
HERE IT IS.....
MORE THAN FIFTY PROGRAMS FOR THE
SEGA
SC-3000 !!
GRANDSTAND
SEGA
MICHAEL HOWARD

MORE THAN FIFTY PROGRAMS FOR THE SEGA SC-3000

BY
MICHAEL HOWARD

P.O. BOX 2353, AUCKLAND, NEW ZEALAND

Author's Introduction

Typing in a computer program is like opening an unknown door. You do not know until you actually open the door — or, in this case, run the program — what experience is waiting for you. Of course, there is a little sign on the door which gives you some indication of what lurks behind the door, but you still don't really know!

The programs or doors, in this book range from destroying aliens to creating Purple People Eaters and from a game of Noughts & Crosses you play against the Sega, to quadratic and linear regression! Even statistics, 3 dimensional graphics, machine code routines, and full adventure games are included, and that is but the tip of the iceberg ... as you will find out!

The object of the book is actually to forward your programming skills, as many ideas are established within these covers. When you enter each program think of what you are entering and before you can say "Merlin-the-cat" you will be an ace programmer!

Whatever you find behind each door, I guarantee you won't be disappointed.

Happy progging!

MICHAEL HOWARD
AUCKLAND

1985

Graphics

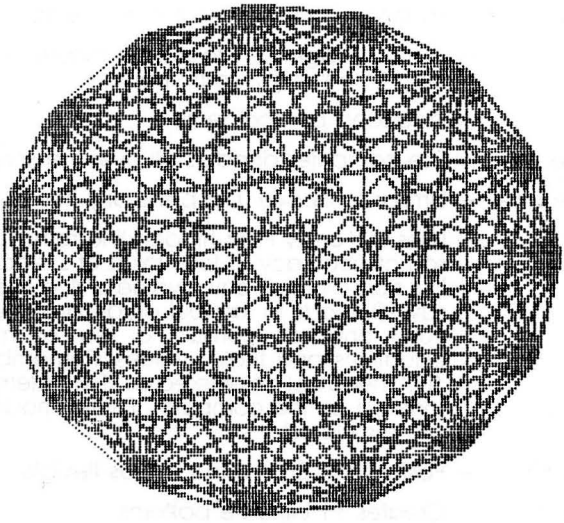
No book on the SC-3000 would be complete without a smattering of graphics programs. So here is a mob of them.

Notes:

The Cats Eye	— Try altering the value of A in Line 10.
Amazing	— Uses "intelligence" to create mazes. It takes a long time but it's worth the wait.
Starburst	— Produced by accident!
Kaleidoscope	— The computer doodles all over the screen.
Line Kaleidoscope	— Similar to above, but press any key to restart.
3D Graphics #1	— Uses the array P for hidden line removal. S is a sort of accuracy, a low value of s leads to fine but slow pictures, say a value of 4; a high value, say 20, leads to fast but crude drawings. Lines 1000 onwards contain a few examples of equations and their S-values. Remember Lines 1000 need not be entered as REM statements are ignored by the computer. Printer modifications are also listed.
3D Graphics #2	— Faster than 3D #1, but not as flexible.
More Graphics	— Creates 11 intricate patterns.
Boxes & Cubes	— Creates random sized, random positioned cubes all over the screen until you press a key.
Waves	— Creates fascinating patterns which can be altered at the touch of a button!
Mind's Eye	— One of my favourites!
Gateway	— Makes a pattern similar to looking down a tunnel.
Star of Ralthuz	— Named after a Dungeons & Dragons character of mine. The program signifies the growth of a wizard, from a slow, long upbringing to full splendour. This program is connected to "Lament of the Wind Wizard", but I won't say too much just yet!

The Cat's Eye

```
10 A=15:DIMA(A,2):C=1:SCREEN2,2:CLS
20 FORB=0TO359STEP360/A
30 A(C,1)=128+COS(B/180*PI)*87
40 A(C,2)=96+SIN(B/180*PI)*95
50 C=C+1:NEXT
60 FORB=1TOA:FORC=1TOA:LINE(A(B,1),A(B,2
))-(A(C,1),A(C,2)):NEXTC,B
```



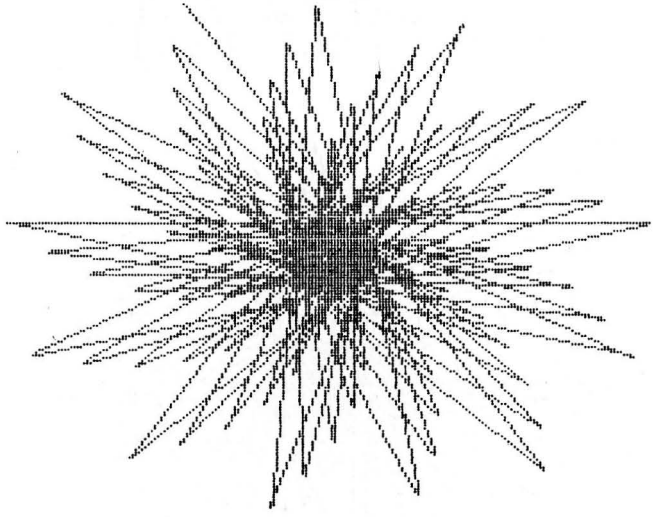
Amazing

```
10 SCREEN 2,2:COLOR15,4,,4:CLS:A$="F0F0F
0F000000000":PATTERNC#255,A$:PATTERNS#0,
A$
20 DIMA$(60,40)
30 FORA=1TO60:A$(A,1)="2":A$(A,2)="2":A$
(A,39)="2":A$(A,40)="2":NEXTA
40 FORA=1TO40:A$(1,A)="2":A$(2,A)="2":A$
(59,A)="2":A$(60,A)="2":NEXTA
50 X=4:Y=4+2*INT(RND(8)*18):CURSOR(X*4)+
6,Y*4:PRINTCHR$(255)
60 A$(X,Y)="2"
70 IFA$(X+2,Y)<>""ANDA$(X-2,Y)<>""ANDA$(
X,Y+2)<>""ANDA$(X,Y-2)<>""THEN200
80 GOSUB260
90 R=INT(RND(8)*4)
100 C=X+2*-(R=0)-2*-(R=1)
110 D=Y+2*-(R=2)-2*-(R=3)
120 IFA$(C,D)<>""THEN80
130 E=(C+X)/2
140 F=(D+Y)/2
150 A$(C,D)="1"
160 A$(E,F)="1"
170 SPRITE0,((4*E)+10,(4*F)-1),0,8:CURSO
R(4*E)+10,4*F:PRINTCHR$(255)
180 X=C:Y=D
```

```
190 GOTO70
200 A$(X,Y)="2":GOSUB260
210 IFA$(X+1,Y)="1"THENA$(X+1,Y)="2":X=X
+2:GOTO70
220 IFA$(X-1,Y)="1"THENA$(X-1,Y)="2":X=X
-2:GOTO70
230 IFA$(X,Y+1)="1"THENA$(X,Y+1)="2":Y=Y
+2:GOTO70
240 IFA$(X,Y-1)="1"THENA$(X,Y-1)="2":Y=Y
-2:GOTO70
250 FORI=0TO1000:BEEP:NEXT
260 SPRITE0,((4*X)+10,(4*Y)-1),0,8:CURSO
R(4*X)+10,4*Y:PRINTCHR$(255):RETURN
```

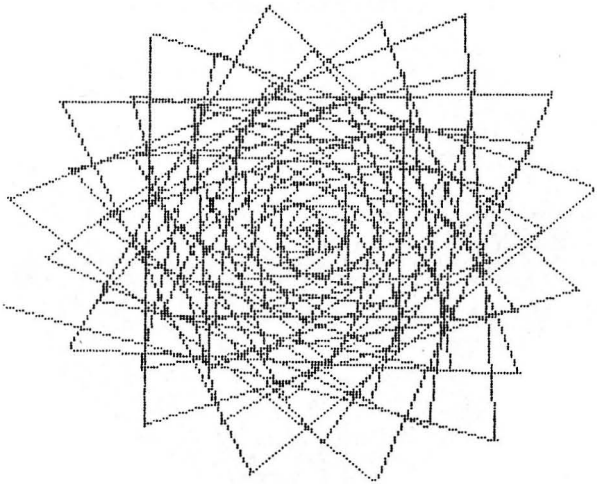
Star Burst

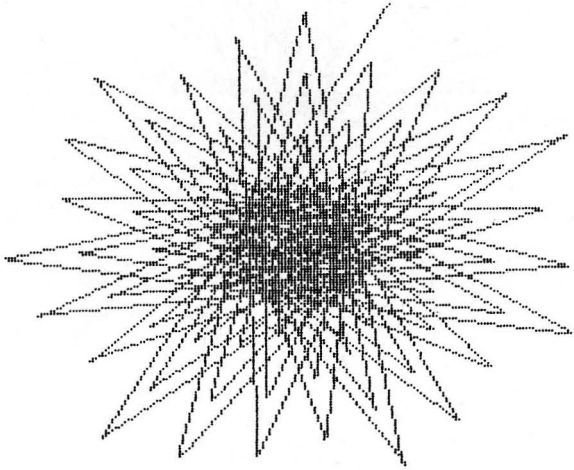
```
10 SCREEN 2,2:CLS
20 V=.8+3*RND(8)
30 PSET(128,96)
40 FORTH=0TO500*VSTEPV
50 R=TH/V:X=128+R*COS(TH):Y=95+R*SIN(TH)
60 IFY>191ORY<0THENFORT=1TO1000:NEXTT:GO
TO10
70 LINE-(X,Y)
80 NEXTTH
```



Star Burst... Printer/Plotter

```
10 LPRINTCHR$(18);"M0,0"
20 V=.8+3*RND(8)
30 LPRINT"M128,96"
40 FORTH=0TO500*VSTEPV
50 R=TH/V:X=128+R*COS(TH):Y=95+R*SIN(TH)
60 IFY>191ORY<0THENFORT=1TO1000:NEXTT:GO
TO10
70 LPRINT"D"+STR$(X)+","+STR$(Y)
80 NEXTTH
```



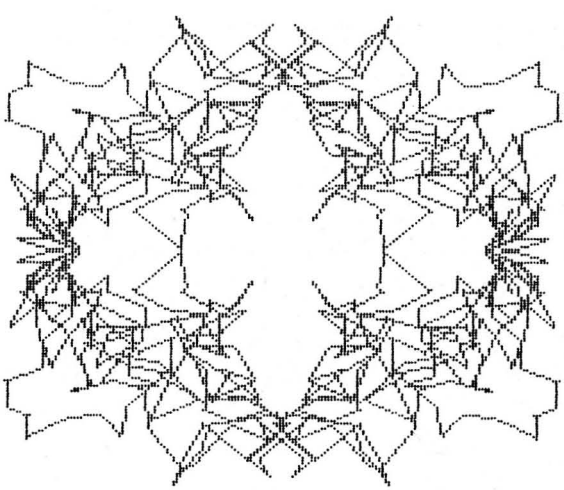


Kaleidoscope

```
10  SCREEN  2,2:CLS:COLOR8
20  X=128:Y=96
30  A=INT(RND(8)*6)+1:ONAGOSUB80,90,100,1
10,120,140
40  IFX<0ORX>255THEN30
50  PSET(X,Y):PSET(255-X,Y):PSET(255-X,19
1-Y):PSET(X,191-Y)
60  IFY<191ANDY>0THEN30
70  FORI=0TO500:NEXTI:GOTO10
80  X=X+1:Y=Y+1:RETURN
90  X=X-1:Y=Y+1:RETURN
100  X=X-1:RETURN
110  X=X+1:RETURN
120  IFRND(8)>.8THENRETURN
130  X=X+1:Y=Y-1:RETURN
140  IFRND(8)>.8THENRETURN
150  X=X-1:Y=Y-1:RETURN
```

Line Kaleidoscope

```
10  SCREEN  2,2:COLOR15,1,(0,0)-(1,1),1:CL
S
20  DEFFNR(X)=INT(RND(8)*X)
30  P=FNR(255):M=FNR(191)
40  P1=FNR(30)-15:M1=FNR(30)-15
50  IFP+P1<0ORP+P1>255ORM+M1<0ORN+M1>191T
HEN40
60  LINE(P,M)-(P+P1,M+M1)
70  LINE(255-P,191-M)-(255-(P+P1),191-(M+
M1))
80  LINE(255-P,M)-(255-(P+P1),M+M1)
90  LINE(P,191-M)-(P+P1,191-(M+M1))
100  P=P+P1:M=M+M1
110  IFINKEY$=""THEN40
120  GOTO10
```

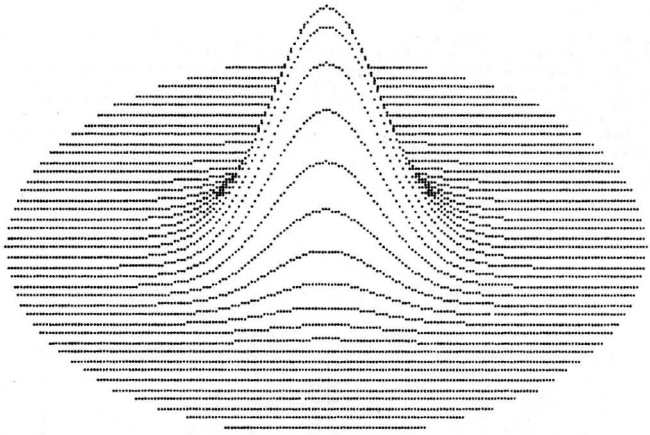


3D Graphics #1

```
10 SCREEN 2,2:CLS
20 S=10
30 DIMP(250,2)
40 FORF=1TO250:P(F,2)=255:IFF>140THENP(F
,2)=F
50 NEXT
60 FORF=-50TO50STEPS
70 A=F:B=50-ABS(F)
80 FORG=-70TO70
90 C=70-ABS(G)
100 GOSUB180
110 NEXTG
120 FORA=F+1TOF+S-1
130 B=50-ABS(A)
140 FORG=-70TO70STEPS
150 C=70-ABS(G)
160 GOSUB180
170 NEXTG,A,F:END
180 T=B*C/800
190 R=A+G+121
200 T=-T*T/20
210 T=INT(80+A-T*80)
220 IFF=-50THENP(R,2)=T
230 IFT<=P(R,1)THEN290
240 P(R,1)=T
250 IFT<0THENT=0
260 IFT>191THENT=191
270 PSET(R+5,191-T)
280 RETURN
290 IFT>=P(R,2)THENRETURN
300 P(R,2)=T:GOTO250
1000 REM
1010 REM              S=5
1020 REM T=LOG(ABS(COS(T)))/10
1030 REM T=-SIN(T)*(1-COS(T))/2
1040 REM
1050 REM              S=4
1060 REM T= EXP(T)/80
1070 REM T=-EXP(T)/80
1080 REM T=COS(T*4)/6
1090 REM
1100 REM              S=10
1110 REM T=(SIN(T*3)+COS(T))/4
1120 REM T=-T*T/20
1130 REM
1140 REM S IS IN LINE 20
1150 REM
1160 REM T IS IN LINE 200
```

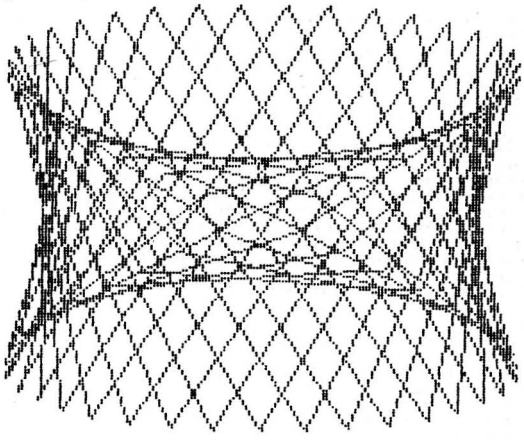
3D Graphics #2

```
10 SCREEN 2,2:CLS
20 DEFFNA(Z)=90*EXP(-Z*Z/600)
30 K=5
40 FORX=-100TO100
50 L=0
60 F=1
70 Z1=0
80 Y1=K*INT(SQR(10000-X*X)/K)
90 FORY=Y1TO-Y1STEP-K
100 Z=INT(80+FNA(SQR(X*X+Y*Y))-.707*Y)
110 IFZ<LTHEN160
120 L=Z
130 PSET(X+128,191-Z),8
140 IFF=0THENZ1=Z
150 F=0
160 NEXTY,X
1000 REM ALSO A(Z)=5*COS(Z/5)
```



