 REM NEW
3010 CLS: PRINT AT 0,8: PAPER 1:"MAKE A NEW ENTRY"
3020 PRINT " PAPER 2:"ENTER YOUR DETAILS": LET A$=""
3030 INPUT LINE B$: IF B$<>" " THEN LET A$=A$+B$+CHR$ 13: PRINT
AT 5,0:A$: GO TO 3030
3040 GO SUB 1200
3050 IF B$="FULL" THEN INPUT AT 0,0: PAPER 2:"MEMORY FULL, ENTRY
NOT MADE:PRESS ENTER": LINE Z$: RETURN
3060 RETURN
4000 CLS: PRINT AT 0,0: PAPER 1:"FIND CHANGE OR DELETE AN ENTRY"
4010 INPUT AT 0,0:"ENTER AN IDENTIFYING KEY OF UP TO 6
CHARACTERS": LINE K$
4020 IF K$="" THEN GO TO 4010
4030 GO SUB 1400: IF A$<>" " THEN CLS: PRINT A$: INPUT AT 0,0:"IS
THIS YOUR ENTRY. Y=YES, N=NO, S=STOP SEARCHING":
LINE Z$: GO TO 4040
4035 INPUT: PAPER 2:"NOT FOUND PRESS ENTER": LINE Z$: RETURN
```

```

4040 IF Z$="N" OR Z$="n" THEN GO SUB 1600: GO SUB 1200: GO TO
4030
4045 IF Z$="S" OR Z$="s" THEN RETURN
4050 PRINT "PAPER 2;" "D=DELETE" "C=CHANGE" "L=LEAVE ALONE"
4060 INPUT "D, C OR L▲"; LINE Z$
4070 IF Z$="D" THEN GO SUB 1600: RETURN
4080 IF Z$="C" THEN GO SUB 1600: PRINT "PAPER 2;" "PRESS ENTER
AND ENTER NEW▲▲▲▲▲VERSION": INPUT LINE Z$: GO TO 3000
4090 IF Z$="L" THEN RETURN
4100 CLS: PRINT A$: GO TO 4050
5010 POKE 26624,an-256*INT (AN/256): POKE 26625, INT (AN/256): CLS:
PRINT PAPER 1;"REWIND TAPE": SAVE "TELEPHONE" LINE 5200
5020 CLS: SAVE "TEL"CODE 26624,6144
5030 CLS: PRINT PAPER 1;"REWIND TO VERIFY" "IF " "Tape loading
error" " OCCURS ENTER " "GO TO 2000" "
5050 VERIFY "TELEPHONE"
5060 VERIFY "TEL"CODE 26624,6144: RETURN
5200 CLEAR 26623: LOAD "TEL"CODE 26624: LET AN=PEEK
26624+256*PEEK 26625: GO TO 2000

```

Home Accounts

I wanted this program to be as simple and easy to use as possible but at the same time flexible enough to be able to cover a variety of accounting methods. I also wanted to be able to store enough accounting data in the 16K Spectrum to make the program worthwhile.

The program allows you one master account and several expenditure accounts. All sources of income are entered into the master account which also, automatically, contains the balance of each expenditure account. Each expenditure account contains details of expenses relevant to that account. This arrangement ties in with most people's income/expenditure patterns which are characterised by a small number of income sources and a wide variety of expenses which may be divided into categories. Income sources must always be entered into the master account, and might typically be a salary, interest on savings and the occasional gift. Expenses must be entered into the appropriate expenditure account. There is never any need to enter expenses as negative amounts (such entries will be ignored anyway) because the program realises that all entries in the expenditure accounts are to be debited from the master account.

The program is flexible in that it allows you to decide how many expenditure accounts you want, and what you want to call them. Careful thought should be given to this decision as, once it is made, it cannot be altered without destroying the details that you subsequently enter into these accounts. In practice, of course, giving yourself a lot of expenditure accounts doesn't allow you to spend more, as the latest balance in each one is entered into the master account as a debit. Perhaps less obvious, is the fact that the more expenditure accounts you have, the smaller the number of entries you can make in each. I suggest that you don't have more than 15 expenditure accounts and that you make one of these into a miscellaneous account. You are sure to find ways of spending money that don't fall neatly into the categories that you have given yourself.

Starting the program

After entering the program, either from this book or from tape, type **RUN** and **ENTER**.

Your first task is to decide upon your expenditure accounts and to give them names. To conserve memory you are only allowed a 6 character mnemonic for each name. You must also enter the date which will be taken as the starting date for your first accounting period. You will then be given a menu with four choices.

1. Make an Entry

You must select the account into which you wish to make an entry. Only sources of income must be entered into the master account. You must enter the date of the entry as a month and day number. I have assumed that you will

N.B. ▲ denotes space, see introduction.

make entries into each account in chronological order. If you don't, no harm is done, but the entries will appear in the order in which you enter them, they will not be sorted. The master account first contains the current balances in each expenditure account with the date given as the date of the last entry in that particular account, all income entries are displayed after that.

You must give a 6 character identifying mnemonic for each entry and you must enter the amount as a decimal in pounds, e.g. £14 is entered as 14.00, 36p is entered as 0.36 or .36.

You are given a chance to check the data you have entered and reject it, otherwise it is entered into the appropriate account whose current status is then displayed.

2. Display an account

Simply select which account you wish to display. You may get a printout too if you have a ZX Printer.

3. Save onto tape

Every time you make some entries, which might typically be 3 times a week, you must save the program and data onto tape. You will, therefore, need at least one, and preferably several, blank tapes. After saving, the computer will ask you to rewind the tape and play it back to verify the recording. You may wish to make several recordings, one after another, to ensure that if one gets damaged you have back-ups. An additional security method, which is standard practice in business circles, is to have a small number (say 3) tapes that you rotate. Call the tapes 1, 2 and 3, say. After day 1, save your program onto tape 1. On day 2 load the program from tape 1, make your new entries and save it onto tape 2. On day 3 load from tape 2 and save on tape 3 but on day 4 load from tape 3 and save onto tape 1 again. Continue rotating in this fashion. If one of your tapes gets destroyed you can always go back one day and re-enter the previous days entries, in fact, if you rotate n tapes, you can lose n-1 tapes and still all is not lost.

4. Start a new accounting period

The choice of accounting period is entirely yours but is limited by the number of entries that you can make in each account. Typically you may choose to start a new period every month.

Before starting a new period, you must save the current program and data onto tape. You may wish to build up a library of such tapes, one for each month. You will be asked to enter the starting date. You will still have the same number of expenditure accounts, with the same names as before, but all entries will have been removed. The only information carried over from one period to the next is the final balance in the old master account which is entered as "brought forward" in the new master account.

Finally, this program will have no time for halfpennies, but I will gladly supply a list of charities if you accumulate a bag full.

```

100 GO TO 1800
1000 REM display an account
1010 CLS : PRINT PAPER 1;C$(C);"▲Account",D1;"":M1;TAB 23;"to";
      TAB 27;D2;"":M2
1020 PRINT AT 1,0;PAPER 2;"DATE▲CATAGORY▲AMOUNT▲▲
      BALANCE"
1030 LET B=0:FOR N=1 TO N(C)
1040 PRINT CODE D$(C,N,1);"":CODE D$(C,N,2);
1050 PRINT TAB 6;I$(C,N);:IF A(C,N)=0 THEN LET A$="0:00":GO TO 1070
1060 LET A$=STR$ A(C,N):LET A$=A$(TO LEN A$-2)+"":A$(LEN A$-1
      TO)
1070 PRINT TAB 22-LEN A$;A$;:LET B=B+A(C,N):IF B=0 THEN LET
      B$="0:00":GO TO 1090
1080 LET B$=STR$ B:LET B$=B$(TO LEN B$-2)+"":B$(LEN B$-1 TO)
1090 PRINT TAB 32-LEN B$;B$
1100 IF (N+2)/22=INT ((N+2)/22) OR N=N(C) THEN INPUT "PRINTOUT
      REQUIRED?▲Y=YES▲":LINE Z$:IF Z$="Y" OR Z$="y" THEN
      COPY:CLS:BORDER 0
1110 NEXT n
1120 IF N(C)=0 THEN INPUT "PRESS ENTER▲":LINE Z$
1130 RETURN
1200 REM enter transaction
1210 GO SUB 3000:CLS:PRINT PAPER 1;"▲MAKE ENTRY IN▲":C$(C);
      "▲ACCOUNT▲":IF N(C)=EN THEN INPUT AT 0,0;"NO ROOM FOR
      MORE▲ENTRIES IN THIS CATEGORY, PRESS ENTER▲":LINE Z$:
      RETURN
1220 PRINT 'PAPER 2;"FIRST ENTER THE DATE"
1230 GO SUB 2000
1270 PRINT AT 2,23;D;"":M
1280 PRINT 'PAPER 2;"SECOND ENTER AN IDENTIFIER OF UP TO 6
      CHARACTERS"
1290 INPUT "ENTER IDENTIFIER▲":LINE H$:PRINT H$
1300 PRINT 'PAPER 2;"THIRD ENTER THE AMOUNT IN POUNDS AS A
      DECIMAL TO TWO DECIMAL▲▲PLACES, NO HALF-PENNIES"
1310 INPUT "ENTER AMOUNT▲":LINE Z$
1320 IF Z$<>" " THEN LET N1=0:FOR N=1 TO LEN Z$:LET N1=N1+N*
      (Z$(N)=".") :NEXT N:FOR N=1 TO LEN Z$:IF Z$(N)>="0" AND
      Z$(N)<="9" OR N=N1 THEN NEXT N:IF N1<>0 THEN GO TO 1340
1330 GO TO 1310
1340 LET A1=0:IF N1<>1 THEN LET A1=100*VAL (Z$(TO N1-1))
1350 IF N1<>LEN Z$ THEN LET A1=A1+VAL Z$(N1+1 TO LEN Z$)
1360 LET A1=INT A1:PRINT A1/100

```

```

1370 PRINT "PAPER 2; LASTLY CHECK THE DETAILS THAT YOU HAVE
ENTERED"
1380 INPUT "ARE THEY CORRECT? N=NO; LINE Z$: IF Z$="N" OR
Z$="n" THEN RETURN
1390 LET N(C)=N(C)+1
1400 LET D$(C,N(C))=CHR$(D+CHR$(M
1410 LET I$(C,N(C))=H$
1420 LET A(C,N(C))=(-1+2*(C=1))*A1
1430 LET B=(-1+2*(C=1))*A1: LET B(C)=B(C)+B: LET B(1)=B(1)+B
1440 IF C<>1 THEN LET A(1,C)=B(C): LET D$(1,C)=D$(C,N(C))
1450 IF (M>M2) OR ((M=1) AND (M2>6)) OR ((M=M2) AND (D>D2)) THEN
LET M2=M: LET D2=D
1460 GO SUB 1000
1470 RETURN
1600 CLS: PRINT PAPER 1; "ACCOUNTS FROM"; D1; "M1"; "TO";
D2; "M2;"
1610 PRINT "PAPER 2; 1: MAKE AN ENTRY"
1620 PRINT "PAPER 2; 2: DISPLAY AN ACCOUNT"
1630 PRINT "PAPER 2; 3: SAVE ONTO TAPE"
1640 PRINT "PAPER 2; 4: START A NEW ACCOUNTING PERIOD"
1650 INPUT "ENTER 1 TO 4; LINE Z$
1660 IF Z$="1" THEN GO SUB 1200: GO TO 1600
1670 IF Z$="2" THEN GO SUB 3000: GO SUB 1000: GO TO 1600
1680 IF Z$="3" THEN GO SUB 2100: GO TO 1600
1690 IF Z$="4" THEN GO SUB 2600: GO TO 1600
1700 GO TO 1600
1800 BRIGHT 1: PAPER 0: INK 7: BORDER 0: CLS: PRINT AT 0,10; PAPER
1; "ACCOUNTS"
1810 PRINT "HOW MANY NAMED EXPENDITURE ACCOUNTS DO
YOU WANT? INPUT "ENTER NUMBER"; CN: LET CN=CN+1
1820 LET EN=INT(320/CN): PRINT "YOU HAVE"; CN-1;
"ACCOUNTS" "WITH"; EN; "ENTRIES EACH"
1830 DIM C$(CN,6)
1840 GO SUB 2400
1850 PRINT "ENTER NAMES OF UP TO 6 CHARS FOR EACH
ACCOUNT"
1860 LET C$(1)="MASTER"
1870 FOR N=2 TO CN: INPUT ("ENTER NAME"; N-1); C$(N): NEXT N
1880 GO SUB 2000: LET M1=M: LET M2=M: LET D1=D: LET D2=D
1890 GO SUB 2800: GO TO 1600
2000 REM DATE
2010 INPUT "ENTER MONTH AS A NUMBER; LINE Z$: IF Z$<>" THEN
FOR N=1 TO LEN Z$: IF Z$(N)>="0" AND Z$(N)<="9" THEN NEXT
N: LET M=VAL Z$: GO TO 2030
2020 GO TO 2010

```

```

2030 INPUT "ENTER DAY AS A NUMBER; LINE Z$: IF Z$<>" THEN
FOR N=1 TO LEN Z$: IF Z$(N)>="0" AND Z$(N)<="9" THEN NEXT
N: LET D=VAL Z$: GO TO 2050
2040 GO TO 2030
2050 RETURN
2100 REM TAPE
2110 CLS: SAVE "ACCOUNTS" LINE 3100 PRINT "REWIND TAPE AND
PLAY TO VERIFY" "IF" "Tape loading error" "OCCURS"
ENTER RETURN"
2120 VERIFY "ACCOUNTS": RETURN
2400 REM INIT ARRAYS
2410 DIM D$(CN,EN,2): DIM I$(CN,EN,6): DIM A(CN,EN): DIM B(CN):
DIM N(CN)
2420 RETURN
2600 REM NEW ACCOUNT PERIOD
2610 LET B1=B(1): GO SUB 2400: CLS: GO SUB 2000: LET M1=M: LET
M2=M: LET D1=D: LET D2=D
2620 GO SUB 2800: LET A(1,1)=B1: LET B(1)=B1
2630 RETURN
2800 REM MAIN ACC INIT
2810 FOR N=1 TO CN: LET D$(1,N)=CHR$(D1+CHR$(M1): LET I$(1,N)=
C$(N): LET A(1,N)=0: LET N(N)=0: NEXT N
2820 LET I$(1,1)="BT/FWD": LET N(1)=CN
2830 RETURN
3000 CLS: PRINT PAPER 1; "CHOOSE ACCOUNT"
3010 FOR N=1 TO CN: PRINT N; TAB 6; C$(N): NEXT N
3020 INPUT "ENTER CHOICE NO. "; C
3030 RETURN
3100 BORDER 0: PAPER 0: INK 7: BRIGHT 1: CLS: GO TO 1600

```

Mastermind

This is a game for two people, your opponent being the Spectrum. The program contains a detailed explanation of how the game is played, but in case that isn't clear and you are not already familiar with it, I will give an explanation here as well.

One player thinks of a 4 digit number (leading zeros are allowed) and his opponent has to guess it in as few a number of goes as possible. Each guess is awarded two scores. The score called "correct" is the number of digits that correspond in value and position. The other score, called "good" is the number that corresponds in value only. Some examples are given.

Answer	7372	Correct	Good
Guess 1	1234	0	2
Guess 2	5678	1	0
Guess 3	2723	0	3
Guess 4	7372	4	0

You may choose whether the Spectrum or you has to think of a number, the opponent must guess it.

The algorithm for guessing a number is held mostly in machine code. This is for speed only and allows the computer to gain a psychological advantage over you by thinking of its next guess in a second or so rather than several minutes.

The machine code algorithm uses a 4 digit counter which starts at zero. Every time the computer has to make a new guess it increments the counter one place and compares the value with all of its previous guesses, generating a Correct and Good score for each of the comparisons. If the Correct and Good scores are the same as those given when the previous guesses were submitted to its opponent (you), then it concludes that the value of the counter could be the correct answer and presents that as its next guess. If the scores differ it increments the counter and tries again.

The machine code is stored in hexadecimal form in the DATA statements at the start of the program. Note that all letters here are entered in upper case, this is important. Check the contents of these DATA statements carefully before running the program, one small error will cause the computer to crash and lose your program. You may wish to save the program onto tape before running.

```

100 GO TO 2000
1000 DATA "3A707B32817B26003A817B6F29E529EB2114791911797B0104
00EDB0000000021A67AD11917D7B010200EDB021717B11757B010
400EDB0210000227F7B21757B11797B06041ABE200C3A807B3C3280
7B36003E0112231310EC0E0421757B060411797B1ABE200C"
1005 DATA "3A7F7B3C327F7B36003E01121310ED0D2320E43A7F7B217D7
BBEC03A807B23BEC03A817B3D32817BC2887B010100C921717B3
E39BE280334181B363023BE2803341812363023BE28033418093630
23BE010000C834CD827B793D010100C818CF"
1010 RESTORE 1000: READ A$: READ B$: LET a$=a$+b$
1020 FOR N=1 TO LEN A$:2: LET B$=A$(2*N-1 TO 2*N)
1030 LET B=16*(CODE B$(1)-48-7*(B$(1)>="A"))+CODE B$(2)-48-7*
(B$(2)>="A")
1040 POKE 31617+N,B: NEXT N: GO TO 2060
1050 POKE 31600,0
1060 FOR N=31601 TO 31604: POKE N, CODE "0": NEXT N
1070 DIM A$(4)
1080 RETURN
1200 LET L=1: IF PEEK 31600=0 THEN LET A$="0000": RETURN
1210 LET L=USR 31769
1220 FOR N=0 TO 3: LET A$(N+1)=CHR$(PEEK (31601+N)): NEXT N
1230 RETURN
1400 LET ATCNT=PEEK 31600: POKE 31600,ATCNT+1
1410 FOR N=0 TO 3: POKE (31000+4*ATCNT+N), CODE A$(N+1): NEXT N
1420 POKE (31400+2*ATCNT),G: POKE (31401+2*ATCNT),C
1430 RETURN
1600 CLS: GO SUB 1050: PRINT AT 0,9: PAPER 1: INK 7:
"MASTERMIND": PRINT INK 5: AT 2,10:"*****CORRECT**
GOOD": LET T=3
1610 GO SUB 1200: PRINT AT T,0:T-2: TAB 10:A$:
1615 IF L=0 THEN INPUT AT 0,0: PAPER 2:"YOU HAVE MADE A MISTAKE
IN*****SCORING. PRESS ENTER": LINE Z$: RETURN
1620 INPUT AT 0,0:"HOW MANY CORRECT DIGITS*****ENTER 0 TO
4": LINE Z$: IF Z$>"4" OR Z$<"0" THEN GO TO 1620
1630 LET c=VAL Z$(1)
1640 INPUT AT 0,0:"HOW MANY GOOD*****DIGITS*****ENTER 0
TO 4": LINE Z$: IF Z$>"4" OR Z$<"0" THEN GO TO 1640
1650 LET G=VAL Z$(1): IF C+G>4 THEN GO TO 1620
1655 PRINT AT T,18:C: TAB 26,G: IF C=4 THEN INPUT AT 0,0: PAPER 2:
"THE ANSWER HAS BEEN FOUND*****PRESS ENTER FOR
ANOTHER GAME": LINE Z$: RETURN
1660 GO SUB 1400: LET T=T+1: GO TO 1610
2000 CLEAR 30999: INK 7: PAPER 0: BRIGHT 1: BORDER 0: CLS
2010 PRINT AT 0,9: PAPER 1:"MASTERMIND"
2020 PRINT INK 5:"*****Mastermind is a number*****guessing game for

```

```

two,▲▲one has to think of a four digit number and the▲▲other has to
guess it"
2030 PRINT " INK 6; "▲▲▲▲After every guess two scores are given. The scores
are called Correct and Good. The Correct▲▲▲score is the number of digits
in the guess that have the same▲▲▲value and are in the same▲▲▲▲▲
position as digits in the answer"
2040 PRINT " INK 5; "▲▲▲▲The Good score is the number of digits that have
the right▲▲▲value but are in the wrong▲▲▲▲▲position. A digit cannot
be▲▲▲▲▲counted in both scores and it▲▲▲cannot add to the Good
score▲▲▲▲twice."
2050 GO TO 1000
2060 INPUT "Press enter to continue▲";a$
2070 CLS : PRINT INK 4; "▲Here are some examples of good and correct
scores" INK 5 "Answer▲▲3203▲▲Correct▲▲Good"
2080 DATA 1234,1,1,5006,1,0,2468,0,1,3203,4,0: RESTORE 2080: FOR n=6
TO 12 STEP 2: READ a,b,c: PRINT AT n,9; INK 6;a; TAB 18;b; TAB 27;
c: NEXT n
2090 INPUT "Press enter to continue▲";a$
2100 CLS : PRINT AT 0,10: PAPER 1;"▲MASTERMIND▲"
2110 PRINT " INK 2; PAPER 7;"▲YOUR OPPONENT IS THE
SPECTRUM▲" "▲YOU CAN PLAY IN TWO WAYS▲"
2120 PRINT " PAPER 2;"1:▲SPECTRUM THINKS OF A NUMBER▲▲AND YOU
GUESS IT▲"
2130 PRINT " PAPER 2;"2:▲YOU THINK OF A NUMBER FOR THE
SPECTRUM TO GUESS▲"
2140 INPUT "ENTER YOUR CHOICE 1 OR 2▲"; LINE Z$: IF Z$="1" THEN
GO SUB 3000: GO TO 2100
2150 IF Z$="2" THEN GO SUB 1600: GO TO 2100
2160 GO TO 2140