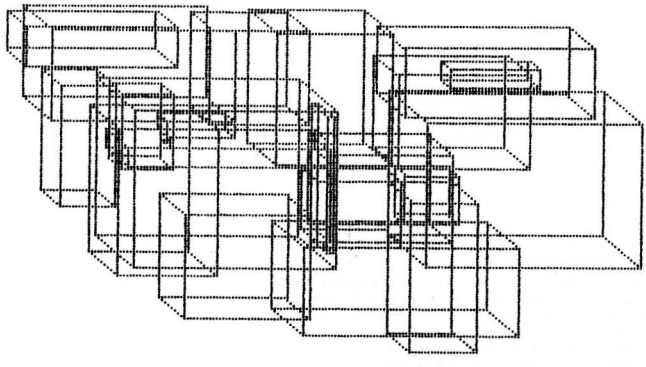
More Graphics...

```
10 SCREEN 2,2:CLS
20 FORR=1TO11
30 READP,Q,K,S,B
40 FORA=1TOB:X=K*SIN(P*A):Y=K*COS(Q*A)
50 IFA=1THENPSET(X+135,Y+90),8
60 LINE-(X+135,Y+90)
70 NEXTA
80 BEEP:BEEP:BEEP
90 IFINKEY$=""THEN90
100 CLS
110 NEXTR
120 DATA7,30,90,1,200,20,10,80,1,72,2,4,
90,1,100,3,4,90,1,200,4,3,90,.5,200,4,3,
90,.2,200,27.8,139,80,.208,200,30,20,80,
1,78,40,20,80,1,100,30,40,80,1,100,10,30
,80,1,100
```



Boxes and Cubes

```
10 SCREEN 2,2:CLS
20 COLORRND(8)*14:FORF=1TO100
30 X=RND(8)*150:Y=RND(8)*90:A=RND(8)*70:
B=RND(8)*60:C=(A+B)/10
40 LINE(X,Y)-(X+A,Y+B),,B:LINE-(A+X+C,B+
Y+C):LINE-(C+X,Y+C),,B
50 LINE-(X,Y):LINE(X,Y+B)-(X+C,Y+B+C):LI
NE(X+A,Y)-(X+A+C,Y+C)
60 IFINKEY$<>""THEN10
70 NEXTF:GOTO10
```

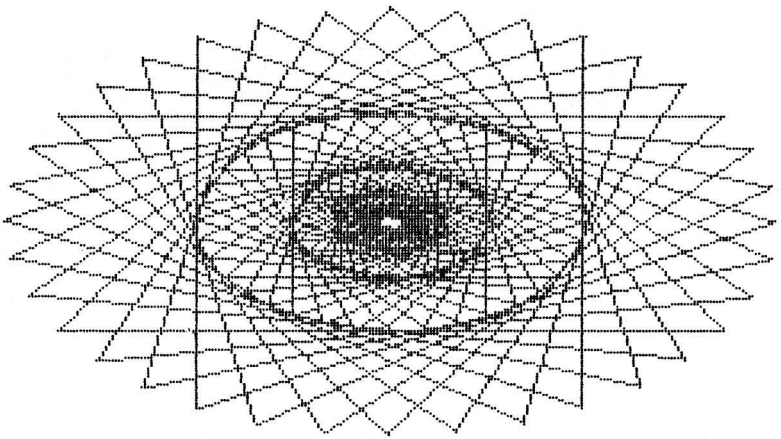


Waves

```
10 DEFFNR(X)=INT(RND(8)*X)
20 SCREEN 2,2:CLS
30 X=FNR(255):Y=FNR(191)
40 L=FNR(255):M=FNR(191)
50 U=15:V=7
60 GOSUB220
70 FORQ=2TO16:FORG=1TO150
80 N=N-1
90 IFN=0THENGOSUB220
100 LINE(X,Y)-(L,M),Q
110 IFINKEY$<>""THENGOSUB230
120 IFX+A>255ORX+A<0THENA=-A
130 IFY+B>191ORY+B<0THENB=-B
140 IFL+C>255ORL+C<0THENC=-C
150 IFM+D>191ORM+D<0THEND=-D
160 X=X+A:Y=Y+B:L=L+C:M=M+D
170 NEXT G
180 FORI=0TO500:NEXTI
190 CLS
200 NEXT
210 IFINKEY$<>""THEN230
220 A=FNR(U)=V
230 B=FNR(U)-V
240 C=FNR(U)=V
250 D=FNR(U)-V
260 N=FNR(20)+10
270 RETURN
```

Minds Eye

```
10 SCREEN 2,2:CLS:DIMA(36),B(36)
20 L=120:J=80
30 FORH=1TO5:FORN=1TO36
40 K=N/18*PI
50 A(N)=128+L*SIN(K):B(N)=88+J*COS(K)
60 PSET(A(N),B(N)),4
70 NEXTN
80 FORN=1TO36:M=N+12
90 IFM>36THENM=M-36
100 LINE(A(N),B(N))-(A(M),B(M)),8
110 NEXTN
120 L=L/2:J=J/2
130 NEXTH
```

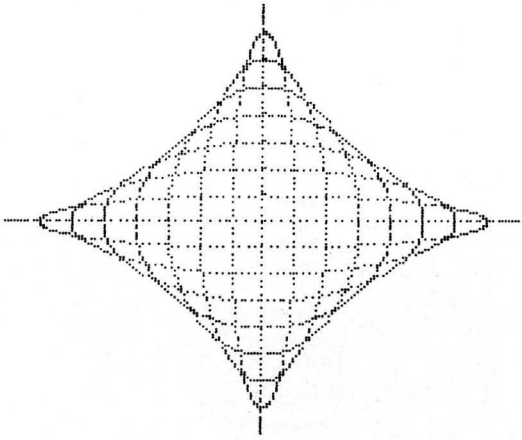


Gateway

```
10 SCREEN 2,2:CLS:DEFFNR(J)=INT(RND(8)*J
)
20 P=FNR(255):Q=FNR(191):S=FNR(9)+1:C=FN
R(14)
30 Z=1
40 FORA=1TO2
50 FORX=0TO254STEPS
60 LINE(P,Q)-(128-127*Z+X*Z,96-Z*95),C
70 NEXTX
80 FORY=0TO191STEPS
90 LINE(P,Q)-(128+127*Z,96+Z*-95+(Y*Z)),
C
100 NEXTY
110 Z=-Z
120 NEXTA
130 IFINKEY$=""THEN130
140 GOTO10
```

Star of Ralthuz

```
10 SCREEN 2,2:CLS
20 X=0:Y=80
30 FORN=0TO 2*PI STEP PI/90
40 PSET(128+X*SIN(N),96+Y*COS(N))
50 NEXTN
60 X=X+10:Y=Y-10
70 IFY=-10THENEND
80 GOTO30
```



Compound Interest

This program calculates compound interest on any sum of money invested at any interest rate for any number of years — so you can work out how many decades it'll take to amass $1,000,000! That's why I wrote it, but I gave up hope of becoming a millionaire when I was predicting into the middle of the 21st century!!

For example, you wish to invest $1,000 in one of two banks:

Bank 1 offers 12.5% interest, calculated every 6 months.
Bank 2 offers 12.8% interest, calculated annually.

Which bank should you choose?

Run the program and input:

Interest %	=	12.5%	
Interval	=	6	
Deposit	=	1000	Bank 1
Regular Deposit	=	0	
Interval between dep.	=	0	

The result after 10 years is $383.90 interest.

Rerun and enter 12.8, 12, 1000, 0, 0.

The result after 10 years is $378.40 interest. So Bank 1 is best.

You can also enter regular deposits and regular interval deposits.

Notes:

10-120 Enter various variables.
A = interest rate.
B = when interest computed (in months).
C = initial investment.
D = subsequent investments/withdrawals.
E = interval between investment/withdrawals.

130-200 Choices of whether to see interest calculated year by year, or total after a number of years, or stop program or do new problem.

210-340 Main calculating loop.

350-440 Show table of interest etc. and after another 10 years. Select how many years you want done in B mode. Goto 210.

450-490 Results of B mode. Total money invested and total profit after EC years. Goto 130.

Compound Interest

```
10 CLS:PRINT"COMPOUND INTEREST":PRINT
20 INPUT"ENTER THE % RATE OF INTEREST ";
A
30 A=A/100
40 PRINT
50 INPUT"INPUT INTERVAL BETWEEN INTEREST
      CALCULATIONS IN MONTHS ";B
60 PRINT
70 INPUT"INPUT ORIGINAL DEPOSIT $";C
80 PRINT
90 INPUT"INPUT REGULAR DEPOSIT $";D
100 PRINT
110 INPUT"INPUT INTERVAL BETWEEN DEPOSIT
S ";E
120 IFE=0THENE=13
130 CLS
140 PRINT"YOU MAY...":PRINT:PRINT"A)SEE
GROWTH OF MONEY OVER A FEW YRS",,,"B)SEE
 SITUATION AFTER A FEW YRS",,,"C)DO NEW
CALCULATIONS",,,"D)STOP PROGRAM"
150 A$=INKEY$
160 IFA$="A"THEN210
170 IFA$="B"THEN420
180 IFA$="C"THEN10
190 IFA$="D"THENPRINT"OKAY.....":END
200 GOTO150
210 CLS:PRINT"GROWTH OF CAPITAL":PRINT:P
RINT"INITIAL DEPOSIT $";C:PRINT:PRINT"AT
";A*100;"% CALCULATED EACH";B;" MTS"
220 IFD>0THENPRINT:PRINT"DEPOSIT OF $";D
;" EVERY";E;" MONTHS"
230 PRINT
240 Y=0:DP=C:TM=0
250 IFA$="A"THENEC=Y+10:PRINT"YEAR DEPOS
IT INTEREST NEW DEPOSIT"
260 FORF=Y+1TOEC:ND=0:RS=0
270 FORG=1TO12:TM=TM+1
280 IFTM/B=INT(TM/B)THENRS=RS+(DP+ND+RS)
*A*B/12
290 IFTM/E=INT(TM/E)THENND=ND+D
300 NEXTG
310 IFA$="A"THENPRINTF;TAB(6);INT(DP*10+
.5)/10;TAB(14);INT(RS*10+.5)/10;TAB(25);
ND
320 DP=DP+ND+RS
330 NEXT F
340 IFA$="B"THEN450
350 PRINT:PRINT"FURTHER 10 YRS? (Y/N)"
360 IFINKEY$="N"THEN130
370 IFINKEY$="Y"THEN20
380 GOTO360
390 Y=Y+10
400 CLS
410 GOTO250
420 PRINT"AFTER HOW MANY YRS WOULD YOU L
IKE TO  SEE THE INVESTMENT...?"
430 INPUTEC
440 GOTO210
450 PRINT"AFTER";EC;" YEARS..":PRINT:PRI
NT"TOTAL MONEY INVESTED $";INT(DP*100+.5
)/100:PRINT
460 PRINT"TOTAL PROFIT $";DP-C-D*E/12*EC
470 PRINT:PRINT:PRINT"PRESS A KEY TO RET
URN"
480 IFINKEY$=""THEN480
490 GOTO130
```

Forth / RPN Emulator

Forth is a language slightly similar to BASIC, that is used extensively in scientific institutions for controlling experiments. The massive radio telescope at Jodrell Bank in Cheshire, England (owned and run by the University of Manchester) is controlled by Forth. The main advantage of Forth is it is very, very fast, typically 40-50 times quicker than BASIC. This implementation of Forth allows you to mess around a bit with Forth and Reverse Polish Notation — it is by no means complete as it only lets you do simple work.

Okay, so how does Forth and RPN work? (Actually Forth works in RPN!) Image the following:

```
2 + 3 = 5
```

This is very simple, but in RPN it is a little different. It is:

```
2 3 + .          The "." produces 5.
```

This is how it works. Forth and RPN work on a system of manipulating an area of memory called a "stack". A stack is just an area of memory which stores numbers. When a stack is created it is empty, thus it contains just zeros. Let's say our stack can hold a maximum of 5 numbers, so it looks like this:

```
0    ← top of stack
0
0
0
0    ← bottom of stack
```

Every time you enter a number, that number is "pushed" onto the stack. In our case we enter 2 and 3, so the stack goes:

```
2                3
0                2
0                0
0                0
0                0
2 entered        3 entered
```

Every time a number is entered the stack is pushed down and the number is placed on top. So now we have entered 2 and 3, our next command is + (remember 2 3 + .). When "+" is entered the top two elements on the stack are added together, the stack is moved up and the result is placed on the top of the stack. When "." is pressed the top of the stack is displayed on the screen. In my little program the stack (which can hold 11 elements) is always in view so "." is not used. So to recap for the sum 2 + 3, which is 2 3 + . in RPN, the stack goes like this:

```
                0  2  3  5  0
                0  0  2  0  0
Stack           0  0  0  0  0
                0  0  0  0  0
                0  0  0  0  0
```

Expression 2 3 + . — When "." is pressed whatever is on the top of the stack is removed and printed on the screen.

See, easy!

+, −, * and / are all supported viz:

	0	5	11	16	5	3	2	8	3	24
	0	0	5	0	16	5	16	0	8	0
Stack	0	0	0	0	0	16	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0

Expression 5 11 + 5 3 − / 3 *

This is the same as ([5 + 11] / [5 − 3]) * 3

Another example — let's say you want to find the average of 17, 42, 69, 19 and 103. In the program (when running it) just enter:

17 42 69 19 103 + + + + 5 / and out comes your answer! Of course you can do much more than this.

In Forth there is also a few commands for manipulating the stack. These are DUP, DROP, SWAP and OVER. They allow the repetition of data near the top of the stack. DUP — duplicate top number, DROP — discard top number, SWAP — swap top 2 numbers over, and OVER — copy 2nd number on stack to the top OVER the original top number. See below:

Initial Stack	DUP	DROP	SWAP	OVER
30	30	20	20	20
20	30	10	30	30
10	20	0	10	20
0	10	0	0	10
0	0	0	0	0

Note — The examples shown above do not follow one another — each is based on the initial stack.

There is an example:

	0	2	2	2	2	4	8	5	40	2	3	6	34	9	43
	0	0	2	2	2	2	2	8	2	40	2	40	0	34	0
Stack	0	0	0	2	2	2	0	2	0	0	40	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Expression 2 DUP DUP DUP * * 5 * SWAP 3 * − 9 +

This is the same as $5x^2 - 3x + 9$ where $x = 2$.

You can also cube numbers. Let's say you want to know what 4^3 is. You would enter 4 DUP DUP * * — as you can see it is fairly easy to use once.

In actual Forth you can define functions (you can't in this). Remember that cube numberthing? Well to define it you would do:

```
:          cube    DUP DUP * *          .          ;
↑          ↑                            ↑          ↑
start define  name   actual function   display    end define
                                        result
```

To use it you would type "4 cube" and the result would be printed. Forth has many, many more commands than this, but this program is only meant as a demonstration, and hopefully show you that there is more to computers than BASIC.