3000 CLS : PRINT AT 0,9: PAPER 1;"▲MASTERMIND▲"
3010 DIM A$(4): FOR N=1 TO 4: LET A$(N)=CHR$(CODE "0"+INT
(RND*10)): NEXT N: LET T=1
3020 PRINT AT 2,10; INK 5;"*****▲CORRECT▲▲GOOD"
3030 INPUT AT 0,0;"ENTER YOUR GUESS, 4 DIGITS OR▲▲" "XXXX" "▲TO
GIVE UP▲"; LINE Z$
3040 IF Z$="XXXX" OR Z$="xxxx" THEN PRINT PAPER 2;"ANSWER IS";
TAB 10;A$: INPUT "PRESS ENTER"; LINE Z$: RETURN
3050 IF LEN Z$=4 THEN FOR N=1 TO 4: IF Z$(N)>="0" AND Z$(N)<="9"
THEN NEXT N: GO TO 3070
3060 GO TO 3030
3070 PRINT T; TAB 10;Z$;
3080 LET B$=A$: LET C$=Z$: LET C=0: LET G=0
3090 FOR N=1 TO 4: IF B$(N)=C$(N) THEN LET C=C+1: LET B$(N)="B";
LET C$(N)="C"

```

```

3100 NEXT N
3110 FOR N=1 TO 4: FOR M=1 TO 4: IF B$(N)=C$(M) THEN LET G=G+1:
LET B$(N)="B": LET C$(M)="C"
3120 NEXT M: NEXT N
3130 PRINT TAB 18;C; TAB 26;G: LET T=T+1
3140 IF C=4 THEN INPUT PAPER 2;"ANSWER FOUND PRESS ENTER▲";
LINE Z$: RETURN
3150 GO TO 3030

```

Breakout

This program is based upon a famous arcade game. A succession of coloured walls start to descend from the top of the screen. A yellow ball bounces around the screen and you have to control the movement of a paddle along the bottom of the court so that the ball hits the paddle and rebounds onto the wall again. Every time the ball hits a wall, it removes a brick and you score a point. Every time the ball misses the paddle and hits the bottom of the court, the coloured walls all move down one place.

The only controls are keys 5 and 8 which move the paddle left and right respectively.

You will soon learn the derivation of the name of the game. If you manage to break through one of the walls, the ball will bounce around between that wall and the next one, scoring many points with no danger of hitting the bottom.

```

100 GO SUB 1000
110 GO SUB 1500
120 GO SUB 2500
130 GO SUB 3500
140 GO TO 120
1000 REM SET UP
1010 BORDER 0: PAPER 0: INK 7: BRIGHT 1: CLS: PRINT AT 0,11: PAPER
1: "▲BREAKOUT▲"
1020 DATA 0,126,126,126,126,126,126,0
1030 DATA 24,60,126,255,255,126,60,24
1040 FOR M=0 TO 1: FOR N=0 TO 7: READ A: POKE USR (CHR$
(97+M))+N,A: NEXT N: NEXT M
1050 LET C$=CHR$ 144: LET D$=CHR$ 145
1060 DIM B$(5,4,32)
1070 FOR A=1 TO 5: FOR B=1 TO 4: FOR C=1 TO 32
1080 LET B$(A,B,C)=C$
1090 NEXT C: NEXT B: NEXT A
1100 LET Y1=-26: LET X3=0: LET Y3=20: LET DY=-1: LET DX=1: LET
SCR=0: LET X4=16
1110 RETURN
1500 REM PRINT WALL
1510 FOR Y=0 TO 4: FOR J=0 TO 3
1520 LET Y2=6*Y+J+Y1
1530 IF Y2<1 THEN GO TO 1560
1540 IF Y2>18 THEN RETURN
1550 GO SUB 3500: BEEP .02,10*Y+J: PRINT AT Y2,0: INK (Y+1)+6*NOT
(Y OR J): B$(Y+1,J+1)
1560 NEXT J
1570 FOR J=0 TO 1
1580 LET Y2=6*Y+4+J+Y1

```

38

N.B. ▲ denotes space, see introduction.

```

1590 IF Y2<1 THEN GO TO 1620
1600 IF Y2>18 THEN RETURN
1610 GO SUB 3500: PRINT AT Y2,0: "▲"
1620 NEXT J: NEXT Y
1630 RETURN
2000 REM CHANGE WALL
2010 LET Y=1+INT ((Y3-Y1)/6)
2020 LET J=1+Y3-Y1-6*(Y-1)
2030 LET B$(Y,J,X3+1)="▲"
2040 LET SCR=SCR+1: PRINT AT 0,0: SCR: BEEP .05,40
2050 RETURN
2500 REM MOVE BALL
2510 IF Y3+DY<1 OR Y3+DY>20 THEN LET DY=-DY: BEEP .1,20
2520 IF X3+DX<0 OR X3+DX>31 THEN LET DX=-DX: BEEP .1,10
2530 PRINT AT Y3,X3: "▲"
2540 LET Y3=Y3+DY: LET X3=X3+DX
2560 IF (SCREEN$ (Y3,X3)="▲") THEN GO TO 2610
2570 GO SUB 2000: LET F1=1
2580 IF SCREEN$ (Y3-DY,X3)="▲" THEN LET DY=-DY: LET F1=0
2590 IF SCREEN$ (Y3,X3-DX)="▲" THEN LET DX=-DX: LET F1=0
2600 IF F1 THEN LET DY=-DY: LET DX=-DX
2610 PRINT AT Y3,X3: INK 6:D$
2620 IF Y3=20 AND DY=1 THEN GO SUB 3000
2630 RETURN
3000 REM check paddle
3010 IF X3=X4 THEN RETURN
3020 LET Y1=Y1+1
3030 IF Y1=19 THEN INPUT AT 0,0: INK 2: PAPER 6: FLASH 1: "GAME
END. PRESS ENTER TO REPLAY▲": LINE Z$: RUN
3040 PRINT AT Y3,X3: INK 6:D$: GO SUB 1500
3050 RETURN
3500 REM PADDLE MOVE
3510 LET L=IN (61438): LET L=L-32*INT (L/32)
3520 LET M=IN (63486): LET M=M-32*INT (M/32)
3530 PRINT AT 21,X4: "▲"
3540 LET X4=X4+(X4<31 AND L=27)-(X4>0 AND M=15)
3550 PRINT AT 21,X4: INK 5: "■"
3560 RETURN

```

39

Defender

In this game you have to defend yourself in an area of space that is dominated by aliens. One by one they come into view, heading on a course straight for you, firing randomly as they travel. Whenever one shoots you or scores a direct hit by running into you, you register a loss. You can only sustain 20 losses before you die and your task is to shoot down as many aliens as you can while remaining alive.

You can move around by pressing the cursor movement keys and you can fire up, down, left or right by pressing one of the cursor movement keys while holding down shift at the same time.

Every alien that you destroy is registered as a successful attack.

```

100 LET X1=10: LET Y1=10: LET AT=0: LET HT=0: BORDER 0: INK 7:
  PAPER 0: CLS: PRINT AT 0,22: PAPER 2;"▲DEFENDER▲": PAPER 1:
  AT 5,22;"▲ATTACKS▲": AT 10,22;"▲LOSSES▲": PRINT AT Y1,X1:
  PAPER 5;"▲"
110 GO SUB 2010
120 GO SUB 2000: GO TO 120
1000 PAUSE 1: LET Z$=INKEY$: IF Z$="" THEN PAUSE 5: RETURN
1010 LET X2=X1: LET X1=X1+(Z$="8" AND X1<20)-(Z$="5" AND X1>1)
1020 LET Y2=Y1: LET Y1=Y1+(Z$="6" AND Y1<20)-(Z$="7" AND Y1>1)
1030 IF X2<>X1 OR Y2<>Y1 THEN PRINT AT Y2,X2;"▲": AT Y1,X1: PAPER
  5;"▲": BEEP .01,10: GO TO 1110
1035 IF CODE Z$>11 THEN GO TO 1110
1040 LET X3=4+X1*8: LET Y3=4+(21-Y1)*8: LET X4=X3: LET Y4=Y3
1050 OVER 1: PLOT X3,Y3
1060 LET X4=X4*(CODE Z$=10 OR CODE Z$=11)+175*(CODE Z$=9)
1070 LET Y4=Y4*(CODE Z$=9 OR CODE Z$=8)+175*(CODE Z$=11)
1080 DRAW X4-X3,Y4-Y3: BEEP .05,30: PLOT X3,Y3: DRAW
  X4-X3,Y4-Y3: OVER 0
1090 IF Y1=Y6 AND SGN (X6-X1)=SGN (X4-X3) THEN LET HT=HT+1:
  LET DY=100
1100 IF X1=X6 AND SGN (Y6-Y1)=SGN (Y3-Y4) THEN LET HT=HT+1:
  LET DY=100
1110 PRINT AT 7,26;HT: IF DY=100 THEN PRINT AT Y6,X6: PAPER 4;"▲":
  FOR M=20 TO 0 STEP -2: BEEP .02,M: NEXT M
1120 RETURN
2000 REM ALIENS
2010 LET DY=-1+2*(Y1<=11): LET DX=-1+2*(X1<=11)
2020 LET X5=21*(DX<0): LET Y5=21*(DY<0)
2030 LET NX=X5-X1: LET NY=Y5-Y1
2040 LET NNY=-ABS (NX)*DY+Y1: IF ABS NY>ABS NX THEN LET
  Y6=NNY: LET X6=X5: GO TO 2100
2050 LET Y6=Y5: LET X6=X1-ABS (NY)*DX: GO TO 2100

```

40 N.B. ▲ denotes space, see introduction.