Notes:

10-	Set up. ST — stack, SP — stack pointer holds size of stack, B$ — commands used.
20-30	Print title and stack.
40	Enter command S reruns program and is NOT standard Forth.
50-150	Handle commands & numbers.
160	Don't recognise command.
170	Goto 20.
180-210	Add to commands used. Make sure SP is in range 0-10.
1000	Move stack down.
1010	Move stack up.

```
Forth/RPN Emulator

10 ERASE:DIMST(10):SP=0:B$=""
20 CLS:PRINT"Forth/RPN Emulator M.Howard
",,,
30 PRINT"Stack",,,,:FORA=0TO10:PRINTST(A
):NEXT:PRINT,,,,B$
40 INPUT"";A$
50 IFA$="S"THEN10
60 IFA$="DUP"THENGOSUB1000:ST(0)=ST(1):G
OTO180
70 IFA$="DROP"THENGOSUB1010:GOTO180
80 IFA$="SWAP"THENT=ST(0):ST(0)=ST(1):ST
(1)=T:GOTO180
90 IFA$="OVER"THENGOSUB1000:ST(0)=ST(2):
GOTO180
100 IFA$="+"THENT=ST(1)+ST(0):GOSUB1010:
ST(0)=T:GOTO180
110 IFA$="*"THENT=ST(1)*ST(0):GOSUB1010:
ST(0)=T:GOTO180
120 IFA$="-"THENT=ST(1)-ST(0):GOSUB1010:
ST(0)=T:GOTO180
130 IFA$="/"THENT=ST(1)/ST(0):GOSUB1010:
ST(0)=T:GOTO180
140 FORA=1TOLEN(A$)
150 IFLEFT$(A$,A)>="0"ANDLEFT$(A$,A)<="9
"THENNEXT:GOSUB1000:ST(0)=VAL(A$):A$=STR
$(VAL(A$)):GOTO180
160 BEEP1:CURSOR14,2:PRINT"Command not k
nown":FORA=0TO200:NEXT:BEEP0:GOTO20
```

```
170 GOTO20
180 B$=B$+" "+A$
190 IFSP>10THENSP=10
200 IFSP<0THENSP=0
210 GOTO20
1000 FORA=9TO0STEP-1:ST(A+1)=ST(A):NEXT:
SP=SP+1:RETURN
1010 FORA=0TO9:ST(A)=ST(A+1):NEXT:SP=SP-
1:RETURN
```

Wally — Simons' Brother-in-Law

The following story is absolutely true (well, some names have been changed to protect the not-so-innocent!). A few years ago a little electronic game called Simon got married, exactly to whom no-one really knows but it is thought that Ms Pac Man is the prime suspect. One thing that is known is that the fruits of their love was a little chappie called Pocket Simon. It is rumoured that Ms Pac Man had a brother. Well, let me tell you — she has! — and his name is Wally. One night Wally got sick of hearing an argument between Simon and his wife, so he sneaked out the back door and hid in the memory of a Sega computer. There he remained for six months, watching various programs, but he got lonely (aw ... !). One night (or early morning), I was doing some research for this book when the program I was writing went bonkers and disappeared! Perplexed (and vexed!) I listed the program and found a program calling itself "Wally", Lines 1-9 (now deleted, as some of the data is personal!) contained the above story and a note saying, "Now you have found me, please don't tell Simon or my sister where I am!" Well, I decided not to, but I did get him to give me a photo of himself, which has been faithfully reproduced below.

Okay, now you can stop crying — here is how the game is played. It is exactly the same as Simon. Wally sets a pattern of sound and colour and you must follow it. Simple ... NO! At the start it is fairly straightforward and simple, with only a few steps to remember, but it gets fiendishly hard near the end.

The computer (er ... Wally!) draws 4 segments of circles each of differing colour. When the pattern is generated, you must respond by pressing the appropriate cursor key (the ones with the arrows on them ... see them? ... good!), e.g. ... the top cursor key represents the blue semi-circle. Wally will prompt you to respond.

Notes: (This is serious now!)

10-70 Data for position and colour of sprites arranged as x, y, colour. Set up screen.

80 Generate random pattern.

90 Start of main loop, Q reads through pattern contained in Z-array.

100 D is a delay. As the game goes on the speed increases.

110-140 Display pattern. Read data from Line 10, display sprite, thus making each segment "light-up". Make a sound proportional to which segment "lights up". Delay. Switch off sprite and sound.

150 Tell the being at the keyboard to get his act together!

160 Loop which counts from 0 to Q. Q don't forget, contains the length of the pattern so far.

170 Wait for response from keyboard. If it is not a cursor (look at the bottom of page 19 of the operator's manual), then go back to Line 170.

180 Turn the value of the key pushed into a number (1-9).

190 Display your choice.

200 Check to see if your response is right.

210 Make a chirp.

220 Wait for you to take your finger off the key.

230 Switch off sound and sprite.

240 Loop end.

250-260 Delete "Your reply". Small pause. Next loop (Q).

270-280 If you complete the whole pattern (most unlikely!) Wally congratulates you.

290-300 Press a key to rerun.

310-320 Ha, ha ... you made a mistake. Delay.

330-370 Play back pattern.

```
Wally... Simon's Brother-in-Law

10 DATA116,65,7,140,90,9,116,115,3,94,90
,11
20 SCREEN 2,2:CLS:PATTERN$#0,"FFFFFFFFFF
FFFFFF"
30 COLOR13:CURSOR97,10:PRINTCHR$(17);"SI
MON";CHR$(16):J=40
40 CIRCLE(120,90),J,4,1,-.625,-.875,BF
50 CIRCLE(125,95),J,6,1,-.875,-.125,BF
60 CIRCLE(120,100),J,12,1,-.125,-.375,BF
70 CIRCLE(115,95),J,10,1,-.375,-.625,BF
80 ERASE:DIMZ(20):FORA=0TO20:Z(A)=INT(RN
```

```
D(8)*4)+1:NEXT
90 FORQ=0TO20
100 D=90-Q*4
110 FORI=0TOQ:RESTORE:FORJ=1TOZ(I):READX
,Y,C:NEXT:SPRITE0,(X,Y),0,C
120 SOUND1,500+(Z(I)*75),15
130 FORDE=0TOD:NEXT:SPRITE0,,,0:SOUND0
140 NEXTI
150 COLOR1:CURSOR84,155:PRINT"You're rep
ly"
160 FORI=0TOQ
170 A$=INKEY$:IFA$<CHR$(28)ORA$>CHR$(31)
THEN170
180 RE=(1ANDA$=CHR$(30))-(2ANDA$=CHR$(28
))-(3ANDA$=CHR$(31))-(4ANDA$=CHR$(29)):R
E=ABS(RE)
190 RESTORE:FORJ=1TORE:READX,Y,C:NEXTJ:S
PRITE0,(X,Y),0,C
200 IFRE<>Z(I)THEN310
210 SOUND1,500+(RE*75),15
220 IFINKEY$<>""THEN210
230 SOUND1,,9:SOUND0:SPRITE0,,,0
240 NEXTI
250 BLINE(84,155)-(156,163),,BF
260 FORI=0TO300:NEXTI:NEXTQ
270 CURSOR20,155:PRINT"You completed the
 entire set..!"
280 FORA=110TO1000STEP20:SOUND1,A,15:NEX
T:SOUND0
290 IFINKEY$=""THEN290
300 GOTO 10
310 SOUND1,110,13:CURSOR60,165:PRINT"Oh
dear you blew it....!":CURSOR60,175:PRIN
T"You lasted";Q;" rounds"
320 FORD=0TO200:NEXTD
330 A=0
340 RESTORE:FORI=1TOZ(A):READX,Y,C:NEXTI
350 SPRITE0,(X,Y),0,C:SOUND1,500+(Z(A)*7
5),15
360 FORD=0TO99:NEXTD:SOUND0:SPRITE0,,0,0
370 IFA<QTHENA=A+1:GOTO340
380 GOTO290
```

Rock, Scissors, Tissue

In this very old game scissors cut tissue, tissue covers rock, and rock blunts scissors. It is a game of psychology. In theory it is impossible to predict what an opponent will do next, but a computer could quite easily try to find a pattern, predict what your next move might be, then give a response which would beat you!

Information about moves is stored in the array A. In the array 1 = rock, 2 = scissors and 3 = tissue.

As rock (1) beats scissors (2), scissors (2) beats tissue (3), and tissue (3) beats rock (1), you can see a pattern emerging. If the computer's move is 1 more than yours (assuming 3 + 1 = 1), then the computer loses; therefore if the computer thinks that you are going to make a certain move, it adds 2 to that value and wins (again!).

Notes:

10	Data for elements.
20-70	Set up variables. Initial value of array A is random.
80-90	Searches through array to find most popular move of player.
110	Calculates computer's response.
130	Await R, S or T.
150	Convert R = 1, S = 2, T = 3.
170-200	Find who has won and increment winner's score by 1.
210-250	Print who wins and update scores, make a noise, press a key.
260	Select Rock, Scissors or Tissue routine.

```
Rock, Scissor, Tissue

10 DATARock,Scissors,Tissue
20 I=1:J=2:K=3:DIMT(2),A(3,3),P$(2)
30 X=RND(-1)
40 L=INT(RND(8)*K+I)
50 Y=INT(RND(8)*K+I)
60 A(L,Y)=I:A(Y,L)=I:P$(I)="Me":P$(J)="Y
ou"
70 SCREEN 1,1:CLS
80 N=I+-(A(Y,1)<=A(Y,J))
90 N=N+-(A(Y,I)<=A(Y,J))*(K-N)
100 L=Y
110 N=N-I+K*-(I=N)
120 PRINTP$(J);"  ";
130 Y$=INKEY$:IFY$<"R"ORY$>"T"THEN130
140 SOUND1,500,15:FORA=0TO10:NEXT:SOUND0
150 Y=ASC(Y$)-81:P=Y:GOSUB260:CURSOR6,0:
PRINTA$
160 PRINTP$(I);"  ";:P=N:GOSUB260:CURSOR6
,1:PRINTA$:SOUND1,110,15:FORA=0TO10:NEXT
:SOUND0
170 A(L,Y)=A(L,Y)+I
180 IFY=NTHEN70
190 W=Y-N+K*-(N>Y)
200 T(W)=T(W)+I
210 PRINTP$(W);"  WINS",,P$(I);T(I),P$(J)
;T(J)
220 A=500:IFW=1THENA=110
230 FORO=0TO3:SOUND1,A,15:FORU=0TO20:NEX
T:SOUND0:NEXT
240 IFINKEY$=""THEN240
250 GOTO70
260 RESTORE:FORA=1TOP:READA$:NEXT:RETURN
```

Lunar Lander

Due to a freak accident on board your Apollo Lunar Landing Module, you must land the beast manually. To slow down increase the thrust.

Notes:

10	Set variables.
20-50	Display status.
60	Enter thrust.
70	Make sure thrust is legitimate.
80-120	Do some working out.
130	Create a random sound.
140	Successful in landing!
160	Oh dear ... you mucked up! Make explosion sound.
170	If you crash you leave a crater, the size of which is proportional to the amount of fuel left.

```
Lunar Lander

10 CLS:V=40:A=1000:F=2500:S=1:U=0
20 CURSOR0,6:PRINT"ALTITUDE";A;"   "
30 CURSOR0,8:PRINT"SPEED    ";V;"  "
40 CURSOR0,3:PRINT"COMPUTER ENTRY";S
50 CURSOR0,10:PRINT"FUEL     ";F-2000-U;"
  "
60 CURSOR0,12:INPUT"THRUST  ";TH
70 IF TH<0 OR TH>50000THENSOUND1,110,15:
FORI=0TO100:NEXT:SOUND0:GOTO60
80 S=S+1
90 U=TH/50000*50:F=F-U:IFF-2000-U<=0THEN
160
100 V=V-((TH/F)-2):A=A-V
110 IFA<=0ANDV<5 THEN 140
120 IFA<=0 THEN160
130 SOUND1,2000+RND(8)*1000,5:GOTO20
140 CURSOR8,4:PRINT"CONGRATULATIONS YOU
MADE IT"
150 GOTO180
160 CURSOR18,3:PRINT"YOU CRASHED...HA..H
A":OUT&H7F,&HE4:FORE=&HF0TO&HFF:OUT&H7F,
E:FORI=0TO20:NEXTI,E
170 PRINT"AND LEFT A CRATER";INT(F/10);"
 MILES WIDE"
180 IFINKEY$=""THEN180
190 GOTO10
```

Spacies

This little program, silly though it may be, is actually quite addictive. The object is very simple — a little beastie (some say UFO) will start coming down from the top of the screen. It is your job as chief "blower-upper" to smash the-innocent-little-thingy-that-doesn't-mean-any-harm to bits! Good fun! Once you blast one, another comes along, but being a bit annoyed (that must be the understatement of the year!) he is quicker than his predecessor. You control your implement of destruction by using the P and Q keys to move, and the space bar to shoot. If you hold down the shift key, you double your speed. The game ends when a beastie lands.

Remember: "Support your local laser base, stomp on a Space Invader!"

Notes:

10-40	Set up screen, design laser base (sprite #0). X = original position of your base. S = speed of space invader.
50	Main loop. Governs "fall" of space invader.
60	Position base and spacie.
70	Handle keyboard entry. "Q" = -4, "P" = 4, "q" = -8, "p" = 8. Check if space is pressed.
80	Loop back.
90	You got sussed — baddies started to colonise your planet!!
100-110	Wait for key to be pressed. Rerun.
120	Laser blast and sound.
130	Check to see if a hit. If so, increase speed. Make an explosion. Next baddie.
140	Goto 80 (i.e. missed!).

```
Spacies!!!

10 SCREEN2,2:COLOR1,1,,1:CLS
20 PATTERNS#0,"109292BAFEFEBA82"
30 X=128:S=.5
40 V=INT(RND(8)*200)+20:PATTERNS#2,"7E81
A581A5BD817E"
50 FORW=0TO191STEPS
60 SPRITE2,(V,W),2,4:SPRITE0,(X,183),0,8
70 A$=INKEY$:X1=4*(A$="Q")-4*(A$="P")+8*
(A$="q")-8*(A$="p"):X=X+X1:IFA$="  "THEN1
20
80 NEXTW
```

```
90 COLOR4:CURSOR10,10:PRINT"They have 1a
nded...":CURSOR10,30:PRINT"Final Score:"
;S*100
100 IFINKEY$=""THEN100
110 GOTO10
120 SOUND3,8000:SOUND4,3,15:SOUND3,5000:
LINE(X+3,180)-(X+3,W+4),14:BLINE(X+3,180
)-(X+3,W+4):SOUND0
130 IFABS(X-V)<=4THENS=S+.25:PATTERNS#2,
"9152000300004A89":SOUND3,19000:FORB=15T
00STEP-.5:SPRITE2,,2,B:SOUND4,3,B:NEXTB:
GOTO40
140 GOTO80
```

Computer Chit-Chat

The following program should be carrying a health warning! It is likely to drive you round the twist, or split your sides (or both). It actually follows the laws of English to produce language such as:

The Lazy Nerd Eats Slowly On Top Of A Damaged Computer

or

A Big Book Reads Awfully

Notes:

10	Data for articles.
20	Data for nouns.
30	Data for adjectives.
40	Data for verbs.
50	Data for adverbs.
60	Data for prepositions.
70	A random number function.
80	A1 = number of articles, N1 = number of nouns, etc.
90-100	Set up arrays and read data into them.
110	S$ will hold final sentence.
120-130	Create and print S$.
140-150	Wait for a key to be pressed, create another sentence by rerunning.
1000-1030	Construct a noun phrase, by extracting a random article, selecting, optionally, a random adjective, then select a noun.

3000-3060 Select optionally an adverb, then call Line 4000 to supply a preposition.

4000-4020 Optionally select a preposition, then jump to 1000 to select a noun phrase, etc.

It is easy to alter the vocab. Just alter the appropriate data and don't forget to change Line 80 accordingly!

```
Computer Chit-Chat

10 DATAA,THE
20 DATACOMPUTER,PRINTER,PROGRAM,BUG,PROG
RAMMER,NERD,MONITOR,BOOK
30 DATABIG,POOR,TINY,LAZY,SHORT,MASSIVE,
CRAZY,DAMAGED
40 DATAWRITES,RUNS,DEBUGS,PRINTS,EATS,RE
ADS,CLIMBS
50 DATASLOWLY,FAST,QUICKLY,NICELY,AWFULL
Y,LAZILY
60 DATAON,IN,UNDER,ON TOP OF
70 DEFFNR(X)=INT(RND(8)*X)+1
80 A1=2:N1=8:D1=8:V1=7:B1=6:P1=4
90 RESTORE:DIMA$(A1),N$(N1),D$(D1),V$(V1
),B$(B1),P$(P1)
100 FORA=1TOA1:READA$(A):NEXT:FORA=1TON1
:READN$(A):NEXT:FORA=1TOD1:READD$(A):NEX
T:FORA=1TOV1:READV$(A):NEXT:FORA=1TOB1:R
EADB$(A):NEXT:FORA=1TOP1:READP$(A):NEXT
110 S$=""
120 GOSUB1000:GOSUB3000:GOSUB4000
130 PRINTS$
140 IFINKEY$=""THEN140
150 GOTO110
1000 A=FNR(A1):S$=S$+A$(A)+" "
1010 IFFNR(10)>5THENA=FNR(D1):S$=S$+D$(A
)+" "
1020 A=FNR(N1):S$=S$+N$(A)+" "
1030 RETURN
3000 A=FNR(V1):S$=S$+V$(A)+" "
3010 IFFNR(10)>5THENA=FNR(B1):S$=S$+B$(A
)+" ":GOSUB4000
3060 RETURN
4000 IFFNR(10)>5THENRETURN
4010 A=FNR(P1):S$=S$+P$(A)+" ":GOSUB1000
4020 RETURN
```

Chords & Music

The following tiny program will create random, but musical, music! It won't turn your micro into a rival for Beethoven or Billy Idol but it will surely beat my vocal abominations!!