```

2060 PRINT AT Y6,X6: PAPER 2;"▲": IF X1=X6 AND Y1=Y6 THEN GO SUB
  3000: GO TO 2090
2065 IF RND < .2 THEN OVER 1: FOR M=0 TO 1: PLOT 8*X6+4,0: DRAW
  0,175: PLOT 0,8*(21-Y6)+4: DRAW 175,0: NEXT M: OVER 0: IF
  X1=X6 OR Y1=Y6 THEN GO SUB 3000: GO TO 2080
2070 GO SUB 1000
2080 PRINT AT Y6,X6;"▲"
2090 LET Y6=Y6+DY: LET X6=X6+DX
2100 IF Y6<0 OR Y6>21 OR X6>21 OR X6<0 THEN GO TO 2000
2110 GO TO 2060
3000 LET AT=AT+1: PRINT AT 12,26;AT
3010 PRINT AT Y1,X1: PAPER 6;"▲": FOR M=30 TO 50 STEP 2: BEEP
  .05,M: NEXT M: PRINT AT Y1,X1: PAPER 5;"▲"
3020 IF AT<20 THEN RETURN

```

Othello

This is a well-known board game for two players. It is played on an eight by eight board which has two colours of pieces, red for the Spectrum and blue for you. The game commences with just four pieces (two from each side) on the board.

The object of the game is to end up with more of your colour pieces than your opponents. Each player takes a turn to place a piece on the board in such a way that he bridges one or more of his opponent's pieces. That is to say that the player must always place his piece adjacent to an opponent's piece in such a way that the opponent's piece or a row of the opponent's pieces are bridged with one of the player's pieces at either end. The diagram below shows a board with opponent's pieces marked "O", player's "P" and possible moves marked "X".

```

      X
    X O P
      O
    O O
  X O P
  
```

When a move is made, all of the pieces bridged change sides. In the example shown, the top-most move would result in a vertical column of 3 pieces changing, the bottom left move would result in a diagonal column of 2 pieces and an additional adjacent piece changing and the other move would only change a single piece.

The game ends when either player cannot move.

When it is your turn to move, point the arrows at your favoured position by use of the cursor keys 5 to 8, and then press 0. If the Spectrum ignores your move, it is trying to tell you that the move is illegal. If you cannot move, or wish to surrender, press 1.

```

100 PAPER 0: INK 7: BRIGHT 1: BORDER 0: CLS: PRINT AT 0,10: PAPER
    1:"▲OTHELLO▲"
110 GO SUB 1000
120 GO SUB 4000
130 IF RND >.5 THEN GO TO 170
140 PRINT AT 21,0: PAPER 2:"SPECTRUM TO MOVE"
150 GO SUB 3000
160 IF F1=2 THEN GO TO 220
170 PRINT AT 21,0: PAPER 2:"YOU TO MOVE▲▲▲▲▲"
180 LET C$="B"
190 GO SUB 5000
200 IF F1=2 THEN GO TO 220
210 GO TO 140
  
```

42

N.B. ▲ denotes space, see introduction.

```

220 GO SUB 6000: PRINT AT 21,0: PAPER 2:"SPECTRUM▲":A:
    "▲▲YOU▲":B"
230 INPUT "PRESS ENTER TO PLAY AGAIN▲": LINE Z$: RUN
1000 REM SET UP
1010 DATA 0,52,4,12,13,5,53,1,59,60,36,44,45,37,61,54,11,43,20,24,25,21,
    38,6,19,51,31,32,33,26,46,14,18,50,30,34,35,27,47,15,10,42,23,29,28,
    22,39,7,58,63,41,49,48,40,62,55,3,57,9,17,16,8,56,2
1020 DIM X(64): DIM Y(64): FOR Y=1 TO 8: FOR X=1 TO 8
1030 READ A: LET X(A+1)=X: LET Y(A+1)=Y
1040 NEXT X: NEXT Y
1050 DATA 0,0,0,3,7,15,31,31,0,0,0,192,224,240,248,248,31,31,15,7,3,0,0,0,
    248,248,240,224,192,0,0,0
1060 DATA 1,1,49,25,13,7,3,1,128,128,140,152,176,224,192,128,0,0,32,48,
    24,12,6,255,255,6,12,24,48,32,0,0
1070 FOR N=0 TO 7: FOR M=0 TO 7: READ A: POKE USR CHR$(N+CODE
    "A")+M,A: NEXT M: NEXT N
1080 DIM PS(2,2,2): DIM AS(2,2)
1090 FOR J=1 TO 2: FOR K=1 TO 2: LET PS(1,J,K)=CHR$(2*J+K+141):
    LET AS(J,K)=CHR$(2*J+K+145): NEXT K: NEXT J
1100 DIM BS(8,8): LET BS(5,5)="A": LET BS(4,4)="A": LET BS(5,4)="B":
    LET BS(4,5)="B"
1110 RETURN
1500 REM EVALUATE MOVE
1510 LET F1=0
1520 LET D$="A": IF C$=D$ THEN LET D$="B"
1530 IF BS(Y1,X1)<>"▲" THEN RETURN
1540 FOR Y=-1 TO 1: FOR X=-1 TO 1
1550 LET Y2=Y1+Y: LET X2=X1+X
1560 IF Y2=0 OR Y2=9 OR X2=0 OR X2=9 OR NOT (X OR Y) THEN
    GO TO 1590
1570 IF BS(Y2,X2)=D$ THEN GO SUB 2000
1580 IF F1 THEN RETURN
1590 NEXT X: NEXT Y
1600 RETURN
2000 REM EVAL MOVE DIRECTION
2010 LET Y3=Y2: LET X3=X2
2020 LET Y3=Y3+Y: LET X3=X3+X: IF Y3=0 OR Y3=9 OR X3=0 OR X3=9
    THEN RETURN
2030 IF BS(Y3,X3)=D$ THEN GO TO 2020
2040 IF BS(Y3,X3)="▲" THEN RETURN
2050 LET F1=1: RETURN
2500 REM MAKE MOVE
2510 LET D$="A": IF D$=C$ THEN LET D$="B"
2520 LET BS(Y1,X1)=C$: LET X3=X1: LET Y3=Y1: GO SUB 4500
2530 FOR Y=-1 TO 1: FOR X=-1 TO 1
  
```

43

```

2540 LET Y2=Y1+Y: LET X2=X1+X
2550 IF Y2=0 OR Y2=9 OR X2=0 OR X2=9 OR NOT (X OR Y) THEN
  GO TO 2610
2560 LET F1=0: IF B$(Y2,X2)=D$ THEN GO SUB 2000
2570 IF NOT F1 THEN GO TO 2610
2580 LET B$(Y2,X2)=C$: LET Y3=Y2: LET X3=X2: GO SUB 4500: LET
  Y2=Y2+Y: LET X2=X2+X
2590 IF B$(Y2,X2)=C$ THEN GO TO 2610
2600 GO TO 2580
2610 NEXT X: NEXT Y
2620 RETURN
3000 REM COMPUTER'S MOVE
3010 LET C$="A": PRINT AT 2,6;"▲": FOR K=4 TO
  19: PRINT AT K,4;"▲": NEXT K
3020 FOR K=1 TO 64: BEEP .1,K: LET Y1=Y(K): LET X1=X(K): GO SUB 1500
3030 IF NOT F1 THEN NEXT K
3040 IF F1 THEN GO SUB 2500: RETURN
3050 LET F1=2: RETURN
4000 REM PRINT BOARD
4010 LET FL=1
4020 FOR Y=1 TO 8: FOR X=1 TO 8
4030 LET Y3=Y: LET X3=X: GO SUB 4500
4040 NEXT X: NEXT Y
4050 LET FL=0: RETURN
4500 REM PRINT PIECE
4510 FOR J=1 TO 2
4520 LET P=1+(B$(Y3,X3)="▲"): PRINT AT 1+2*Y3+J,4+2*X3: INK
  1+(B$(Y3,X3)="A"): PAPER 7-((X3+Y3)/2=INT ((X3+Y3)/2)):PS(P,J)
4530 NEXT J
4540 IF FL=0 THEN FOR K=30 TO 15 STEP -1: BEEP .008,K: NEXT K
4550 RETURN
5000 REM USER'S MOVE
5010 LET S=1: LET R=1
5020 GO SUB 5500
5030 LET Z$=INKEY$: IF Z$="" THEN GO TO 5030
5040 BEEP .1,20: LET S=S+(Z$="8" AND S<8)-(Z$="5" AND S>1)
5050 LET R=R+(Z$="6" AND R<8)-(Z$="7" AND R>1)
5060 IF Z$="0" THEN LET X1=S: LET Y1=R: GO SUB 1500: IF F1 THEN GO
  SUB 2500: RETURN
5070 IF Z$="1" THEN LET F1=2: RETURN
5080 GO TO 5020
5500 REM print arrows
5510 PRINT AT 2,6;"▲"
5520 FOR J=4 TO 19: PRINT AT J,4;"▲": NEXT J
5530 PRINT AT 2,4+2*S;A$(1)

```

```

5540 FOR J=1 TO 2: PRINT AT 1+2*R+J,4;A$(2,J): NEXT J
5550 RETURN
6000 REM SCORE
6010 LET A=0: LET B=0
6020 FOR Y=1 TO 8: FOR X=1 TO 8
6030 LET A=A+(B$(Y,X)="A"): LET B=B+(B$(Y,X)="B")
6040 NEXT X: NEXT Y
6050 RETURN

```

Kingdom

This is a land and resources management game in which you have been elected to a 5 year term of office as ruler of your kingdom. Your country is about 30 miles by 70 miles and has an area of 2,000 square miles, of which half is useful for agriculture and mining. Your job is to decide upon the budget for your country for each year. Your currency is in Rods and each of your countrymen need to have 100 Rods per annum spent on them to survive. You may plant your land at a cost of between 10 and 15 Rods per square mile and the crops, when sold, will fetch about 60 Rods per square mile. Crop yields, however, may be adversely affected by pollution. You may also sell your land for mining to foreign industry at about 100 Rods/square mile. Foreign workers will join your country but the industry will provide all of their needs. As industry grows, so does pollution, although you can limit this by spending money on controlling it. Pollution kills the crops, kills your countrymen and also destroys wildlife – thus making the country less attractive to tourists who are another valuable source of income. On average, tourists spend 30 Rods each whenever they visit.

There are several ways in which you can be thrown out of office. If you allow your countrymen to starve when there is money in the treasury, or if more than one hundred people die in one year, or if your nation's population drops by more than a third, then you will find yourself without a job. Another threat is posed by the foreign workers in your country, who, when they outnumber your nationals, are likely to revolt.

Good luck.

```
100 GO SUB 1000: GO SUB 1500
110 GO SUB 2000
120 FOR Y=1 TO 5
130 GO SUB 2500: GO SUB 4000: GO SUB 1500: GO SUB 4500
140 INPUT "press enter"; LINE Z$: IF FI=0 THEN NEXT Y: CLS: PRINT
    "Your success has been" " " "outstanding you have been" " " "elected to
    another term in" " "office.": INPUT "PRESS ENTER"; LINE Z$:
    GO TO 120
150 CLS: PRINT "Your failure has brought an" " " "early end to your rule.":
    STOP
1000 REM GRAPHICS
1010 DATA 68,78,95,255,255,255,255,255
1020 DATA 24,44,124,60,24,28,58,89,156,24,40,72,140,68,196,12
1030 DATA 131,224,24,7,7,24,224,131,128,70,47,255,255,47,70,128
1040 DATA 0,0,0,0,0,0,36,126
1050 DATA 24,153,219,90,90,255,255,255
1060 RESTORE 1000: FOR N=0 TO 6: FOR M=0 TO 7
1070 READ A: POKE USR CHR$(97+N)+M,A: NEXT M: NEXT N
1080 LET A$=CHR$ 147+CHR$ 148+"▲"
```

```
1090 DIM B$(2,2): LET B$(1)=CHR$ 145: LET B$(2)=CHR$ (146)
1100 DATA 0,0,0,0,0,0,0,0: READ IN,DIN,PL,ST,PO,NP,TR
1110 RETURN
1500 REM DISPLAY
1510 PAPER 0: INK 7: BRIGHT 1: BORDER 0: CLS
1520 PRINT AT 0,11: PAPER 2: "▲KINGDOM▲"
1530 FOR N=-6 TO 6 STEP 2: LET N1=ABS N
1540 FOR M=N TO N+1: PRINT AT 9+M,8+N1:
1550 FOR P=1 TO 16-2*N1: PRINT: PAPER 1: "▲": NEXT P
1560 NEXT M: NEXT N
1570 IF IN=0 THEN GO TO 1630
1580 PRINT AT 20,0: "INDUSTRY CONTINUES▲": IF DIN THEN PRINT "TO
    GROW":
1590 PRINT TAB 0: BEEP .5,30: PAUSE 50: LET P=INT (IN/25)+1: LET
    P1=INT (P/10)
1600 LET P=P-10*P1
1610 FOR Q=0 TO P1: FOR R=0 TO 10*(Q<P1)+P*(Q=P1)
1620 PRINT AT 6+2*Q,16+R: PAPER 1: INK 6: CHR$ 144: BEEP
    .2,R+10*Q: NEXT R: NEXT Q
1630 PAUSE 25: IF DIN=0 THEN GO TO 1690
1640 PRINT AT 20,0: "BRINGING MORE FOREIGN WORKERS": TAB 0:
    BEEP .5,30: PAUSE 50
1650 LET P=INT (DIN/20)+1
1660 IF P>6 THEN LET P=6
1670 FOR N=1 TO P: FOR M=30 TO 11+N STEP -1: FOR R=1 TO 2: PRINT
    AT 12+R,M: INK 3: PAPER 8: B$(R)
1680 NEXT R: BEEP 0.05,M+2*N: NEXT M: PAUSE 30: NEXT N
1690 PAUSE 25: IF ST+PO=0 THEN GO TO 1780
1700 IF ST=0 THEN GO TO 1720
1710 PRINT AT 20,0: ST: "▲PEOPLE DIED OF STARVATION": TAB 0: BEEP
    .5,30: PAUSE 50
1720 IF PO=0 THEN GO TO 1740
1730 PRINT AT 20,0: PO: "▲PEOPLE DIED FROM POLLUTION": TAB 0:
    BEEP .5,30: PAUSE 50
1740 LET P=INT ((ST+PO)/25)+1: IF P>6 THEN LET P=6
1750 FOR N=1 TO P: FOR M=8 TO 7 STEP -1: FOR R=1 TO 2: PRINT AT
    10+R,M: INK 4: B$(R): NEXT R: BEEP .3,2*M-2*N: NEXT M
1760 PAUSE 15: PRINT AT 11,7: "▲": PRINT AT 12,7: INK 4: A$: BEEP 1.0:
    PAUSE 15: FOR M=7 TO 0 STEP -1: PRINT AT 12,M: INK 4: A$: BEEP
    .05,2*M-2*N: NEXT M: PRINT AT 12,0: "▲"
1770 NEXT N
1780 PAUSE 25: IF PL=0 THEN GO TO 1860
1790 PRINT AT 20,0: "THE LAND WAS PLANTED": TAB 0: BEEP .5,30:
    PAUSE 50: LET PL1=INT (PL/25)+1
1800 LET PL2=INT (PL1/10): LET PL3=PL1-10*PL2
```

```

1810 FOR N=0 TO PL2: FOR M=0 TO 10*(N<PL2)+PL3*(N=PL2)
1820 PRINT AT 10-2*N,14-M; INK 4; PAPER 1; CHRS 149: BEEP
      .1,M+10*N: NEXT M: NEXT N
1830 PAUSE 25: PRINT AT 20,0;"THE CROPS GREW"; TAB 0: BEEP .5,30:
      PAUSE 50
1840 FOR N=0 TO PL2: FOR M=0 TO 10*(N<PL2)+PL3*(N=PL2)
1850 PRINT AT 10-2*N,14-M; INK 4; PAPER 1; CHRS 150: BEEP
      .1,M+10*N: NEXT M: NEXT N
1860 IF NP=0 THEN GO TO 1910
1870 PAUSE 25: PRINT AT 20,0;NP;"PEOPLE JOINED THE COUNTRY";
      TAB 0: BEEP .5,30: PAUSE 50
1880 LET NP1=INT (NP/10)+1: IF NP1>10 THEN LET NP1=10
1890 FOR N=1 TO NP1: FOR M=30 TO 17+N STEP -1: FOR R=1 TO 2:
      PRINT AT 12+R,M; INK 5; PAPER 8; BS(R)
1900 NEXT R: BEEP .05,M: NEXT M: NEXT N
1910 IF TR=0 THEN GO TO 1970
1920 PRINT AT 20,0;TR;"TOURISTS VISITED THE COUNTRY"; TAB 0:
      BEEP .5,30: PAUSE 50
1930 LET TR1=INT (TR/100)+1: IF TR1>6 THEN LET TR1=6
1940 FOR N=1 TO TR1: LET RD=INT (5*RND): FOR M=25 TO 7 STEP -1:
      FOR R=1 TO 2
1950 PRINT AT 16+R,M; INK 2+RD; BS(R): NEXT R: BEEP .05,M+2*N:
      PAUSE (M=16)*15+1: NEXT M
1960 PRINT AT 17,7;"▲": AT 18,7;"▲": NEXT N
1970 RETURN

2000 REM INIT
2010 LET A=60000+INT ((RND-RND)*1000)
2020 LET B=500+INT ((RND-RND)*10)
2030 LET C=0: LET D=2000
2040 RETURN
2500 REM INPUT
2510 LET W=100+INT ((RND-RND)*10)
2520 LET V=10+INT (RND*6)
2530 CLS
2535 PRINT "you have a 5 year term of office" "this is the state of
      your" "country at the start of year▲";y
2540 PRINT "Your country has▲";d;"sq. miles" "of land of which 1000 sq.
      miles" "is unusable."
2550 IF IN THEN PRINT IN;"sq. miles have already been" "sold to foreign
      industry."
2560 PRINT "You have▲";b;"countrymen."
2570 IF c THEN PRINT c;"foreign workers are also" "living there"
2580 PRINT "Land can be sold to foreign" "industry at▲";w;"Rods per
      square" "mile."

```

```

2590 PRINT "Agriculture will cost▲";v;"Rods" "per square mile to be
      planted."
2600 PRINT "Your treasury contains▲";a;"Rods."
2610 PRINT "How many sq. miles of land will" "you sell▲";: INPUT H
2620 IF H>1000-IN THEN INPUT AT 0,0;("You only have▲";1000-in;"▲to
      sell." "Try again▲");h: GO TO 2620
2630 PRINT h
2640 PRINT "How many Rods will you spend on" "your countrymen▲";:
      INPUT i
2650 IF i>a THEN INPUT AT 0,0;("Your treasury only has▲";a;"Rods. Try
      again▲");i: GO TO 2650
2660 PRINT i
2670 PRINT "How many sq. miles will you" "plant▲";: INPUT j
2680 IF j>a-i THEN INPUT AT 0,0;("You only have▲";a-i;"▲Rods
      left" "Try again▲");j: GO TO 2680
2690 IF j>b*1.5 THEN INPUT AT 0,0;("Each countryman can only
      plant" "1.5 sq. miles. Try again▲");j: GO TO 2680
2700 IF j>1000-in-h THEN INPUT AT 0,0;("You only have▲";1000-in-h;
      "sq. miles" "for planting. Try again▲");j: GO TO 2680
2710 PRINT j
2720 PRINT "How many Rods will you spend on" "pollution control▲";:
      INPUT k
2730 IF k>a-i-j*v THEN INPUT AT 0,0;("You only have▲";a-i-j*v;"Rods
      left." "Try again▲");k: GO TO 2730
2740 RETURN

4000 REM RESULTS
4010 LET A1=A-I-V*J-K: LET A=A+H*W-(I-V*J-K)
4020 LET DIN=H: LET IN=IN+H
4030 LET PL=J
4040 LET ST=INT (B-I/100): IF ST<0 THEN LET ST=0
4050 LET PO=INT ((IN/5-K/2500)+INT (RND*10)*(IN>0))
4060 IF PO<0 THEN LET PO=0
4070 LET NP=INT ((I/100-B)/10+K/25-IN/50-PO-ST)
4080 IF NP<0 THEN LET NP=0
4090 LET NP=INT (NP): LET B=B+NP-ST-PO
4100 LET TR=INT (B-IN): IF tr<0 THEN LET tr=0
4110 LET C=C+INT (H/2)+INT (RND*10)*(H<>0)
4120 LET CR=1-IN/2000
4130 LET A=A+INT (CR*PL*60)+INT (tr*40)
4140 RETURN
4500 REM CONSEQUENCES
4510 CLS: LET FL=0
4520 IF ST+PO>0 THEN PRINT "You had to pay▲";INT
      (9*(ST+PO));"Rods in" "funeral expenses.": LET A=INT

```

```

(A-9*(ST+PO)): IF A<0 THEN LET IN=IN+INT (A/W): LET A=0:
PRINT "You had to sell land to raise" "the cash."
4530 IF CR<.95 THEN PRINT "Pollution reduced the harvest"
4540 IF CR<.8 THEN PRINT "seriously"
4550 PRINT INT (CR*J); "▲sq. miles of crops were" "successfully harvested
and sold" "for▲"; INT (cr*j*60); "▲Rods."
4555 IF tr THEN PRINT "You gained▲"; INT (tr*40); "▲Rods
from" "tourism."
4560 IF a1 AND st THEN PRINT "You allowed people to starve" "when there
was money in the" "treasury."; LET fl=1
4570 IF c>b THEN PRINT "There are now more foreign" "workers than
nationals in your" "country and they are staging a" "revolt."; LET fl=1
4580 IF st+po>100 THEN PRINT "You have allowed more than one"
" hundred people to die this year."; LET fl=1
4590 IF b<330 THEN PRINT "During your period in office," "your population
has fallen by" "more than a third."; LET fl=1
4600 RETURN

```

Fighter

Despite its small size, I'm sure that you will find this to be one of the most compulsive games in the collection.

You are a fighter pilot shooting down enemy VIC 20s as they appear. You can move your sights with the cursor movement keys 5 to 8 and fire by pressing "0".

At first the enemy appear silently on reconnaissance missions and so you have to watch carefully for them – they disappear almost immediately. You must aim your sights at where they last appeared and fire into the darkness. If you score a hit you will hear them tumble out of the skies.

As you become more proficient and your enemies get shot down, they appear more often. Remember, you must always aim at the last point at which they appeared so don't let your gaze wander for a moment in case you miss an appearance.

When the ratio of appearances to those shot down reaches a certain limit, the enemy start attacking – they come in close and fire as they appear.

Eventually you will die, but how many of the enemy can you take with you.