When run, there is a short pause, then the sonic harmony will start and continue until you get a power cut!

Notes:

10-30 Data for chords.

40 Read data into array A.

50 Create music by transferring data from A.

60 Play it. To speed up the tones, decrease the value of the L-loop e.g. FOR L = 0 TO 0 ... will be very fast.

```
Chords and Music

10 DATA319,379,239,179,358,284,319,426,2
53,338,426,284,284,379,451,301,379,451,3
01,379,253
20 DATA358,426,268,319,402,268,301,402,4
78,301,358,451,268,338,451,253,338,402,4
02,319,230
30 DATA358,301,239,379,319,451,402,338,2
39,338,284,451
40 DIMA(16,7),Z(50,7):X=RND(-1):RESTORE:
FORI=0TO16:FORJ=0TO2:READA(I,J):A(I,J+3)
=A(I,J):NEXTJ:A(I,6)=A(I,0):A(I,7)=A(I,1
):NEXTI
50 FORI=0TO50:B=INT(RND(8)*17):FORJ=0TO7
:Z(I,J)=A(B,J):NEXTJ,I
60 FORI=0TO50:FORJ=0TO7:FORK=1TO3:SOUNDK
,Z(I,J)+(K*4),16-K:NEXTK:FORL=0TO55:NEXT
L,J,I:GOTO60
```

16 x 16 Sprite Generator

Okay, let's set the record straight — no doubt you've seen many character generators for this micro. Well, 99.99% are for 8 x 8 generation only — this little piece of code lets you produce 16 x 16 beasties for use in MAG 1 or MAG 3. To use, just move the cursor around the screen using the cursor keys. To enter (or erase) a point press the space bar, to restart press R, to actually generate your gizmo after entering all the points press CR. To see your handiwork press F and hold it down.

Notes:

10 Set up screen.

20 Continue to set up screen. Note the rubbish inside the quotation marks is in fact entered in ENG DIER's mode i.e. q is actually a "┌", 2 is "—", s is "┘", etc.

30 Set up array. X & Y are the coordinates of the cursor.

40 Position cursor (which is ENG DIER's shift V, which is a checker board shape). Then overprint with whatever is stored at that point — this gives a flashing effect and stops the erasing of data.

50 Await keyboard entry.

60 Make a sound.

70-100 See if key pressed is R, F, CR or space bar.

110-160 If a cursor key is pressed then increment or decrement x or y. Check to see if the resulting movement would take the cursor out of the 16 x 16 grid. If so, cause a "wrap-around" effect.

170 Jump back to 90.

180-190 This is jumped to if the space bar is pressed. It fills or deletes a point. The V's are in fact ENG DIER's V (which is the solid shape).

200 Data for binary.

210-240 To generate a 16 x 16 sprite. 4, 8 x 8 sprites must be generated first. This is done by supplying coordinates for all 4 sprites and calling the routine which actually converts the info to hexadecimal. A & B are the coordinates, C the position on the position on the screen the data is to be printed. N is the sprite no.

250-260 Place new sprite on screen. Pause, rerun.

270-290 Converts data held in A$ to hexidecimal. Remember "V" is actually a solid square. Q$ contains the final hex data.

300-330 This part is executed if F is pushed. It flips from MAG 1 to MAG 3 and back again if you hold down a key. A is a count. If A reads 75 it causes the MAG to change, and A is set to 0. F is the current MAG.

16x16 Sprite Editor

```
10 SCREEN 1,1:COLOR4,15:CLS:MAG1:F$=CHR$
(229)
20 PRINT"q22222222222222222w":FORA=1TO16:
PRINT"3                  3";HEX$(A-1):NEXT
:PRINT"a22222222222222222s",," 0123456789
ABCDEF":CURSOR20,9:PRINT"up,dn,lt,rt,spc
",TAB(20);"F,cr,R"
30 X=1:Y=1:ERASE:DIMA$(15,15),B$(15,15):
FORA=0TO15:FORB=0TO15:A$(A,B)="  ":NEXTB,
A
40 CURSORX,Y:PRINT"":B$=A$(X-1,Y-1):CURS
ORX,Y:PRINTB$
50 C$=INKEY$:IFC$=""THEN40
60 SOUND1,1200,15:SOUND0
70 IFC$="R"THEN10
80 IFC$="F"THENGOSUB300
90 IFC$=CHR$(13)THEN200
100 IFC$="  "THENGOSUB180
110 X=X-(C$=CHR$(28))+(C$=CHR$(29))
120 Y=Y-(C$=CHR$(31))+(C$=CHR$(30))
130 IFX>16THENX=1
140 IFY>16THENY=1
150 IFY<1THENY=16
160 IFX<1THENX=16
170 GOTO40
180 IFA$(X-1,Y-1)="  "THENA$(X-1,Y-1)=F$:
RETURN
190 IFA$(X-1,Y-1)=F$THENA$(X-1,Y-1)="  ":
RETURN
200 DATA128,64,32,16,8,4,2,1
210 FORA=0TO3:Q$(A)="":NEXT:X=0:FA=0:TA=
7:FB=0:TB=7:GOSUB250
220 X=2:FA=8:TA=15:FB=0:TB=7:GOSUB250
230 X=4:FA=0:TA=7:FB=8:TB=15:GOSUB250
240 X=6:FA=8:TA=15:FB=8:TB=15:GOSUB250:G
OTO280
250 X1=X/2:CURSOR20,X:FORA=FATOTA:T=0:RE
STORE:FORB=FBTOTB:READQ:IFA$(B,A)=F$THEN
T=T+Q
260 NEXTB:T$=HEX$(T):IFT<16THENT$="0"+T$
270 PRINTT$;:Q$(X1)=Q$(X1)+T$:NEXTA:PATT
ERN#X1,Q$(X1):RETURN
280 SCREEN2,2:CLS:MAG1:SPRITE0,(128,96),
0,1
290 FORA=0TO1000:NEXT:SCREEN1,1:GOTO50
300 SCREEN 2,2:A=0:F=1:MAG1
310 IFA>75THENF=(3ANDF=1)+(1ANDF=3):MAGF
:A=0
320 A=A+1:IFINKEY$<>""THEN310
330 SCREEN1,1:RETURN
```

Star Trek ... The Author's View (that's me!)

In my opinion, Star Trek is the best thing since sliced bread and plug-in ROM cartridges. But little is known about the origins of the program (and films). Well, fellow computniks, I have been doing a lot of studying (and watching tellie) and at last I have found the origins to the world's best series — here goes.

In the early to mid sixties (around the time I was hatched) a not-so-young gentleman got out his computer (which turned out to be a predecessor to the SEGA SC-3000) and literally got it to write scripts for him. His name was Gene Roddenberry, and now for the big news ... I managed to pinch a copy of that very program! Without more ado, here it is ...

Notes:

10	Set up array Z$. Print title.
20	Declare a function to produce random numbers.
30	Select a crew member. This is done in a special (patent pending) method. Here is an example of how it works. Let's say A takes on the value of 4. Okay, firstly we restore 1000 thus pointing to Kirk. Now if A is bigger than 1, which it is, we restore 1020, thus pointing to Spock. Now if A is bigger than 2, which it is, we store 1030, thus pointing to Scottie. Now if A is bigger than 3, which it is, we restore 1040 thus pointing to Sulu. Now if A is bigger than 4, which it isn't, we are stuck at Sulu. Okay? Good!
40	After selecting a crew member we read Q and Q$. Q holds the number of sentences that person has listed, and Q$ is his/her name. Then a sentence is selected.
45-47	Here is where Z$ comes into use. It is used to make sure that a specific sentence is not repeated too often.
50	If Kirk says, "Take us out of orbit Mr Sulu," the program ends.
55-60	Wait for a key to be pressed. Make a beep. Jump back.
1000-1070	Data for each crew member. The number at the beginning of each block, is one less than the number of sentences.

NOTE: Only bridge crew are listed. This is because other crew members such as security, don't say much. They just get killed!

Star Trek... The Authors View

```
10 DIMZ$(5):CLS:PRINT"STAR TREK.........."
20 DEFFNR(X)=INT(RND(8)*X)
30 A=FNR(7)+1:RESTORE1000:IFA>1THENRESTORE1020:IFA>2THENRESTORE1030:IFA>3THENRESTORE1040:IFA>4THENRESTORE1050:IFA>5THENRESTORE1060:IFA>6THENRESTORE1070
40 READQ,Q$:Q=Q+1:A=FNR(Q):FORB=0TOA:READA$:NEXTB
45 FORA=0TO5:IFZ$(A)=A$THEN30
47 NEXTA:FORA=0TO4:Z$(A)=Z$(A+1):NEXTA:Z$(5)=A$:PRINTQ$,,A$
50 IFLEFT$(A$,4)="Take"THENPRINT:PRINT"THE END.....":END
55 IFINKEY$=""THEN55
60 BEEP:GOTO30
1000 DATA9,"KIRK:","I'm responsible for the lives of 3000 crewmen","It's a 1000 to 1 against,but it's our only chance!","What is it Spock?","We have no time","Set phasers to stun"
1010 DATA"Take us out of orbit Mr. Sulu","What are you going to do to my ship?","Inform Star-fleet command Lt. Uhura","To the transporter room","Get a landing party ready"
1020 DATA6,"SPOCK:","It appears to be some kind of unknown energy captain","Fascinating","Most illogical","Any one for 3D chess?","No life forms on the surface","A Nitrogen-Oxygen atmosphere,similar to that found on Earth","I doubt if phasers will work"
1030 DATA5,"SCOTTIE:","The enjins canna take it","She canna take the strain cap'n","All defences out Cap'n","One more and we're done for","Impulse is out","We can just manage Warp 5"
1040 DATA3,"SULU:","Romulans Captain","It just....disappeared","It's the Klingons","But captain,it would take us into an  uncharted region..."
1050 DATA2,"CHEKOV:","What is it...?","But captain...","I've never seen one like it"
1060 DATA6,"McCOY:","He's dead Jim","He was just standing there","Meet me in the Sickbay..","This shot should do it","You need to rest Jim","But you're in no fit state","Physically he's fine...But.."
1070 DATA1,"UHURA:","I can't raise them on any channel","They're jamming all channels Captain"
```

SSSnake

This program has been nicked! Pirated by me! Shock, horror. Sorry about that bit of drama — the program I have listed was nicked from an old machine-code program I wrote for another computer some years ago, so it's not that bad ("Shame", I hear you say — I bet you thought you were in for a bit of excitement!).

In this game you take on the part of SSSid the SSStarved SSSnake. He must run around the screen gobbling up dots (no, this is not Pac Man), whilst leaving a trail behind him. He must not bump into his trail or the wall. You control SSSid by using the Q key to go up, Z down, P right, I left. As the game progresses you have to devour more dots and this leads to longer trails therefore making the game very hard. The actual high score set by my now deceased cat is 286.

Notes:

10 Set screen colouring. Define a random number function, and define the shape of the objects to be gobbled. Set high score to 0.

20-30 Define shape of SSSid. TS is a variable used to detect if a screen is completed. S is the number of munchies on the screen. P is your position on the screen (roughly central). D is set to a random initial direction. Score set to 0.

40-60 Set up display. The e's are in fact ENG DIER's shift e (which is a chess board shape). The Vpoke places SSSid (you) on the screen.

70-100 Place munchies at random positions on screen. A holds the position. If this position is already occupied re Vpeek (a) is not a space (32), then select another random number.

110 Make a sound to warn you that the time is nigh.

120 If a key pressed = P, Q, I or Z then alter D1 accordingly. If D1 then D = D1, this is the same as if D1 <> 0 then D = D1. Basically this means that if a key is pressed the alter D. If a key is NOT pressed then D (which holds the direction you are facing 1 = right, -1 = left, -40 = up, 40 = down) stays as it is. This makes the SSSnake continue in the direction it is heading.

130 Let B = what is in the space next to your position and your direction. If B is not a space (32) and not a bit of food (255) then you must have hit either the wall or your own trail, so jump to the death bit.

140 If B = a bit of grub (255), then increase score, make a burp. If S = SC then you have complete the screen — if this is true jump to 160.

150 Add your position and direction together to get your new position. Place you on the screen by Vpoking you. Goto120.

160-180 Make a pretty sound. Increase number of food. Get ready.

190-250 Dead! Place an asterisk in your death position. The outs all help in creating an explosion. Check for new high score. Print high score, await a key to be pressed.

SSSnake

```
10 COLOR4,15:DEFFNR(X)=INT(RND(8)*X):PAT
TERNC#255,"FF848484848484FF":HI=0
20 PATTERNC#127,"007878787878780O":TS=0:
C$=A$:S=6:P=&H3DA3
30 D=SGN(RND(8)-RND(8)):SC=0
40 CLS:PRINT"eeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeee"
50 FORI=1TO18:CURSOR0,I:PRINT"e
                              e":NEXTI
60 CURSOR0,19:PRINT"eeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeSCORE:";TS:VPOKEP,127
70 FORI=1TO5
80 A=&H3C02+FNR(800):IFVPEEK(A)<>32THEN8
0
90 VPOKEA,255
100 NEXTI
110 FORI=110TO300STEP5:SOUND1,I,15:NEXTI
:FORI=15TO0STEP-.25:SOUND1,,I:NEXTI
120 Z$=INKEY$:D1=(1ANDZ$="P")-(1ANDZ$="I
")-(40ANDZ$="Q")+(40ANDZ$="Z"):IFD1THEND
=D1
130 B=VPEEK(P+D):IFB<>32ANDB<>255THEN190
140 IFB=255THENSC=SC+1:CURSOR6,20:PRINTS
C+TS:SOUND1,2000,15:SOUND0:IFSC=STHEN160
150 P=P+D:VPOKEP,127:GOTO120
160 VPOKEP+D,229:FORI=160TO255:SOUND1,I,
15:NEXTI:SOUND1,,10:SOUND0:S=S+1:TS=TS+S
:CURSOR6,20
170 CLS:PRINT"GET READY FOR ROUND#";S-5
180 FORI=0TO500:NEXTI:GOTO30
190 VPOKEP+D,42:OUT&H7F,&HE4:FORI=&HF0TO
&HFF:OUT&H7F,I:FORZ=0TO6:NEXTZ,I
200 CURSOR10,20:PRINT"HA HA HA...."
210 IFTS+SC>HITHENHI=TS+SC
220 CURSOR27,20:PRINT"HI";HI
230 IFINKEY$<>""THEN230
240 IFINKEY$=""THEN240
250 GOTO20
```

Minefield

This is one of my favourite little games, 'cos if you make a mistake you get blown up!

The game is really easy — you guide a little chappie ("+") through a field dodging mines. You move the man by using the "P" and "Q" keys to move right and left respectively. You move forward automatically, the speed of which is governed by the level of difficulty.

When you pass away (die), as you eventually will, your score will be displayed. A good score is around 40,000. Oh, and by the way, don't bump into the fence surrounding the field or you'll get zapped!