```

100 BORDER 0: PAPER 0: INK 7: CLS: LET x2=0: LET y2=200: LET ap=0:
LET sd=0: GO SUB 1000: PRINT AT 0,22: PAPER 2: "▲FIGHTER▲";
PAPER 1; AT 5,22: "▲APPEARED▲"; AT 10,22: "SHOT DOWN"
110 GO SUB 2000
1000 LET x=4: LET y=4: OVER 1: PLOT x,0: DRAW 0,175: PLOT 0,y: DRAW
175,0: OVER 0
1010 PAUSE 1: LET z$=INKEY$
1020 IF z$="" THEN PAUSE 5: RETURN
1030 LET x1=x: LET y1=y
1040 LET x=x+8*((z$="8" AND x<172)-(z$="5" AND x>4))
1050 LET y=y+8*((z$="7" AND y<172)-(z$="6" AND y>4))
1060 IF x1<>x THEN OVER 1: PLOT x1,0: DRAW 0,175: PLOT x,0: DRAW
0,175
1070 IF y<>y1 THEN OVER 1: PLOT 0,y1: DRAW 175,0: PLOT 0,y: DRAW
175,0
1090 OVER 0: IF z$="0" THEN LET y3=INT ((175-y)/8): LET x3=INT (x/8):
PRINT AT y3,x3: PAPER 2: OVER 1: "▲": BEEP .2,30: PRINT AT y3,x3:
OVER 1: "▲": IF y3=y2 AND x3=x2 THEN FOR m=30 TO 10 STEP -2:
BEEP .05,m: NEXT m: LET sd=sd+1: PRINT AT 12,26: SD: LET
Y3=200
1100 RETURN
2000 LET N1=(150-5*SD-10*(ap/(sd+1)))*(1+RND*.5): LET
N1=N1*(N1>0)+10: FOR N=0 TO N1
2010 GO SUB 1010
2020 NEXT n
2025 IF AP/(SD+1)>=2 THEN GO TO 2070

```

N.B. ▲ denotes space, see introduction.

```

2030 LET X2=INT (RND*22): LET Y2=INT (RND*22)
2040 PRINT AT Y2,X2: OVER 1: PAPER 5;"▲"
2050 PAUSE 10: PRINT AT Y2,X2: OVER 1;"▲"
2060 GO TO 2100
2070 LET X3=INT (X/8): LET Y3=INT ((175-Y)/8): LET X2=X3+INT
(4*(RND-RND)): LET Y2=Y3+INT (4*(RND-RND)): IF X2<0 OR
X2>21 OR Y2<0 OR Y2>21 THEN GO TO 2070
2080 PRINT AT Y2,X2: PAPER 5: OVER 1;"▲": FOR M=0 TO 5: BEEP
.03,10: PAUSE 3: NEXT M: IF X2=X3 AND Y2=Y3 THEN STOP
2090 PRINT AT Y2,X2: OVER 1;"▲"
2100 LET ap=ap+1: PRINT AT 7,26;AP: GO TO 2000

```

Millionaire

At last you have the chance to make a million. A wealthy relative has died and left you ten thousand pounds with specific instructions that you are to invest it in the stock market with the aim of becoming a millionaire. Your relative took a great personal interest in the stock market and has recommended five companies for you to invest in.

- 1: M&S, Minerals and Steel company, in the minerals industry
- 2: GKN, General Kitchen Needs, in the catering industry
- 3: BR, The Battery Radio company, an electronics firm
- 4: ICL, Inter City Lorries, experts in transport
- 5: BL, British Linen, a homespun textiles business

Each day presents you with the opportunity to trade on the stock market. When you have made all of the transactions that you want, at the end of the day you read the evening papers, and any news that may affect the prospects of the five companies is displayed. You are then given the chance to see a graph of variation of share prices for any of the five companies that you are interested in.

After that you go to bed, dream about being a millionaire and wake up to view your table of stocks and decide, on the basis of the previous evening's news, which to buy and which to sell. And so life goes on.

Some of the news items will mention a particular industry or company and its effects will be limited to that particular sector of the economy. Other items will be more general, and will affect the whole of industry.

The fluctuations in prices always follow trends that can be predicted from the news flashes, so don't ignore the news.

```

100 GO SUB 500
110 GO SUB 1500: GO SUB 2000: GO SUB 2500: GO SUB 3500: INPUT
"PRESS ENTER TO CONTINUE▲": LINE Z$: GO SUB 1000: GO TO 110
500 REM init
510 BORDER 0: INK 7: BRIGHT 1: PAPER 0: CLS
520 DIM C$(5,3): DIM D$(5,11): DIM P(5,20): DIM D(5): DIM E(5): DIM Q(5):
LET DAY=1: LET CSH=10000
530 DATA "M&S","Minerals","GKN","Catering","BR","Electronics",
"ICL","Transport","BL","Textiles"
540 FOR n=1 TO 5: READ c$(n),d$(n): LET P(N,1)=100+INT
(20*RND-20*RND)
550 NEXT N
560 RETURN
1000 REM display graphs
1010 CLS: PRINT AT 0,5: PAPER 1;"▲DISPLAY PRICE GRAPHS▲"

```

```

1020 PRINT " PAPER 2; "1-5 DISPLAY A GRAPH"; PAPER 0; "1=M&S"
      "2=GKN" "3=BR" "4=ICL" "5=BL" INK 6; "6=SHOW
      STOCK TABLE"
1030 INPUT "ENTER 1 TO 6"; LINE Z$: IF Z$>="1" AND Z$<="6" THEN
      LET CO=VAL Z$: GO TO 1050
1040 GO TO 1030
1050 IF CO=6 THEN RETURN
1060 CLS: PRINT AT 0,0; PAPER 1; "▲SHARE PRICE MOVEMENTS▲";
      C$(CO)'D$(CO)
1070 FOR N=0 TO 360 STEP 20: PRINT AT 20-N/20,0;N: NEXT N
1080 PLOT 8,12: DRAW 245,0: FOR N=3 TO 30: PLOT N*8,12: DRAW 0,-6:
      NEXT N
1090 IF DAY<=20 THEN FOR N=1 TO DAY-1: PLOT 24+N*8,12+P(CO,N)/
      2.5: DRAW 8,-(P(CO,N)-P(CO,N+1))/2.5: NEXT N: GO TO 1130
1100 FOR N=2 TO 20: LET DAY1=DAY-21+N: LET NROT=DAY1-20*INT
      (DAY1-1)/20
1110 LET NROT1=NROT+1-20*(NROT=20): PLOT 24+N*8,12+P
      (CO,NROT)/2.5: DRAW 8,(P(CO,NROT1)-P(CO,NROT))/2.5: NEXT N
1130 INPUT "PRESS ENTER TO CONTINUE▲"; LINE Z$: GO TO 1000
1500 REM PRINT TABLE
1510 CLS: PRINT AT 0,4; PAPER 1; "▲TABLE OF STOCKS HELD▲"
1520 LET NROT=DAY-20*INT ((DAY-1)/20): LET NROT1=NROT-1+20*
      (NROT=1)
1530 PRINT AT 3,0; PAPER 2; "STOCK▲PRICE▲CHANGE▲QTY▲
      VALUE"
1540 FOR N=1 TO 5: PRINT AT 4+N,0; C$(N); TAB 7; P(N,NROT); TAB 14;
1550 LET CN=(P(N,NROT)-P(N,NROT1))*(DAY<>1)
1560 IF CN>0 THEN PRINT "+";CN;
1570 IF CN<0 THEN PRINT CN;
1580 IF CN=0 THEN PRINT "NIL";
1590 PRINT TAB 21;: IF Q(N)=0 THEN PRINT "NIL";: GO TO 1610
1600 PRINT Q(N);
1610 PRINT TAB 26; Q(N)*P(N,NROT)
1620 NEXT N
1630 PRINT INK 5; "CASH IN HAND▲"; CSH
1640 PRINT AT 13,0; PAPER 2; "1-5=▲BUY OR SELL SHARES"; PAPER 0;
      "1=M&S" "2=GKN" "3=BR" "4=ICL" "5=BL" INK 6;
      "6=▲NO FURTHER TRADING"
1650 RETURN
2000 REM TRADE SHARES
2010 INPUT "ENTER A NUMBER 1-6▲"; LINE Z$: IF Z$>="1" AND
      Z$<="6" THEN LET CO=VAL Z$: GO TO 2030
2020 GO TO 2010
2030 IF CO=6 THEN RETURN
2040 FOR N=13 TO 21: PRINT AT N,0; "▲"; TAB 0: NEXT N

```

```

2050 PRINT AT 13,0; "HOW MANY▲"; C$(CO); "▲SHARES DO YOU" "WISH
      TO BUY OR SELL▲" "ENTER A +ve NUMBER TO BUY OR▲" "-ve
      TO SELL▲"
2060 INPUT Q: LET Q=INT Q
2070 IF Q<-Q(CO) THEN INPUT AT 0,0; ("YOU ONLY HAVE▲"; Q(CO);
      "▲SHARES" "TO SELL▲TRY AGAIN▲"); Q: GO TO 2070
2080 IF Q+Q(CO)>999 THEN INPUT AT 0,0; "YOU MAY ONLY HOLD 999
      SHARES OF" "ANY STOCK,▲TRY AGAIN▲"; Q: GO TO 2070
2090 IF Q*P(CO,NROT)>CSH THEN INPUT AT 0,0; "YOU CAN'T AFFORD
      IT▲TRY AGAIN▲"; Q: GO TO 2070
2100 LET Q(CO)=Q(CO)+Q
2110 LET CSH=CSH-Q*P(CO,NROT)
2120 GO SUB 1500
2130 GO TO 2000
2500 REM CHANGE PRICES
2510 CLS: LET DAY=DAY+1: LET NROT=DAY-20*INT ((DAY-1)/20): LET
      NROT1=NROT-1+20*(NROT=1)
2520 FOR N=1 TO 5
2530 LET D(N)=D(N)-1
2540 LET P(N,NROT)=P(N,NROT1)+INT (E(N)*(7+.6*RND))
2550 IF D(N)<=0 THEN LET D(N)=1+INT (RND*10): GO SUB 3000
2560 IF P(N,NROT)<20 THEN LET P(N,NROT)=20
2570 IF P(N,NROT)>340 THEN LET P(N,NROT)=340
2580 NEXT N
2590 RETURN

3000 REM NEW PRICE TRENDS AND PREDICTIONS
3010 LET CO=N: LET DE=SGN (RND-RND): LET DEN=INT
      (RND*10+2)*DE: LET E(CO)=DEN
3020 GO TO 3030+10*INT (8*RND)
3030 PRINT "Major▲"; d$(co); "▲industry" "strike▲";: IF de>0 THEN PRINT
      "averted": RETURN
3035 PRINT "planned": RETURN
3040 PRINT d$(co); "▲industry affected" "by import duty▲";: IF de>0 THEN
      PRINT "reduction": RETURN
3045 PRINT "increase": RETURN
3050 PRINT "World trade conference predicts" "d$(co);▲";: IF de>0 THEN
      PRINT "boom": RETURN
3055 PRINT "slump": RETURN
3060 PRINT "Senior▲"; c$(co); "▲manager▲";: IF de>0 THEN PRINT "wins
      award" "for efficiency": RETURN
3065 PRINT "named in▲" "fraud case": RETURN
3070 PRINT "Scientists announce discovery" "that could make▲"; d$(co);
      "▲industry▲";: IF de>0 THEN PRINT "more profitable": RETURN
3075 PRINT "redundant": RETURN

```

```

3080 PRINT "Government announce new" d$(co); IF de>0 THEN PRINT
    "subsidy": RETURN
3085 PRINT "tax": RETURN
3090 PRINT c$(co); chairman predicts: IF de>0 THEN PRINT "higher
    profits": RETURN
3095 PRINT "more layoffs": RETURN
3100 RETURN
3500 REM GENERAL PREDICTIONS
3510 IF RND < .6 THEN RETURN
3520 LET DE=SGN(RND-RND): FOR N=1 TO 5: IF SGN
    (P(N,NROT)-180)=DE THEN GO TO 3540
3530 LET P(N,NROT)=P(N,NROT)+DE*(5+INT(6*RND)): LET
    E(N)=E(N)+DE
3540 NEXT N
3550 IF DE>0 THEN GO TO 3570+10*INT(RND*5)
3560 GO TO 3620+10*INT(RND*5)
3570 PRINT "Chancellor promises less tax": RETURN
3580 PRINT "Trade secretary announces major" deals with Arab states":
    RETURN
3590 PRINT "Fall in interest rates soon" "says Treasury
    spokesman": RETURN
3600 PRINT "New trade links with China": RETURN
3610 PRINT "Tory popularity grows as" unemployment falls": RETURN
3620 PRINT "Employment secretary predicts" gloomy future": RETURN
3630 PRINT "Interest rates rise by 3/4%": RETURN
3640 PRINT "CBI chairman has little hope for" industrial growth": RETURN
3650 PRINT "TUC talk of general strike soon": RETURN
3660 PRINT "Labour party win key by-election": RETURN

```

Rotate

You are given an array of the numbers 1 to 24 arranged in a random fashion on a 6x4 grid. Your task is to rearrange them into the correct order starting with 1 in the top left corner. You can only move the numbers by rotating groups of four clockwise. To do this, you must move your white marker, which looks like a BR Inter-city symbol, to the centre of the four that you wish to rotate and then press "0". The marker may be moved by pressing the cursor movement keys 5 to 8.

You are allowed to cheat once, by which I mean that the computer allows you to swap two numbers that are horizontally adjacent by putting your marker between them and pressing "0".

You are given a score of how many moves you have taken, so that you can compete against your friends.

If you wish to give up, press 1 and you will be given the chance to start all over again.

```

100 BORDER 0: PAPER 0: INK 7: BRIGHT 1: CLS: PRINT AT 0,12: PAPER
    1:"ROTATE"
110 GO SUB 4000: GO SUB 1500: GO SUB 2500
120 GO SUB 3500
130 IF FL=2 THEN INPUT "PRESS ENTER TO PLAY AGAIN": LINE Z$:
    RUN
140 GO TO 120
1000 REM print no.
1010 DIM B$(2)
1020 LET B$=STR$ B(Y,X)
1030 PRINT AT 2+2*Y,3+3*X: PAPER 6: INK 0: B$
1040 BEEP .2,6*Y+X: RETURN
1500 REM INIT
1510 DIM B(4,6)
1520 FOR N=1 TO 24
1530 LET X=1+INT(RND*6): LET Y=1+INT(RND*4)
1540 IF B(Y,X) THEN GO TO 1530
1550 LET B(Y,X)=N: NEXT N
1560 LET CNT=0: LET X3=1: LET Y3=1: LET FL=1
1670 RETURN
2000 REM ROTATE
2010 LET B=B(Y3,X3): LET B(Y3,X3)=B(Y3+1,X3)
2020 LET B(Y3+1,X3)=B(Y3+1,X3+1)
2030 LET B(Y3+1,X3+1)=B(Y3,X3+1)
2040 LET B(Y3,X3+1)=B
2050 FOR Y=Y3 TO Y3+1: FOR X=X3 TO X3+1: GO SUB 1000: NEXT X:
    NEXT Y
2060 LET CNT=CNT+1: PRINT AT 0,0: CNT

```

N.B. ▲ denotes space, see introduction.

```

2070 RETURN
2500 FOR Y=1 TO 4: FOR X=1 TO 6
2510 GO SUB 1000
2520 NEXT X: NEXT Y: PRINT AT 5,8; CHR$ 144
2530 RETURN
3000 REM SWAP
3010 LET B=B(Y3+.5,X3): LET B(Y3+.5,X3)=B(Y3+.5,X3+1)
3020 LET B(Y3+.5,X3+1)=B
3030 FOR X=X3 TO X3+1: LET Y=Y3+.5
3040 GO SUB 1000: NEXT X
3050 LET CNT=CNT+1: PRINT AT 0,0; CNT
3060 LET FL=0
3080 RETURN
3500 REM MOVE
3505 PAUSE 10: IF INKEY$ <> "" THEN GO TO 3505
3510 LET Z$=INKEY$
3520 IF Z$="" THEN GO TO 3510
3530 BEEP .2,30: PRINT AT 3+2*Y3,5+3*X3;"▲"
3540 LET X3=X3+(Z$="8" AND X3<5)-(Z$="5" AND X3>1)
3550 LET Y3=Y3+.5*(1+NOT FL)*((Z$="6" AND Y3<3.4)-(Z$="7" AND
Y3>.6)): IF NOT FL THEN LET Y3=INT Y3+(Y3<.9)
3560 PRINT AT 3+2*Y3,5+3*X3; CHR$ 144
3570 IF Z$="0" AND FL AND (ABS (Y3-INT (Y3))>.1) THEN GO SUB 3000:
RETURN
3580 IF Z$="0" AND (ABS (Y3-INT (Y3))<.1) THEN GO SUB 2000: RETURN
3590 IF Z$="1" THEN LET FL=2
3600 RETURN
4000 DATA 8,4,254,0,0,127,32,16
4010 FOR N=0 TO 7: READ A: POKE USR "A"+N,A: NEXT N
4020 RETURN

```

Red Cross

In this game you have to drive a small, light blue, armoured "scorpion" vehicle around a battlefield. The battle is over and you have to make your way to all of the wounded, each one being marked with a red cross flag.

Your task is endangered by mines planted in the field. Invisible to the naked eye, they can only be detected by the mine detector fitted underneath your vehicle. Every square that you move out of is given a colour to indicate the proximity of mines. Blue is safe, danger increases through red, magenta, green, cyan, yellow to white.

You must trace a path around the battlefield, visiting each of the red crosses in any order, without getting yourself blown up.

Time, however, is not on your side – people are dying – and it ticks away every time you move, so try to trace your path with the minimum number of moves.

```

100 BORDER 0: INK 7: PAPER 0: BRIGHT 1: CLS: PRINT AT 0,10; PAPER
2;"▲RED CROSS▲"
110 GO SUB 3000: GO SUB 1000: GO SUB 2000
120 GO SUB 4000
130 IF NOT FL THEN GO TO 120
140 FOR N=40 TO 0 STEP -2: BEEP 0.01,N: NEXT N
150 PRINT AT Y3,X3; INK 2; PAPER 6; FLASH 1; CHR$ 144
160 FOR N=10 TO 34 STEP 2
170 BEEP N/100,15-N/2: PAUSE N/2: NEXT N: BEEP 2,-2: BEEP .1,40
180 INPUT "PRESS ENTER TO PLAY AGAIN▲"; LINE Z$: RUN
1000 DIM a$(20,32): DIM b(20,32)
1010 FOR n=1 TO 20
1020 LET x=1+INT (RND*32): LET y=1+INT (RND*20)
1030 IF a$(y,x)=CHR$ 145 THEN GO TO 1020
1040 LET A$(Y,X)=CHR$ 145
1050 NEXT N
1060 FOR n=0 TO 19
1070 LET X1=3+INT (RND*28): LET Y1=3+INT (RND*16)
1080 IF B(Y1,X1)=40 OR A$(Y1,X1)=CHR$ 145 THEN GO TO 1070
1090 FOR Y=Y1-1 TO Y1+1: FOR X=X1-1 TO X1+1
1100 LET B(Y,X)=B(Y,X)+(B(Y,X)<6)
1110 NEXT X: NEXT Y
1120 FOR Y=Y1-2 TO Y1+2: FOR X=X1-2 TO X1+2
1130 LET B(Y,X)=B(Y,X)+(B(Y,X)<6)
1140 NEXT X: NEXT Y
1150 LET B(Y1,X1)=40
1160 BEEP .1,2*N: NEXT N
1170 RETURN
2000 REM PRINT BASES

```