Notes:

10	Clear screen. Input the level of difficulty, make sure it is inbetween 1 and 100. The level is stored in LV.
20	Clear screen. Set a new random number pattern (see page 77 of operator's manual).
30-40	Set up border. Note, the 'i' is not an i but ENG DIER's shift G (which is a sort of snapped chess board!).
50	Place a number of mines (determined by level [LV]) at random positions on screen.
60	M is your y (downward) position. X is a random x (sidewards) position. Place you at the M, X position. Warn you with an audible alarm.
70	Increase score.
80	Check to see if you have bumped into anything. This is done by converting your X and M position to a memory location on the screen (see pages 143-144 of operator's manual). The mines are CHR$ (233).
90	Place you at your position again.
100	Delay controlled by level.
110	Decrease your y position. This makes you go up the screen. If M = Q you have reached the other side of the field.
120-130	If Q is pressed X is increased by one. If P is pressed X is decreased by 1. Loop back to 70.
140-170	You made a boo-boo and got blown up! Make oral explosion sound.
180	Cause field to scroll up screen.
190	Tell you, you've finished that field. Increase level of difficulty by decreasing LV. Delay. Loop to 20.
200-210	Press a key routine.

```
Mine Feild

10 SCREEN 1,1:CLS:INPUT"Level 1-Difficul
t,100-Easy ";LV:S=0:IFLV<10RLV>100THEN10
20 CLS:L=RND(-1)
30 FORI=0TO31:CURSORI,0:PRINT"i":CURSORI
,21:PRINT"i":NEXTI
40 FORI=1TO20:CURSOR0,I:PRINT"i";TAB(31)
;"i":NEXTI
50 FORI=1TO65+(40-(LV/2)):CURSORINT(RND(
8)*31),INT(RND(8)*17)+1:PRINT"i":NEXTI
60 M=20:X=INT(RND(8)*28)+1:CURSORX,M:PRI
NT"+":FORI=0TO4:BEEP:NEXT I
70 S=S+230
80 IFVPEEK(M*40+X+&H3C02)=233THEN140
90 CURSORX,M:PRINT"+":SOUND1,1000,15:SOU
ND0
100 FORG=1TOLV:NEXT G
110 M=M-1:IFM=0THEN180
120 A$=INKEY$:X=X-(1ANDA$="0")+(1ANDA$="
P")
130 GOTO70
140 FORI=15TO0STEP-1:SOUND3,2000:SOUND4,
3,I:CURSORX,M:PRINT"*":CURSORX,M:PRINT"+
":CURSORX,M:PRINT" ":NEXT
150 CLS:S=S-300
160 CURSOR0,10:PRINT"You Crashed...Score
:";S:CURSOR3,15:PRINT"Another Go..?"
170 GOTO200
180 FORI=1TO45:PRINT:NEXTI
190 CURSOR2,10:PRINT"Well done.Score so
far is";S:CURSOR5,15:PRINT"Get Ready for
 next go..":LV=LV-(LV/3):FORL=0TO1100:NE
XTL:GOTO20
200 IFINKEY$<>"Y"THEN200
210 GOTO10
```

Quadratic Regression

What a mouthful! With a title like that it should stay where I got the idea from — in a maths book! Seriously folks, regression is one of the most useful statistical methods for examining scientific, economic or social data. Basically it allows you to enter a few points and from then on you can find any point. Baffled? Well, imagine this — we all know that 212°F = 100°C, and that 32°F = 0°C, but imagine if, even though we know this data, we want to know what -40°F is in celsius. Well, this little program allows you to do this. Or, say we know that a lump of steel expands by 2.2% at 500°C and 3% at 750°C and 10% at 2000°C (purely arbitary values chosen), and we want to know by how many % that lump expands by at -100°C. Well, this gem of a program lets you do this, and much more — convert any metric to imperial sizes and vice versa, convert literally any data, so long as at least 2 points are known.

I am not going to give a rundown of how the program works because as you can see it is fairly complex. But I will give you an example of how it works. Let's take the temperature example.

How many points ... 2.
Enter x value 212
Enter y value 100
Enter x value 32
Enter y value 0

Point 1 212°F = 100°C

Point 2 32°F = 0°C

Enter a value for x -40 — We want to know what -40°F is.
If x = 40 then y = -40 -40°F = -40°C

To the more mathematical minded among you, remember not only is linear regression supported but quadratic regression is also, but many more points must be entered so as to fit the quadratic curve. The form of a quadratic equation is:

$$y = ax^2 + bx + c$$

This program has many possible serious applications.

```
Quadratic Regression

10 CLS
20 T=2:A=0:B=0:C=0:D=0:E=0:F=0:G=0:H=0
30 INPUT"HOW MANY POINTS...";N
40 FORM=1TON:PRINT"POINT";M
50 INPUT"ENTER X VALUE";X
60 INPUT"ENTER Y VALUE";Y
70 A=A+X:B=B+Y
80 C=C+ABS(X)^T:D=D+ABS(Y)^T
90 E=E+ABS(X)^3:F=F+ABS(Y)^4
100 G=G+X*Y
110 H=H+ABS(X)^T*Y
120 NEXTM
130 Z=(C-ABS(A)^T/N)*(F-ABS(C)^T/N)-ABS(
E-A*C/N)^T
140 I=((F-ABS(C)^T/N)*(G-B*A/N)-(H-B*C/N
)*(E-A*C/N))/Z
150 J=((C-ABS(A)^T/N)*(H-B*C/N)-(G-B*A/N
)*(E-A*C/N))/Z
160 Z=B/N-A/N*I-C/N*J
170 CLS
180 INPUT"ENTER A VALUE FOR X ";X
190 CLS
200 Y=Z+I*X+J*ABS(X)^T
210 PRINT"IF X =";X;" THEN Y=";Y
220 GOTO180
```

Bingo

This is a Granny's delight! The only problem is, is that you need a printer as cards are run off. I'm not going to explain how to play Bingo because you already know (or if you don't, find out!). It allows you to select the number of players (2 or more generally), then there is a pause as each player's cards are generated. This may take some time as the computer must check that numbers on the same card are not replicated. Then each player's cards are printed on the printer, and the game commences as the computer generates random numbers. The winner is the first to get all their numbers called up. When a winner is declared, you press "H" and you start to enter the numbers on the winner's card. If a number has not been called the game continues. Otherwise an "All okay" message appears ... we have a winner!

Notes:

10	Clear the screen. Define a random number function.
20	Enter number of players.
30	Delete old arrays. Re-dimension arrays.
40-90	Generate cards for all players. Lines 70 and 80 make sure there is no replication.
100-130	Print cards. Z is the spacing of numbers.
140-150	Pretty title.
160	Set number of numbers so far (C = 1).
170-190	Print a non-replicated number. Make a pretty burp.
200-210	Wait for a key to be pressed. If "H" is pressed then a winner has said his card has been completed so jump to 230.
220	A key has been pressed, it is not "H", so increase C and goto 170.
230-270	Check that all numbers on winner's card are correct.

```
Bingo

10 CLS:DEFFNR(X)=INT(RND(8)*X)+1
20 INPUT"ENTER NUMBER OF PLAYERS ";N
30 ERASE:DIMA(N,23),S(100)
40 FORB=1TON
50 FORC=0TO23
60 Z=FNR(100)
70 F=1:FORD=0TOC:IFA(B,D)<>ZTHEN NEXTD:F
=0:A(B,C)=Z
80 IFFTHEN60
```

```
90 NEXTC,B
100 FORB=1TON:LPRINT"----------------------
----------------------Player #";B
110 Z=0:FORC=0TO23:LPRINTTAB(Z);A(B,C);:
Z=Z+4:IFZ>30THENZ=0:LPRINT
120 NEXTC:LPRINT:NEXTB
130 LPRINT"--------------------------------
----------"
140 CLS:PRINT"Bingo"
150 PRINT:PRINT"Okay...lets go"
160 C=1
170 PRINT"#";C;" =";
180 Z=FNR(100):IFS(Z)THEN180
190 S(Z)=1:PRINTZ:SOUND1,2000,15:SOUND2,
3000,15:SOUND3,4000,15:SOUND0
200 A$=INKEY$:IFA$=""THEN200
210 IFA$="H"THEN230
220 C=C+1:GOTO170
230 CLS:PRINT"Start to enter the numbers
"
240 FORB=0TO23
250 INPUTZ:IFS(Z)<>1THENPRINT"ERROR #";Z
;" has not been selected":GOTO220
260 PRINT"Okay":NEXTB
270 PRINT"All okay"
```

```
------------------------------------Player # 1
 98  94   4   80   1   5   37  79
 63  99   7   85   9  68   77  61
 24  64  21   65  90  32   48  17

-----------------------------------
```

Calendar

Way, way back in the good ol' days of the Music Hall (so they say!), there was a little man whose act, so he claimed, was based on fabulous feats of memory. Give him the date your pet Purple People Eater was hatched and he'll tell you what DAY it was born on! Amazing stuff! No ... not really (sorry if I shattered all your ideas!). The act is based on maths, not memory, and the artist only has to be able to mentally manipulate a couple of formulae.

I discovered all this info from a book I found in a deep, dark, dusty and dangerously-deadly dungeon (my bedroom!), and I was urged to convert the formulae to the SEGA.

Here we go ... in 1752 England and her colonies adopted the Gregorian calendar, so this means that any date from the 1st of January 1752 to whenever can be worked out.

The 2 formulae used were worked out by a German mathematician called C.F. Gauss. The first formula is:

$$X = INT ([2.6 * M] - .2 + D + Y + INT [Y / 4] + INT [C / 4] - [2 * C])$$

In this:

M = month. According to the figuration Mar = 1, Apr = 2 ... Dec

= 10, Jan = 1 and Feb = 2, but Jan and Feb are considered as months of the previous year.

D = Day in that month re 1-31 or 1-30 or 1-28 or 1-29.

Y = Last 2 digits of year.

C = First 2 digits of year.

Now X is worked out, it will be negative, so multiples of 7 are added to X until it becomes positive. Then the 2nd formula is applied.

Z = X – (7 * INT [X / 7])

The result is then applied to days of the week. 0 = Sunday, 1 = Monday, etc.

```
Calender

10 DATASUNDAY,MONDAY,TUESDAY,WEDNESDAY,T
HURSDAY,FRIDAY,SATURDAY
20 CLS
30 INPUT"ENTER YEAR ";A
40 INPUT"ENTER MONTH ";N
50 INPUT"ENTER DATE ";D
60 IFN>2THENM=N-2:GOTO90
70 IFN=1THENM=11
80 IFN=2THENM=12
90 J=A/100:C=INT(J):Y=(J-C)*100:IFN=10RN
=2THENY=Y-1
100 X=INT((2.6*M)-.2+D+Y+INT(Y/4)+INT(C/
4)-(2*C))
110 FORG=7TO56STEP7
120 IFX<0THENX=X+G
130 IFX>0THEN150
140 NEXTG
150 Z=X-(7*INT(X/7))
160 CLS:RESTORE
170 FORI=0TOZ:READA$:NEXTI
180 PRINTD;"/";N;"/";A,,"IS A ";A$
```

The Code Machine

This code writing and cracking is a must for all you cloak-and-dagger types! It produces code which is just about impossible to decode, unless you know the secret "seed-word". Oh, by the way, you create the seed-word!

The system is based on using this seed to start the code alphabet of the corresponding letter of the message being coded. Sounds confusing? This simple example should clarify things.

Let's say the seed is DAVID.
And my message is simply HELLO.

The first letter of my message ("H") is coded starting the alphabet at "D" not A (the first letter of the seed is "D" from "DAVID"). Thus "H" = "K".

"E" is coded using "A" as the start so remains unaltered. "E" = "K". Now "L" is dealt with differently inasmuch as "L" being the 12th letter of the normal alphabet, and starting our code alphabet at "V", we run out of letters at "Z"! So all we do is start again at "A" so "L" = "G". The alphabet can be viewed as an endless loop or circle with "A" following "Z".

We continue in this manner and the message ends up as "KEGTR". When the message is longer than the seed, the whole series is repeated until the message has been coded (or decoded).

Notes:

10-50 Set up arrays and variables. Heading select mode (1 or 2).

60-120 Enter seed. Search along seed, assessing each character by means of its ASCII number, discarding any not in the range "A"-"Z". The new seed is stored in array CB.

140-380 Accept message (or coded message), convert all characters into ASCII numbers and store in array ML. Compute message and store in array CM.

390-460 Print coded or decoded message.

```
Code Machine

10 ERASE:DIMCB(210),ML(210),CM(210):C=0:
L=0:M=0
20 CLS:PRINTTAB(7);"The code machine",TA
B(7);"---- ---- --------"
30 CURSOR9,5:PRINT"Select mode",,,TAB(9)
;"1. Encode message",,,TAB(9);"2. Decode
 message"
40 A$=INKEY$:IFA$<"1"ORA$>"2"THEN40
50 S=ASC(A$)-48
60 CLS:INPUT"Enter seed ";CB$
70 X=LEN(CB$)
80 FORI=1TOX:Z$=MID$(CB$,I,1)
90 IFASC(Z$)<65 OR ASC(A$)>90THEN120
100 C=C+1
110 CB(C)=ASC(Z$)
120 NEXTI
130 ON S GOTO140,150
140 PRINT:PRINT"Message..":GOTO160
150 PRINT:PRINT"Coded message.."
160 INPUTM$:Y=LEN(M$)
170 FORJ=1TOY
180 ML(J)=ASC(MID$(M$,J,1))
190 NEXTJ
200 FORK=1TOY
210 L=L+1
220 IFML(L)<65ORML(L)>90THEN370
230 M=M+1
240 CL=CB(M)
250 IFM<=CTHEN280
260 M=M-C
270 GOTO240
280 ON S GOTO290,330
```

```
290 CM(K)=ML(L)+CL-65
300 IF CM(K)<=90THEN380
310 CM(K)=CM(K)-26
320 GOTO380
330 CM(K)=ML(L)-CL+65
340 IF CM(K)>=65THEN380
350 CM(K)=CM(K)+26
360 GOTO380
370 CM(K)=ML(L)
380 NEXTK
390 CLS
400 ON S GOTO410,430
410 PRINTTAB(8);"Coded Message:"
420 GOTO440
430 PRINTTAB(7);"Decoded Message:"
440 FORL=1TOY:PRINTCHR$(CM(L));:NEXTL
450 IFINKEY$=""THEN450
460 GOTO10
```

Revenge of the Mutant Apples!!

Beware, fellow cyberphiliacs, whilst you are munching on that crunchy, juicy, green apple (it's okay — red apples are safe!), some of our kinsmen are having to fight giant sized Mutant Apples (green ones) on the very edge of time (and sanity!). Hold on a minute — what? Ha, ha! Well, how's this? You have been selected to join the force to combat the second deadliest foe in the cosmos. Whilst in space you must simply dodge the apple by using the "Q" and "P" keys to move left and right respectively, but watch out! — the Mutant Apples go around in pairs.

Notes:

10-20 Redefine characters 254 and 33. 254 = you, 33 = Mutant Apples.

30 Clear screen. Set X to rougly 3 lines down in the middle of the text screen. X will store your position on screen.

40 S holds score. (1E6 = 1,000,000.) Place Mutant Apples on screen.

50 Check to see if you have crashed into a Mutant Apple.

60 Place you back on the screen.

70 If I$ = "Q" decrease X (move left). If I$ 5 "P" increase X (right).

80-90 Make sure you don't go off the edges. If you want to increase the speed of the game remove these lines.

100 Loop back.

110-150 You muffed it! Make a sound. Show score. Press a key.

```
Revenge of the Mutant Apples

10 PATTERNC#254,"88D8F8F8A8702020"
20 PATTERNC#33,"081078FCFCBC9C78"
30 CLS:X=&H3CDA
40 FORS=0TO1E6:CURSORRND(8)*35,22:PRINT"
!!":PRINT
50 IFVPEEK(X)=33THEN110
60 VPOKEX,254
70 I$=INKEY$:X=X+(I$="Q")-(I$="P")
100 NEXTS
110 VPOKEX,42
120 FORA=15TO0STEP-.2:SOUND3,5400:SOUND4
,3,A:NEXT
130 CURSOR0,0:PRINT"Score:";S
140 IFINKEY$=""THEN140
150 GOTO30
```

Ice Cream

Can you tell the difference between a Lolly-Munch-Cherry-Choc-Bomb and a Finger 'n' Face-Freezin'-Fire-Fruit-Fantasy ... ? You can! Well this is the game you've been waiting for! But you'll have to be a real cool customer to become the top ice cream vendor in Segasville. Up to 4 players can take part in this educational and fun-to-play buying and selling game.