```

2010 FOR N=1 TO 20: FOR M=1 TO 32
2020 IF A$(N,M)=CHR$(145) THEN PRINT AT N,M-1; INK 2; PAPER 7;
      CHR$(145): BEEP .1,2*N+M/16
2030 NEXT M: NEXT N
2040 LET X3=0: LET Y3=1: PRINT AT 1,0; INK 5; CHR$(144)
2050 LET CNT=0: BEEP .6,0
2060 RETURN
3000 REM USR GRAPHICS
3010 DATA 0,56,63,56,126,255,126,0
3020 DATA 24,24,255,255,24,24,0,0
3030 DATA 0,42,124,63,124,63,84,0
3040 FOR M=0 TO 2: FOR N=0 TO 7
3050 READ A: POKE (USR CHR$(97+M))+N,A
3060 NEXT N: NEXT M
3070 RETURN
4000 REM MOVE
4010 PAUSE 1: IF INKEY$ <> "" THEN GO TO 4010
4020 PAUSE 1: LET Z$=INKEY$: IF Z$="" THEN GO TO 4020
4030 BEEP .2,30: LET FL=0
4040 LET DX=(Z$="8" AND X3<31)-(Z$="5" AND X3>0)
4050 LET DY=(Z$="6" AND Y3<20)-(Z$="7" AND Y3>1)
4060 IF A$(Y3+DY,X3+1+DX)=CHR$(145) THEN RETURN
4080 PRINT AT Y3+DY,X3+DX; INK 5; CHR$(144)
4090 IF DX OR DY THEN PRINT AT Y3,X3; INK 1+B(Y3,X3+1); CHR$(146)
4100 LET Y3=Y3+DY: LET X3=X3+DX
4110 IF (DX OR DY) THEN LET CNT=CNT+1: PRINT AT 0,0; CNT
4120 IF Z$="0" THEN INPUT "PRESS ENTER TO PLAY AGAIN▲"; LINE Z$:
      RUN
4130 IF B(Y3,X3+1)=40 THEN LET FL=1: RETURN
4140 RETURN

```

Graphics

This program is a programmers utility as well as a lot of fun. It enables you to make up user defined graphics characters in a graphic and interactive way.

Together with the upper and lowercase character set, numbers and symbols, the Spectrum also allows you to generate your own graphics characters. Each character is made up of an eight by eight grid of dots. When the program is run it displays on the left, an enlarged version of a graphics character. Two cursor arrows may be moved by use of the cursor movement keys, to point to one of the 64 pixels within the character. Pressing 1 will turn it into a foreground pixel (white) while pressing 0 will result in a red background element.

On the right is a display area of 5x5 character positions under which appears the lifesize version of the graphics character that you are working on.

Pressing P will enable you to print the current character into the display area, you will be asked for horizontal and vertical coordinates. This is especially useful for building up larger user defined shapes by stacking several characters together.

Pressing N will enable you to start designing a different character. There are 21 in all, labelled A through to U but the program uses the last 4 to draw the cursor arrows, leaving 17, A-Q.

Pressing C will clear the display area.

Pressing S will save your user defined characters onto tape. They may be reloaded with the command 'LOAD "GRAPHICS" CODE' which will only load the user defined graphics and will not upset any of the program or variables in the computer.

```

100 GO SUB 5000
110 GO SUB 3000
120 LET X$="A": GO SUB 1000: PRINT AT 13,21; PAPER 1;X$;"->>":
      LET Z$=" ": GO SUB 3530
130 GO SUB 3500
140 GO TO 130
1000 REM edit
1010 DIM F$(8,8)
1020 FOR N=1 TO 8
1030 LET R=PEEK (USR X$+N-1): LET R2=128: FOR M=1 TO 8: LET
      R1=INT (R/R2): LET F$(N,M)=CHR$(R1+CODE "0"): LET
      R=R-R1*R2: LET R2=R2/2
1040 GO SUB 1500: NEXT M: NEXT N
1050 LET CX=1: LET CY=1
1060 RETURN
1500 REM PRINT PIXEL
1510 LET K=2*M: FOR J=2+2*N TO 3+2*N: PRINT AT J,K; PAPER
      2+5*(F$(N,M)="1");"▲▲"
1520 NEXT J

```

```

1530 RETURN

2000 REM MAKE GRAPHIC
2010 FOR N=1 TO 8
2030 POKE (USR X$+N-1), VAL ("BIN"+F$(N)): NEXT N
2040 RETURN
2500 REM DISPLAY
2510 FOR N=1 TO 5
2520 PRINT AT 5+N,23;B$(N): NEXT N
2530 RETURN

3000 REM CLEAR SCREEN
3010 BORDER 0: PAPER 0: INK 7: BRIGHT 1: CLS
3020 PRINT AT 0,4: PAPER 1: "USER DEFINED GRAPHICS"
3030 PRINT AT 2,22: PAPER 1: "DISPLAY"
3040 PRINT AT 4,23: "ABCDE"
3050 FOR N=1 TO 5: PRINT AT 5+N,21;N: NEXT N
3060 DIM B$(5,5)
3070 PRINT AT 15,19: PAPER 2: "P": PAPER 0: "Print graphic"
3080 PRINT AT 16,19: PAPER 2: "N": PAPER 0: "New graphic"
3090 PRINT AT 17,19: PAPER 2: "C": PAPER 0: "Clear display"
3100 PRINT AT 18,19: PAPER 2: "S": PAPER 0: "Save on tape"
3110 RETURN
3500 REM MOVE
3510 PAUSE 5: IF INKEY$ <> " " THEN GO TO 3510
3520 PAUSE 5: LET Z$=INKEY$: IF Z$=" " THEN GO TO 3520
3530 BEEP .1,30
3540 LET CX=CX+(CX<8 AND Z$="8")-(CX>1 AND Z$="5")
3550 LET CY=CY+(CY<8 AND Z$="6")-(CY>1 AND Z$="7")
3560 PRINT AT 2,2: "*****": AT 2,CX*2;A$(1)
3570 FOR N=4 TO 19: PRINT AT N,0: " ": NEXT N
3580 PRINT AT 2+2*CY,0;A$(2,1): AT 3+2*CY,0;A$(2,2)
3590 IF Z$="P" OR Z$="p" THEN GO SUB 4000
3600 IF Z$="N" OR Z$="n" THEN GO SUB 4500
3610 IF Z$="C" OR Z$="c" THEN DIM B$(5,5): GO SUB 2500
3620 IF Z$="1" OR Z$="0" THEN LET F$(CY,CX)=Z$: LET N=CY: LET
M=CX: GO SUB 1500: GO SUB 2000: GO SUB 25000
3630 PRINT AT 13,26: PAPER 1: CHR$(144+CODE X$-65)
3640 IF Z$="S" OR Z$="s" THEN SAVE "GRAPHICS" CODE USR "A",21*8
3650 BEEP .2,10: RETURN

4000 REM PRINT GRAPHIC
4010 PRINT AT 21,0: PAPER 2: FLASH 1: "PRINT GRAPHIC ONTO DISPLAY"
4020 INPUT AT 0,0: "ENTER VERTICAL COORDINATE 1-5": LINE Y$: IF
Y$>="1" AND Y$<="5" THEN LET Y=VAL Y$(1): GO TO 4040
4030 GO TO 4020

```

```

4040 INPUT AT 0,0: "ENTER HORIZONTAL COORDINATE A-E": LINE Y$:
IF Y$>="A" AND Y$<="E" THEN LET X=CODE Y$(1)-CODE "A":
GO TO 4060
4050 GO TO 4040
4060 PRINT AT 5+Y,23+X: CHR$(144+CODE X$-65)
4070 PRINT AT 21,0: "*****"
4080 LET B$(Y,X+1)=CHR$(144+CODE X$-65)
4090 RETURN
4500 REM NEW GRAPHIC
4510 PRINT AT 21,0: PAPER 2: FLASH 1: "SELECT NEW GRAPHIC
CHARACTER"
4520 INPUT "ENTER A LETTER A-Q": LINE V$: IF V$>="a" AND
V$<="q" THEN LET V$=CHR$(CODE V$-32)
4530 IF V$>="A" AND V$<="Q" THEN LET X$=V$(1): GO SUB 1000: GO
TO 4550
4540 GO TO 4520
4550 PRINT AT 13,21: PAPER 1;X$: " ->>"
4560 PRINT AT 21,0: "*****"
4570 RETURN
5000 REM ARROWS
5010 DATA 1,1,49,25,13,7,3,1,128,128,140,152,176,224,192,128,0,0,32,48,
24,12,6,255,255,6,12,24,48,32,0,0
5020 FOR N=0 TO 3: FOR M=0 TO 7
5030 READ A: POKE (USR CHR$(82+N))+M,A: NEXT M: NEXT N
5040 DIM A$(2,2)
5050 FOR N=1 TO 2: FOR M=1 TO 2
5060 LET A$(N,M)=CHR$(158+2*N+M)
5070 NEXT M: NEXT N
5080 RETURN

```

Patterns

This program, I will admit, is pretty useless but none the less, I hope you will agree, very attractive. It arose from some doodlings with the computer and consists of two routines that draw interesting screens. The first routine which starts at line 50 (type **RUN 50**) draws a rainbow, and is the fastest rainbow drawing program that I could write.

The second routine (**RUN 1700**) draws interference patterns, and although I stumbled on it largely by accident, I was so impressed that I left it running in my living room, with the lights down low, all evening. I offer it to you in the hope that you may derive similar pleasure and fascination from it.

```
50 REM rainbow
100 BORDER 0: INK 7: PAPER 0: RESTORE 1000: FOR m=0 TO 7
110 READ a,b,c: FOR n=0 TO 11*(m<7): PLOT PAPER b: INK c: INVERSE
    a:0,n+12*m: DRAW PAPER b: INK c: INVERSE a:255,0,-1.5: IF ABS
    ((m/2)-INT (m/2))>.1 THEN LET n=n+7
120 BEEP .01,30: NEXT n: NEXT m
130 PAPER 0: INK 7: INVERSE 0
140 STOP
1000 DATA 0,0,2,1,6,2,0,6,4,1,5,4,0,5,1,1,3,1,0,3,0,1,0,0
1700 REM interference
1705 BORDER 0: INK 7: PAPER 0: BRIGHT 1: CLS: OVER 1
1710 LET DX=8*2↑(1+INT (RND*3)): LET X1=256/DX: LET C=5+INT
    (RND*2): FOR G=0 TO 1: INK C: IF G THEN INK 8-C
1720 FOR K=0 TO X1-1: FOR N=0 TO 1: FOR X=0 TO DX-1: PLOT
    DX*K,175*N: DRAW X,175-350*N: PLOT DX*(K+1)-1,175*N: DRAW
    -X,175-350*N: NEXT X: NEXT N: NEXT K
1730 NEXT G: GO TO 1700
```

About the Author

Richard Altwasser

Born in Worcester in 1957, Richard Altwasser gained an exhibition and honours degree in engineering at Trinity College Cambridge. He worked for 18 months on micro-based automation in Worcester before joining Sinclair in September 1980. In 1981 after the launch of the ZX81 he was made responsible for computer research, a task which led to the development of the hardware for the ZX Spectrum. He has also been writing software for the ZX81 and his book the 'Cambridge Collection' has sold more than 30,000 copies.

After the launch of the ZX Spectrum in May 1982, he left Sinclair to devote more time to writing and marketing software.