Each player runs a van selling various cold delights. You are in charge for 7 days and each morning you must buy stock, but remember to keep an eye on the weather and at weekends more people are around! So keep your fingers crossed for a blistering hot Saturday and not a rainy Monday!

Notes:

10 Set up arrays. RS = ice cream remaining
RC = cones remaining
RH = choc-ice remaining
RL = lolly remaining
GA = ice cream sold
GD = choc-ice sold
GE = lollies sold
CS = various sales
TA = takings
PR = profit
A = selling potential of cones i.e. your price
C = no. of cones bought
D = selling potential of cartons
S = no. of cartons bought
E = selling potential of lollies
L = no. of lollies bought
PA = return value of A
PB = return value of B
PE = return value of E
PD = return value of D
X = cash in hand

30-50	Press a key.
60-270	Evaluate data.
300-360	Select weather.
380-390	Select day.
400-420	Start.
430-1010	Main loop. Enter sales, promotions, costs, etc, Most working out.
1020-1260	End-of-day trading and summary.

```
Ice Cream

10 DIMRS(4),RC(4),RH(4),RL(4),GA(4),GD(4
),GE(4),CS(4),TA(4),PR(4),A(4),C(4),D(4)
,S(4),E(4),L(4),PA(4),PB(4),PE(4),PD(4)
20 GOTO400
30 PRINT"PRESS ANY KEY TO CONTINUE....."
40 IFINKEY$=""THEN40
50 RETURN
60 FORI=1TOV:IFA(I)-9<1THENP(I)=1:GOTO11
0
70 IFA(I)-9<3THENPA(I)=.7:GOTO110
80 IFA(I)-9<6THENPA(I)=.5:GOTO110
90 IFA(I)-9<10THENPA(I)=.25:GOTO110
100 PA(I)=0
110 NEXTI
120 RETURN
130 FORK=1TOV
140 FORI=1TOV:IFD(I)-10<1THENPD(I)=1:GOT
O190
150 IFD(I)-10<3THENPD(I)=.7:GOTO190
160 IFD(I)-10<6THENPD(I)=.5:GOTO190
170 IFD(I)-10<10THENPD(I)=.25:GOTO190
180 PD(I)=0
190 NEXTI
200 RETURN
210 FORI=1TOV:IFE(I)-5<1THENPE(I)=1:GOTO
260
220 IFE(I)-5<3THENPE(I)=.7:GOTO260
230 IFE(I)-5<6THENPE(I)=.5:GOTO260
240 IFE(I)-5<10THENPE(I)=.25:GOTO260
250 PE(I)=0
260 NEXTI
270 RETURN
280 X(K)=INT(X(K)*100)/100:RETURN
290 PRINT"YOU HAVE OVERSPENT.TRY AGAIN":
PRINT:RETURN
300 ON W GOTO 310,320,330,340,350
310 B$="SUNNY TEMP 33°C":MK=1000:GOTO360
320 B$="SUNNY TEMP 23°C":MK=700:GOTO360
330 B$="CLOUDY TEMP 20°C":MK=500:GOTO360
340 B$="SHOWERY TEMP 15°C":MK=300:GOTO36
0
350 B$="RAIN...!":MK=100
360 IFDA=6ORDA=7THENMK=MK*2
370 RETURN
380 DATAMONDAY,TUESDAY,WEDNESDAY,THURSDA
Y,FRIDAY,SATURDAY,SUNDAY
390 RESTORE380:FORI=1TODA:READA$:NEXT:RE
TURN
400 CLS:PRINT"ICE CREAM SALES....":PRINT
410 PRINT"IS THERE 1,2,3 OR 4 ICE-CREAM
VANS??"
```

```
420  INPUTV:V=INT(V):IFV<1ORV>4THEN2275
430  DA=0:FORI=1TO4:X(I)=100:SS(I)=0:SC(I
)=0:SH(I)=0:SL(I)=0:NEXTI
440  FORD=1TO7
450  DA=DA+1
460  W=INT(RND(8)*5)+1:GOSUB300
470  FORK=1TOV
480  GOSUB380
490  CLS
500  Q=X(K)
510  PRINT"      VAN";K;"      ";A$:PRINT
520  PRINT"WEATHER FORECAST ";B$:PRINT
530  GOSUB1270
540  PRINT"HOW MANY CARTONS OF ICE-CREAM
DO YOU  WISH TO BUY AT $6 EACH?":PRINT"(
1 CARTON HOLDS 100 PORTIONS)"
550  INPUTS(K):IFS(K)*6<=X(K)THEN570
560  GOSUB290:GOTO550
570  X(K)=X(K)-S(K)*6:GOSUB280
580  GOSUB1270
590  PRINT"HOW MAY CONES AT 3c EACH"
600  INPUTC(K):IFC(K)*.03<=X(K)THEN620
610  GOSUB290:PRINT:GOTO600
620  X(K)=X(K)-C(K)*.03:GOSUB280
630  GOSUB1270
640  PRINT"HOW MANY CHOC-ICES AT 10c EACH
"
650  INPUTCH(K):IFCH(K)*.1<=X(K)THENGOTO6
70
660  GOSUB290:PRINT:GOTO650
670  X(K)=X(K)-CH(K)*.1:GOSUB280
680  GOSUB1270
690  PRINT"HOW MANY LOLLIES AT 5c EACH"
700  INPUTL(K):IFL(K)*.05<=X(K)THEN720
710  GOSUB290:PRINT:GOTO700
720  X(K)=X(K)-L(K)*.05:GOSUB280
730  GOSUB1270
740  PRINT:PRINT"DO YOU WISH TO ALTER ANY
THING ?(Y/N)"
750  A$=INKEY$:IFA$<>"Y"ANDA$<>"N"THEN750
760  IFA$="N"THEN780
770  X(K)=Q:GOTO490
780  SS(K)=SS(K)+S(K)*100
790  SC(K)=SC(K)+C(K):SH(K)=SH(K)+CH(K)
800  SL(K)=SL(K)+L(K)
810  CLS:PRINT"WHAT IS THE SELLING PRICE
FOR AN":PRINT"ICE-CREAM CONE IN CENTS ?
"
820  INPUTA(K)
830  PRINT"WHAT IS THE SELLING PRICE FOR
A ":PRINT"CHOC-ICE IN CENTS "
840  INPUTD(K)
850  PRINT"WHAT IS THE SELLING PRICE FOR
A ":PRINT"LOLLY IN CENTS "
860  INPUTE(K)
870  PRINT:PRINT"DO YOU WISH TO CHANGE AN
YTHING?(Y/N)"
880  A$=INKEY$:IFA$<>"Y"ANDA$<>"N"THEN880
890  IFA$="N"THEN910
900  GOTO810
910  GOSUB60:GOSUB140:GOSUB210
920  GA(K)=PA(K)*MK:IFGA(K)>SS(K)THENGA(K
)=SS(K)
930  IFGA(K)>SC(K)ANDSS(K)>SC(K)  THEN GA(
K)=SC(K)
940  GE(K)=PE(K)*MK:IFGE(K)>SL(K)THENGE(K
)=SL(K)
950  GD(K)=PD(K)*MK:IFGD(K)>SH(K)THENGD(K
)=SH(K)
960  RS(K)=SS(K)-GA(K):RC(K)=SC(K)-GA(K)
970  RH(K)=SH(K)-GD(K)
980  RL(K)=SL(K)-GE(K)
990  X(K)=X(K)+(GA(K)*A(K)+GD(K)*D(K)+GE(
K)*E(K))/100
```

```
1000 GOSUB280:GOSUB140
1010 NEXTK
1020 FORK=1TOV
1030 GOSUB380
1040 PRINT"PORTIONS":PRINT
1050 CLS
1060 PRINT"      VAN";K;"    ";A$:PRINT
1070 PRINT"TRADING POSITION AT THE END O
F THE DAY"
1080 PRINTTAB(10);"STOCK";TAB(20);"STOCK
";TAB(30);"STOCK":PRINT
1090 PRINTTAB(10);" A.M.";TAB(20);"SOLD"
;TAB(30);"LEFT":PRINT
1100 PRINT"ICE"
1110 PRINT"CREAM";TAB(10);SS(K);TAB(20);
GA(K);TAB(30);RS(K)
1120 PRINT"CONES";TAB(10);SC(K);TAB(20);
GA(K);TAB(30);RC(K):PRINT
1130 PRINT"CHOC-ICES";TAB(10);SH(K);TAB(
20);GD(K);TAB(30);RH(K):PRINT
1140 PRINT"LOLLIES";TAB(10);SL(K);TAB(20
);GE(K);TAB(30);RL(K):PRINT
1150 GOSUB1270
1160 PRINT:PRINT:GOSUB30
1170 SS(K)=RS(K):SC(K)=RC(K)
1180 SH(K)=RH(K):SL(K)=RL(K)
1190 NEXTK,D
1200 CLS
1210 PRINTTAB(15);"SUMMARY":PRINT:PRINT
1220 PRINTTAB(2);"VAN";TAB(8);"CASH IN H
AND";TAB(25);"PROFIT":PRINT
1230 FORK=1TOV
1240 PRINTTAB(2);K;TAB(12);X(K);TAB(26);
X(K)-100
1250 NEXTK
1260 END
1270 PRINT"----------------------"
1280 PRINT"CASH IN HAND $";X(K)
1290 PRINT"----------------------":RETURN
```

```
     VAN 1      MONDAY

WEATHER FORECAST SUNNY TEMP 33°C
                \
--------------------
CASH IN HAND $ 100
--------------------
HOW MANY CARTONS OF ICE-CREAM DO YOU
WISH TO BUY AT $6 EACH?
(1 CARTON HOLDS 100 PORTIONS)
? 10
--------------------
CASH IN HAND $ 40
--------------------
HOW MAY CONES AT 3c EACH
? 100
--------------------
CASH IN HAND $ 37
--------------------
HOW MANY CHOC-ICES AT 10c EACH
? 100
```

```
     VAN 1    MONDAY
TRADING POSITION AT THE END OF THE DAY
                STOCK       STOCK       STOCK
                 A.M.       SOLD        LEFT
ICE
CREAM           1000         900         900
CONES            100          50          50

CHOC-ICES        100          33          66

LOLLIES          100         100           0

---------------------
CASH IN HAND $ 32
---------------------

PRESS ANY KEY TO CONTINUE....
```

Slide Puzzle

Okay, here's your chance to relieve the frustration you felt with The Cube. I actually wrote this program to try and get better at those games in which you have to move numbered tiles around so that they read 1-15 consecutively with the space being in the bottom righthand corner. Well, I'm still useless at them!

When RUN, the computer may take up to 10 seconds to set up the puzzle. This delay is due to the set-up procedure necessary to avoid impossible puzzles. Then the computer will state the number of moves it should take to do.

When replying to "WHAT IS YOUR MOVE", the player must specify first the direction of the move (left, right, up or down) and second, the number of pieces to be moved (3 is maximum). For convenience, only the first letter of the direction need be entered, and if the number of pieces to move is 0 or non-existant, the computer will assume you wish to move as many as possible.

Notes:

A stores the game. It is a 16-element array. The position of the space is stored in D. N is the number of pieces to move. D is thedirection moved. A$ stores the possible directions. M is the number of moves.

310-450 Set up puzzle.

520-650 Print out puzzle.

660-670 Check if puzzle is completed.

710-720 Find numeric equivalent to direction chosen.

810-820 Check vertical movement.

980-990 Check horizontal movement.

920-970 Moves pieces.

```
Slide Puzzle
SOLVABLE IN 18 MOVES

+-----------------------+
!                       !
!      1      6      2      3!
!                       !
!      9     13      5      7!
!                       !
!            14     15     10!
!                       !
!     11      8     12      4!
!                       !
+-----------------------+

MOVE # 1 WHAT IS YOUR MOVE DOWN


SOLVABLE IN 18 MOVES

+-----------------------+
!                       !
!             6      2      3!
!                       !
!      1     13      5      7!
!                       !
!      9     14     15     10!
!                       !
!     11      8     12      4!
!                       !
+-----------------------+

MOVE # 2 WHAT IS YOUR MOVE R
```

```
 Slide Puzzle

1 DEFFNR(X)=INT(RND(8)*X)
10 CLS:ERASE:DIMA(16),D(4)
20 PRINTTAB(9);"NUMBER PUZZLE",TAB(9);"-
--------------"
250 A$="LRUD":M=0:D(1)=-1:D(2)=1:D(3)=-4
:D(4)=4
310 FORI=1TO15:A(I)=I:NEXTI
340 A(16)=0
350 P=16
360 R=FNR(10)+12
370 FORW=1TOR:S=0
390 N=FNR(3)+1
400 IFABS(D)=4THEND=D(FNR(2)+1):GOTO420
410 D=D(INT(FNR(2)+3))
420 GOSUB820
430 IFN<>0ANDS=1THENN=0:S=0:GOTO420
440 IFN=0ANDS=1THENS=0:D=-D:GOTO420
450 NEXTW
460 PRINT
480 W=0:PRINT
520 CLS:PRINT"SOLVABLE IN";R;" MOVES":PR
INT:PRINT"+-----------------------+"
530 PRINT"!                       !"
```

```
540 M=M+1
550 FORI=0TO12STEP4
560 PRINT"!";
570 FORJ=1TO4
580 IFA(I+J)=0THENPRINT"        ";:P=I+J:GO
TO610
590 IFA(I+J)<10THENPRINT"     ";A(I+J);:GO
TO610
600 PRINT"    ";A(I+J);
610 NEXTJ
620 PRINT"!"
630 PRINT"!                              !"
640 NEXTI
650 PRINT"+----------------------+"
660 FORI=1TO15
670 IFA(I)=ITHENNEXTI:GOTO1040
680 PRINT
690 PRINT"MOVE #";M;
700 INPUT" WHAT IS YOUR MOVE ";Q$
710 FORI=1TO4:IFLEFT$(Q$,1)=MID$(A$,I,1)
THEN780
720 NEXTI:GOTO700
780 PRINT:D=D(I)
790 GOSUB810
800 GOTO520
810 N=VAL(RIGHT$(Q$,1))
820 IFP-(D*N)>0 AND P-(D*N)<17 AND P-D>0
 AND P-D<17 THEN870
830 IFW<>0THENS=1:RETURN
840 SOUND1,110,15:FORI=0TO200:NEXTI:SOUN
D0:GOTO700
870 C=1
880 IFABS(D)=1THEN980
890 IFN<>0THEN920
900 IFP-(C*D)>0ANDP-(C*D)<17THENC=C+1:GO
TO900
910 N=C-1
920 FORI=1TON:A(P)=A(P-D):A(P-D)=0:P=P-D
:NEXTI
970 RETURN
980 E=INT((P-1)/4)*4+1
990 IFP-(N*D)<E OR P-(N*D)>E+3 OR P-D<E
OR P-D>E+3 THEN830
1000 IFN<>0THEN920
1010 IFP-(C*D)>=E AND P-(C*D)<E+4THENC=C
+1:GOTO1010
1020 N=C-1
1030 GOTO920
1040 M=M-1
1050 PRINT"WELL DONE THAT TOOK";M;"GOES"
1060 END
```

Edumaths

Here is your chance to answer those critics who claim your Sega is only good for games. This program does +, −, × and ÷, and you can set the level of difficulty. The program is straightforward and doesn't need any instructions.

Notes:

20-80 Enter type of topic.

150-370 Multiplication.

380-470 Summary.

480-620 Division.

670-900 Addition.

2000 Correct routine.

3000 Wrong routine.

4000 Input answer routine.

5000 If you make two errors the routine is called.

6000 Enter range of numbers routine.

7000 Generate random numbers dictated by 730.

8000 Print question no routine.

K = Answer.
E, F = Two parts to question.
W = No, wrong.
C = No, correct.
D = Highest number range (0 is lowest).
Q = Question no.
G = Entered answer.
A, I = General variables.

```
Edumaths

20 CLS:CURSOR10,5:PRINT"1] MULTIPLY",TAB
(10);"2] DIVISION",TAB(10);"3] ADDITION"
,TAB(10);"4] SUBTRACTION"
30 CURSOR7,10:INPUT"CHOOSE A TOPIC.";Y$
40 IFY$="1"THEN150
50 IFY$="2"THEN480
60 IFY$="3"THEN700
70 IFY$="4"THEN940
80 GOTO30
150 GOSUB6000
190 FORQ=1TO10:CLS:GOSUB7000
```

```
230 GOSUB8000:CURSOR9,10:PRINT"Multiply"
;E;"  X";F:K=E*F
240 GOSUB4000
260 IFG<>KTHEN310
270 C=C+1
280 GOSUB2000
293 NEXTQ
300 GOTO380
310 GOSUB3000
320 W=W+1
330 GOSUB4000
340 IFG<>KTHENGOSUB5000:GOTO293
370 GOTO280
380 CLS:CURSOR6,10:PRINT"Out of TEN you
got":PRINT:PRINTTAB(8);C;" Correct":PRIN
T:PRINTTAB(8);W;" Wrong":PRINT:PRINT
440 IFC=10THENPRINT"Excellent score keep
 it up..!!!":FORA=0TO6:FORB=500TO550STEP
5:SOUND1,B,15:NEXTB,A:SOUND0:GOTO460
450 IFW>5 THENPRINT"You should try an ea
sier level.."
455 FORA=1TOC:SOUND1,1000,15:FORI=0TO75:
NEXTI:SOUND0:NEXTA:FORA=1TOW:SOUND1,110,
15:FORI=0TO75:NEXTI:SOUND0:NEXTA
460 CURSOR8,19:PRINT"Press any key":IFIN
KEY$=""THEN460
470 GOTO20
480 GOSUB6000
485 FORQ=1TO10:GOSUB8000
490 GOSUB7000:IFF=0THEN490
560 CURSOR9,10:PRINT"Divide";E*F;" /";F:
K=E
570 GOSUB4000
580 IFG<>ETHEN630
590 C=C+1:GOSUB2000
610 NEXTQ
620 GOTO380
630 GOSUB3000:W=W+1
640 GOSUB4000
650 IFE<>GTHENGOSUB5000:GOTO670
660 GOSUB2000
670 GOTO610
700 GOSUB6000
740 FORQ=1TO10:GOSUB7000
750 GOSUB8000
780 CURSOR9,10:PRINT"Add";E;" +";F
790 GOSUB4000
800 K=E+F:IFG<>KTHEN860
810 C=C+1
815 GOSUB2000
820 NEXTQ:GOTO380
860 GOSUB3000
870 W=W+1
880 GOSUB4000
890 IFK<>GTHENGOSUB5000:GOTO820
900 GOTO815
940 GOSUB6000
950 FORQ=1TO10:GOSUB8000
960 GOSUB7000:IFF>ETHEN960
965 CURSOR9,10:PRINT"Subtract";E;" -";F
970 GOSUB4000:K=E-F
1050 IFG<>KTHEN1100
1060 C=C+1
1070 GOSUB2000
1080 NEXTQ:GOTO380
1100 GOSUB3000
1110 W=W+1:GOSUB4000
1150 IFK<>GTHENGOSUB5000:GOTO1080
1160 GOTO1070
2000 CURSOR7,15:PRINT"*****.CORRECT.+++*
*":FORA=500TO1300STEP50:SOUND1,A,15:NEXT
:FORA=15TO0STEP-.25:SOUND1,,A:NEXTA:RETU
RN
3000 CURSOR5,15:PRINT"  wrong try again
```

```
!!":FORA=300TO110STEP-20:SOUND1,A,15:NEX
TA:FORA=15TO0STEP-.25:SOUND1,,A:NEXTA:CU
RSOR5,15:PRINT"                    ":R
ETURN
4000 CURSOR1,21:INPUT"answer >> ";G:RETU
RN
5000 IFK<>GTHENCURSOR5,3:PRINT"The corre
ct answer was";K:SOUND1,110,15:SOUND2,11
1,15:SOUND3,113,15:FORA=0TO500:NEXTA:FOR
A=15TO0STEP-.5:SOUND1,,A:SOUND2,,A:SOUND
3,,A:NEXTA:RETURN
6000 CLS:W=0:C=0:CURSOR1,21:INPUT"Enter
the range of numbers 0-";B:RETURN
7000 E=INT(RND(8)*B):F=INT(RND(8)*B):RET
URN
8000 CLS:CURSOR0,0:PRINT"Question #";Q:R
ETURN
```

Horror Music

This program produces very eerie horror-style music. To operate just depress keys. To get very low notes press CTRL as well as letters, and to get very high notes hold down ENG DIER's, also shift works.

Notes:

10 See if a key is pressed.

20 Convert the key pressed to a number. Set all 3 sound channels to highest volume and offset the tones a little to create a droning effect.

30 If a key is still being depressed go back to 30.

40 Start to decrease the volume, if a key is pressed go back to 10.

50 Decrease volume and go back to 10.

An offshoot of the program is the next program, Lament of the Wind Wizard. You will probably be learning more of this soon, in a future program.

```
Horror Music

10 A$=INKEY$:IFA$=""THEN10
20 A=ASC(A$):SOUND1,109+A*2,15:SOUND2,11
0+A*2,15:SOUND3,112+A*2,15
30 IFINKEY$<>""THEN30
40 FORA=15TO0STEP-.3:SOUND1,,A:SOUND2,,A
:SOUND3,,A:IFINKEY$<>""THEN10
50 NEXT:GOTO10
```

```
Lament of the Wind Wizard

10 DATA408,1,344,13,408,2,344,13.2,408,1
3.2,64,9,408,7,64,13.2,26,9.3,344,12.9,4
6,13.5,64,9.3,408,12.6,26,13.5,64,1,408,
13.2,26,13,46,13.2,64,9,408,4.8,46,13.2,
64,9.9,408,13.2,64,13.5,26,13.2,46,5.7,6
4,13.5,46,13.1,26,13,10,11,408,3,408,2,3
44,0
20 RESTORE10:FORA=0TO32
30 READB,C:B=B+109
40 FORD=DTO15:SOUND1,B,D:SOUND2,B+1,D:SO
UND3,B+3,D:NEXT
50 FORD=15TOCSTEP-.40:SOUND1,B,D:SOUND2,
B+1,D:SOUND3,B+3,D:NEXTD,A
```

Base 10 to any base

This program converts numbers from base 10 to any other base. First input your number, then the base you wish to convert to. The program will then convert the number, showing its calculations.

Notes:

10 A$ will hold the converted number.

20 Enter number.

30 Enter desired base.

40 Reprint data.

50 Because A is altered its value is stored in C.

60-100 Calculate digits by dividing number by base and keeping remainder. Print base remainder.

110 Repeat if unfinished.

120 Print answer.

```
Base 10. to any base

10 A$="":INPUT"Number ";A
20 INPUT"Base to convert to? ";B
30 CLS:PRINT"Decimal #";A,,,
40 PRINT"Base";TAB(10);"Number";TAB(20);
"Remainder"
50 C=A
60 B$=CHR$(48+A-INT(A/B)*B)
70 PRINTB;TAB(10);A;TAB(20);
80 Z=ASC(MID$(B$,1,1)):IFZ>57THENZ=Z+7
90 PRINTCHR$(Z):A$=CHR$(Z)+A$
100 A=INT(A/B)
110 IFA>0THEN60
120 PRINT:PRINTC;" In base";B;" Is ";A$
```

Prime Number Tester

A prime number is a number not divisible by any number except 1 and itself. When you enter a number the program will tell you if it is prime or not. It does this by counting from 3, 5, 7 etc. up to the square root of the number in question, and trying to divide the numbers. If the result is zero, the number is not prime.

Notes:

10 Enter number.

20 Check if number is divisible by 2.

30 Count from 3 to square root of N. The reason for only counting up to the square root is that Mathematically, divisors of a number (if any exist) lie, one below the square root and one above. Also the program counts 3, 5, 7, 9 ... not 3, 4, 5, 6, 7, 8 because even numbers need not be tested. The N MOD D bit is the same as if N MOD D <> 0 then NEXT.

40 Not prime.

50 Prime.

My highest prime number is 9,812,811,031 which takes about 16 minutes to evaluate. It is not possible to evaluate 11 digit numbers due to inaccuracies in the ROM (in fact the Sega is much more accurate than most home micros).

```
Prime Number Tester

10 INPUT"ENTER A NUMBER ";N
20 IFN/2=INT(N/2)THEND=2:GOTO40
30 FORD=3TOSQR(N)STEP2:IF N MOD D THENNE
XTD:GOTO50
40 PRINTN;" ISN'T A PRIME.DIV BY";D:GOTO
10
50 PRINTN;" IS A PRIME NUMBER":GOTO10
100 REM

        HI=9812811031
                                    TOOK
ABOUT 16 MIN. TO WORK OUT!
```

Bubble Sort

Bubble sorting is a method of sorting into alphabetical order up to 2000 elements. It works on a simple method, image 5 elements:

MERLIN
FRED
HELLO
COMPUTER
COMPETE

Now MERLIN goes after FRED

MERLIN	FRED
FRED	MERLIN
HELLO	HELLO
COMPUTER	COMPUTER
COMPETE	COMPETE

etc ...

FRED	FRED	FRED	COMPETE
HELLO	HELLO	HELLO	COMPUTER
MERLIN	COMPUTER	COMPUTER	FRED
COMPUTER	MERLIN	COMPETE	HELLO
COMPETE	COMPETE	MERLIN	MERLIN

Notes:

10 Set up array. C is a count of number of data.

20 Enter data.

30 Check to see if "999" is entered.

40 Enter B$ into A$, increment C, goto 20.

50-100 Sort.

110 Print out re-arranged data.

```
Bubble Sort

10 DIMA$(2000):C=0
20 INPUT"ENTER DATA,999=END ";B$
30 IFB$="999"THEN50
40 A$(C)=B$:C=C+1:GOTO20
50 CLS:PRINT"SORTING..."
60 FORN=1TOC
70 IFA$(N-1)>A$(N)THENZ$=A$(N-1):A$(N-1)
=A$(N):A$(N)=Z$
80 NEXTN
90 FORN=1TOC:IFA$(N-1)>A$(N)THEN60
100 NEXTN
110 FORN=1TOC:PRINTA$(N):NEXTN
```

Noughts & Crosses

Unlike many programs of this ilk it IS possible to beat this version once in a while, although draws and losses are more probable. Some moves will be made more or less instantly, while others may take a few seconds. The strategy used is to look for a winning move or if no winning move is possible, look for a possible winning move by you and block it. If no such move is found a move is chosen at random.

Notes:

10 Data for positioning of X's and O's on Lines screen.

20-60 Prompt and display grid.

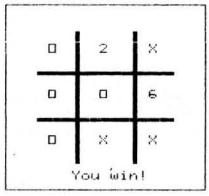
80-140 Your go.

150-260 Computer's go. The essence of the computer's "intelligence" is Lines 180 and 280. The computer reads this data and uses it to know where to place its move. E.g. if position 2 and position 1 both have a cross, then the next move will be position 3.

310-370 See if a win, loss or draw.

380 Erase routine.

```
Noughts and Crosses
```




```
10 DATA87,49,119,49,150,49,87,84,119,84,
150,84,87,119,119,119,150,119:RESTORE10:
ERASE
20 SCREEN 2,2:CLS:X1=0:DIMB$(9):CLS:PRIN
T"DO YOU WANT FIRST GO (Y/N)?":FORA=1TO9
:B$(A)="  ":NEXT
30 A$=INKEY$:IFA$=""THEN30
40 IFA$<>"Y"ANDA$<>"N"THEN30
50 CLS
60 LINE(60,20)-(185,160),1,B:FORA=104TO1
40STEP36:LINE(A,40)-(A+2,140),10,BF:NEXT
:FORA=70TO106STEP36:LINE(80,A)-(165,A+2)
,,BF:NEXT:FORA=1TO9:READX,Y:CURSORX,Y:CO
LOR4:PRINTHEX$(A):NEXT
70 IFA$="N"THEN150
80 GOSUB380:CURSOR102,146:COLOR13:PRINT"
Your go":BEEP:BEEP
```

```
90 A$=INKEY$
100 IFA$<"1"ORA$>"9"THEN90
110 IFB$(VAL(A$))>" "THEN90
120 B$(VAL(A$))="O"
130 RESTORE10:FORO=1TOVAL(A$):READX,Y:NE
XT:BLINE(X,Y)-(X+8,Y+8),,BF:CURSORX,Y:CO
LOR6:PRINT"O":FORA=210TO250STEP10:SOUND1
,A,15:NEXT:FORA=15TO0STEP-1:SOUND1,,A:NE
XT
140 A$="OOO":GOTO280
150 GOSUB380:CURSOR110,146:COLOR1:PRINT"
My go":BEEP:BEEP
160 IFB$(5)=" "THENF=5:GOTO250
170 FORB=1TO2:C$=MID$("XO",B,1)+MID$("XO
",B,1)
180 DATA2,4,1,2,6,3,4,8,7,6,8,9,1,5,9,1,
9,5,5,9,1,3,5,7,5,7,3,3,7,5,1,2,3,1,3,2,
2,3,1,4,5,6,5,6,4,4,6,5,7,8,9,7,9,8,9,8,
7,1,4,7,4,7,1,1,7,4,2,5,8,2,8,5,5,8,2,3,
6,9,3,9,6,6,9,3
190 RESTORE180:FORA=1TO106STEP3:READD,E,
F:IF(B$(D)+B$(E)=C$)ANDB$(F)=" "THEN250
200 NEXTA,B
210 A$="":FORA=1TO9:IFB$(A)=" "THENA$=A$
+STR$(A)
220 NEXT
230 F=VAL(MID$(A$,(INT(RND(8)*LEN(A$))+1
),1))
240 IFB$(F)<>" "THEN230
250 B$(F)="X"
260 RESTORE10:FORO=1TOF:READX,Y:NEXT:BLI
NE(X,Y)-(X+8,Y+8),,BF:CURSORX,Y:COLOR12:
PRINT"X":FORA=500TO550STEP10:SOUND1,A,15
:NEXT:FORA=15TO0STEP-1:SOUND1,,A:NEXT
270 A$="XXX"
280 DATA1,2,3,4,5,6,7,8,9,1,4,7,2,5,8,3,
6,9,1,5,9,3,5,7
290 RESTORE280:FORO=0TO7:READA,B,C:R$=""
:R$=R$+B$(A)+B$(B)+B$(C):IFR$=A$THEN340
300 NEXT
310 X1=X1+1:IFX1=9THEN370
320 IFA$="XXX"THEN80
330 IFA$="OOO"THEN150
340 IFA$="OOO"THENGOSUB380:COLOR4:CURSOR
102,146:PRINT"You win!"
350 IFA$="XXX"THENGOSUB380:COLOR8:CURSOR
108,146:PRINT"I win!"
360 FORDE=1TO500:NEXTDE:GOTO10
370 GOSUB380:CURSOR111,146:PRINT"DRAW":F
ORDE=1TO500:NEXTDE:GOTO10
380 BLINE(102,146)-(160,154),,BF:RETURN
```

The Dictator

You are a dictator in a village for a duration of 10 years. Each year you can buy and sell land, sow your land with corn, and feed the peasants. If you do not feed your people properly — 10 bags of corn per person per year — some will starve and if too many die the survivors might rebel and kill you! Charming!

Every acre requires 1 bag of corn to be sown. The harvest from the land is your only income, unless you go in for real estate — selling and buying land at different prices. Beware of rats — they love corn!

Notes:

10 Set up variables.

20-100 Display current status.

120-200 Enter land sold.

210-280 Enter land bought.

290-430 Enter acres to be sown.

440-500 How many bags to feed peasants with?

510-590 Calculations.

600-650 Your people rebel!

660-750 10 year report.

760-770 Too little land.

780 Erase a line. Uses QQ.

790-810 Checks syntax of inputs and converts to a number.

820-830 Too little corn.

840 Print a line routine.

```
THE DICTATOR
-----------------------------------
POP. OF CITY IN YR. 1 IS 100
  0 FOLK CAME TO THE CITY
  0 FOLK STARVED!!
YOU HAVE 1125 BAGS OF CORN
AND 1075 ACRES OF LAND
CORN YIELD 5 BAGS PER ACRE
LAND COSTS 12 BAGS PER ACRE
RATS GOBBLED 1000 BAGS OF CORN
-----------------------------------

HOW MANY ACRES DO YOU SELL? 0
HOW MANY ACRES BOUGHT? 75
HOW MANY ACRES TO BE SOWN? 975
HOW MANY BAGS TO FEED POP.? 1099

-----------------------------------
```

```
10 R=RND(-1):P=100:Y=1:SP=0:TS=0:NP=0:C=
3000:A=1000:H=5:L=INT(RND(8)*5)+10:R=100
0:AP=0:SC=0
20 CLS:PRINT"THE DICTATOR":GOSUB840
30 PRINT"POP. OF CITY IN YR.";Y;" IS";P
40 PRINTNP;" FOLK CAME TO THE CITY"
50 PRINTSP;" FOLK STARVED!!"
60 GOSUB820
70 PRINT"CORN YIELD";H;" BAGS PER ACRE"
80 PRINT"LAND COSTS";L;" BAGS PER ACRE"
90 PRINT"RATS GOBBLED";R;" BAGS OF CORN"
100 GOSUB840
110 CURSOR0,17:GOSUB840
120 CURSOR0,12:INPUT"HOW MANY ACRES DO Y
OU SELL? ";A$:GOSUB790
130 IFZ=1THEN160
140 IFAS<=ATHEN190
150 GOSUB760
160 QQ=12
170 GOSUB780
180 GOTO120
190 A=A-AS:C=C+AS*L
200 GOSUB820
210 CURSOR0,13:INPUT"HOW MANY ACRES BOUG
HT? ";A$:GOSUB790
220 IFZ=1THEN250
230 IFAS*L<=CTHEN270
240 GOSUB830
250 QQ=13:GOSUB780
260 GOTO210
270 A=A+AS:C=C-AS*L
280 GOSUB820
290 CURSOR0,14:INPUT"HOW MANY ACRES TO B
E SOWN? ";A$:GOSUB790
300 IFZ=1THEN330
310 IFAS<=ATHEN350
320 GOSUB760
330 QQ=14:GOSUB780
340 GOTO290
350 IFAS<=CTHEN380
360 GOSUB830
370 GOTO330
380 IFAS<=P*10THEN410
390 CURSOR0,19:PRINT"YOU ONLY HAVE";P;"
WORKERS"
400 GOSUB770:GOTO330
410 C=C-AS
420 GOSUB820
430 AP=AS
440 CURSOR0,15:INPUT"HOW MANY BAGS TO FE
ED POP.? ";A$:GOSUB790
450 IFZ=1THEN480
460 IFAS<=CTHEN500
470 GOSUB830
480 QQ=15:GOSUB780
490 GOTO440
500 C=C-AS:GOSUB820
510 SP=0:IFP*10=ASTHEN550
520 SP=P-INT(AS/10)
530 P=P-SP:TS=TS+SP
540 IFSP>P*(RND(8)*5+10)/10THEN600
550 Y=Y+1:IFY=11THEN660
560 H=1+INT(RND(8)*5):C=C+AP*H:R=0
570 IFC>=10000THENR=C-10000
580 R=R+INT(C*ABS(RND(8)-.5)):C=C-R:NP=I
NT(RND(8)*30):P=P+NP:L=INT(RND(8)*5)+10
590 CLS:GOTO20
600 CLS:PRINT"YOUR PEOPLE HAVE REBELLED
DUE TO YOUR LACK OF KNOWLEDGE OF NUTRITI
ON..!!":GOSUB840:PRINT"IN YOUR SHORT DIC
TATORSHIP YOU STARVED";TS;" PEOPLE":GOSU
B840
610 PRINT"YOU WILL BE EXECUTED AT DAWN..
."
```

```
620 GOSUB840
630 PRINT"ANOTHER GO..?(Y/N)"
640 IFINKEY$="Y"THEN10
650 GOTO640
660 CLS:PRINT"10 YEAR REPORT..":GOSUB840
670 PRINTTS;" PEOPLE STARVED"
680 IFA<1000THENPRINT"YOU SOLD";1000-A;"
 ACRES OF LAND"
690 IFA>=1000THENPRINT"YOU BOUGHT";A-100
0;" ACRES OF LAND"
700 GOSUB840
710 SC=ABS(INT(100*((150-TSP)/150)*(A/15
00)*(P/150)))
720 GOSUB840:PRINT"YOUR SCORE";SC;" "
730 A$="LOUSY":IFSC>=20THENA$="PATHETIC"
:IFSC>=30THENA$="NOT BAD":IFSC>=55THENA$
="GOOD":IFSC>=70THENA$="EVER THOUGHT OF
ENTERING POLITICS.?"
740 PRINT"RATING:",,A$
750 GOTO620
760 CURSOR0,19:PRINT"YOU ONLY HAVE";A;"
ACRES OF LAND"
770 FORI=0TO300:NEXTI:QQ=19:GOSUB780:RET
URN
780 SOUND1,110,14:FORI=0TO37:CURSORI,QQ:
PRINT" ";:NEXTI:SOUND0:RETURN
790 IFA$=""THENZ=1:RETURN
800 FORI=1TOLEN(A$):IFMID$(A$,I,1)<"0"OR
MID$(A$,I,1)>"9"THENZ=1:RETURN
810 NEXTI:Z=0:AS=VAL(A$):RETURN
820 CURSOR0,5:PRINT"YOU HAVE";C;" BAGS O
F CORN ":PRINT"AND";A;" ACRES OF LAND
 ":RETURN
830 CURSOR0,19:PRINT"YOU ONLY HAVE";C;"
BAGS OF CORN":GOTO770
840 PRINT"----------------------------------
--":RETURN
```

The Land of Sorcery

In this game you must do a very simple task — save a Princess! It sounds really easy when Bing the King tells you, but you don't know where she is, how to find her or if she's still alive! You also have to fight various nasty, malevolent beings and avoid traps! Good stuff!!

The only clues I'll give are the following (don't read them if you intend to solve the game).

1. You have to find a rare band of metal first. This enables you to see the door to the room in which the Princess is held.
2. You then have to go to a cold place with the Princess.

Okay, scalpels to the ready, let's dissect the program.

Notes:

10 Defines a random function called R. Because random nos are used all the time it is easier to define a function and keep referring to this e.g. for a number between 1-10 we would say LET A = FNR (10). See pages 134-135 of operator's manual.

30-110 Sets up the game. Of the arrays (dim statements) D$ is the description of the various areas (see Lines 530, 540 & 550). M is the direction and movements from a given area. (See Lines 560 & 570). The way this works is very simple. Look at the first 6 bits of data in Line 560. This data is the movement data for location no. 1 (a grassy meadow). It is arranged in the form of N, S, E, W, Up and Down, so if you N from location 1 you will go to location 8, south to 2, east to 3, west goes nowhere as does up and down. Okay!? Good!
OB$ is the array holding all the items around the land (see Line 580).
CO$ is the objects being carried.
The Key cannot be placed in rooms 19, 12, 7, 4 or 16 because these rooms or areas are trapped! The Princess is always in room 20.
PO is room marker — it holds your current position.
PF — if this ever equals 1, you have got the Princess.
KF — if this ever equals 1 you have found the Key.
DE — if this equals 1, you're dead!
PF, KF & DE are called flags.
V$ is a list of all the commands available — North, South, East, West, Up, Down and take.
MC is a count of monsters chopped up!
C holds the number of items carried.

120 Prints your current location. Remember PO and D$? The part, if DE THEN 400 is the same as if DE = 1 THEN 400. This is a check to see if you're dead.

130 Tells you what you see.

140 What exits are available. It works by scanning M and V$, if the value of M(PO) <> 0 then there is an exit and that exit is read from V$. So V$ is dual purpose!

160 If a random no. from 1-6 equals 1 a monster is encountered. Those who play Dungeons & Dragons will recognise this as a wandering Monster roll.

170 Awaits next command. F = 0, if F stays as 0 the command accepted is rejected and a jump is made to Line 210.

230-290 Movement. Firstly a check is made to see if you can go in that direction. If you can the PO variable is changed to the new room or area. If PO is 4, 16, 12 or 7, DE is set to 1, signifying you are dead!
If you are in room # 20, PF = 1 signifying you have found the Princess. If you have the Princess (PF = 1) and you are in position # 22, you have won!

300-390 Take. Firstly a check is made to see if you can take anything.

By the way, if PO = 2 you are by the tall tree and you can't take that!
If you are already carrying 3 items (C = 3) (which is the maximum) you have to drop something.
If you have found the Key, then you can see the door to the Princesses' cell (M[19,2]=20), and KF is set to 1 showing you have the Key.
Line 370 looks complicated but it is just a swapping mechanism. It is a temporary string.
Line 380 checks to see if you have dropped the Key.

400-440 Death, score and another go?

450-500 Monster attacks. Select a baddy (see Line 509). Create your hit-points and the monsters. HP = Its, H = Yours. Line 460 dictates who strikes first. If Your/Its hit points reach 0 You/It are no more!

510-520 Game end.

530 Data.

```
Welcome to the Land of Sorcery
------------------------------------------
You are.....
In a grassy meadow
You can see
a golden ring
Exits are:NSE
Now What..T
It's yours
Now What..N
Okay..
------------------------------------------
You are.....
In a damp cave
You can see
Nothing of interest
Exits are:NSD
Now What..


The Land of Sorcery

10 DEFFNR(X)=INT(RND(8)*X)+1
20 CLS:PRINT"Welcome to the Land of Sorc
ery"
30 ERASE:DIMD$(22),M(22,5),OB$(22),CO$(3
)
40 RESTORE:FORI=1TO22:READD$(I):NEXT
50 RESTORE 560:FORI=1TO22:FORJ=0TO5:READ
M(I,J):NEXTJ,I
60 FORI=1TO22:IFFNR(10)<6THENRESTORE 580
:Q=FNR(18):FORJ=1TOQ:READA$:NEXT:OB$(I)=
A$
70 NEXT I
80 OB$(20)="skeleton dressed in Royal ro
bes"
90 C=-1:A=FNR(19):OB$(A)="Golden Key":IF
A=19 OR A=12 OR A=7 OR A=4 OR A=16THEN90
100 OB$(2)="tall tree"
110 PO=1:PF=0:KF=0:V$="NSEWUDT":DE=0:MC=
0
120 PRINT"------------------------------
```

```
---------You are.....",,D$(PO):IFDETHEN4
00
130 PRINT"You can see":A$="Nothing of in
terest":IFOB$(PO)<>""THENA$="a "+OB$(PO)
140 PRINTA$:PRINT"Exits are:";:FORI=0TO5
:IFM(PO,I)<>0THENPRINTMID$(V$,I+1,1);
150 NEXTI:PRINT
160 IFFNR(6)=1THEN450
170 INPUT"Now What..";Q$:F=0
180 FORI=1TO7:IFMID$(Q$,1,1)=MID$(V$,I,1
)THENF=I
190 NEXTI
200 ONF+1GOTO210,230,230,230,230,230,230
,300
210 A$="What..?":IFFNR(10)<5THENA$="Try
again..!"
220 PRINTA$:GOTO170
230 IFM(PO,F-1)<>0THEN250
240 PRINT"You can't go that way...!":GOT
O170
250 PRINT"Okay..":PO=M(PO,F-1)
260 IFPO=4 OR PO=16 OR PO=12 OR PO=7THEN
DE=1
270 IFPO=20THENPF=1
280 IFPO=22ANDPFTHEN510
290 GOTO120
300 IFOB$(PO)=""ORPO=2THENPRINT"There is
 nothing to take..!":GOTO170
310 IFC=3THEN340
320 C=C+1:CO$(C)=OB$(PO):PRINT"It's your
s":OB$(PO)="":IFCO$(C)<>"Golden Key"THEN
170
330 KF=1:M(19,2)=20:GOTO170
340 PRINT"You are carrying too much.What
 do you want to leave behind?":FORI=0TO3
:PRINTI;"]";CO$(I):NEXTI:PRINT4;"]";OB$(
PO)
350 INPUTQ:Q=INT(Q):IFQ<0ORQ>4THEN610
360 IFQ=4THEN170
370 T$=CO$(Q):CO$(Q)=OB$(PO):OB$(PO)=T$:
PRINT"Okay.."
380 KF=0:M(19,2)=0:FORI=0TO3:IFCO$(I)="G
olden Key"THEN330
390 NEXTI:GOTO170
400 PRINT"HA..HA..HA,You are dead..."
410 PRINT"Final Score:";:T=(C+1)*1000+(M
C*2000):IFPFTHENT=T+5000
420 PRINTT:PRINT:PRINT"Another go?"
430 IFINKEY$<>"Y"THEN430
440 GOTO10
450 PRINT"All of a sudden you are attack
ed by a":RESTORE590:Z=FNR(5):FORA=1TOZ:R
EADA$:NEXTA:PRINTA$:HP=FNR(20)+3:H=FNR(2
0)+5
460 IFFNR(10)>5THEN490
470 PRINT"The ";A$;" hits you":H=H-FNR(5
):IFH<5THENPRINT"You don't look too good
!":IFH<1THENPRINT"In fact...":GOTO400
480 FORI=0TO500:NEXT
490 PRINT"You hit the ";A$:HP=HP-FNR(6):
IFHP<5THENPRINT"It doesn't feel too good
":IFHP<1THENPRINT"In fact it's dead!!!":
MC=MC+1:GOTO170
500 FORI=0TO500:NEXT:GOTO470
510 PRINT"All of a sudden a blinding fla
sh comesfrom the blue sky.The skeleton y
ou     carry falls to the floor,& it chan
ges into a beautiful Princess!!":PRINT:P
RINT"You've saved her...!!!",,,:PRINT"CO
NGRATULATIONS",,,,:PRINT:GOTO410
520 END
530 DATAIn a grassy meadow,In a dark For
est,On a small beach,Drowning in the sea
,Amongst sand dunes,In the rolling hills
```

```
,falling off a high hill,In a damp cave,
In a putrid dungeon,In a torture chamber
.A dead body is onthe rack,In a supply r
oom
540 DATAIn a cell.The door just slammed
behindyou...You're TRAPPED!,In an office
.The desk is smashed,In the lower dungeo
n,In the even lower dungeon,In a dark ro
om...all of a sudden you fall down a pi
t,A guard house,In a barrocks
550 DATAIn a dining hall.All the chairs
are smashed up..as is the table!,A sma
ll cell,In a tall tree.To the far North
you see a range of hills & to the East
a deep-dark sea,Up a mountain.It's ver
y cold up here!
560 DATA8,2,3,0,0,0,1,0,0,0,21,0,5,2,4,1
,0,0,0,0,0,0,0,0,0,6,3,0,0,0,0,7,5,7,22,0,
0,0,0,0,0,0,0,0,8,1,0,0,0,9,10,0,0,0,8,14,
11,9,0,13,0,0
570 DATA0,10,0,0,0,12,0,0,0,0,0,0,0,0,9,10
,0,0,0,0,0,0,0,0,9,15,0,17,0,16,14,0,0,0,0
,0,0,0,15,0,0,0,18,0,0,19,17,17,0,17,0,0
,0,9,0,0,0,0,0,19,0,0,0,0,0,0,0,2,0,6,0,
0,0,0
580 DATAgolden ring,bronze shield,dead r
at,bottle,mirror,ruby,diamond,brass key,
Grey cloak,staff,scroll,Crown,candle,med
allion,rope,bag,set of rusted armour,cor
pse
590 DATAMad Goblin,Crazy Orc,Vampire Bat
,Drooling Ghoul,Zany Zombie
```

Statistics Package

The sheer thought of STATISTICS makes me cringe and sweat! That's why I wrote this programme. After a few hours of searching through my bedroom for a couple of stats books, and clearing away the dust, I began writing it. The result of my labours (that's a laugh for a start!) is a program that allows you to enter data, graph the data, get 2 types of deviation, the mean, sum of data and sum of data squared. If you are going to use the graph you are restricted to 25 units of data — the reason for this is to keep the data clear when displayed in graph form.

Notes:

10 S is sum of data. SS is sum of data squared.

20-60 Do you want to graph the data? If you do MX (maximum amount of data) is set to 25, else set to 1000.

70-100 Clear screen. Dimension array to hold data after erasing previous info. Set count to zero. Input data andcheck if "Z" has been entered.

110 Convert data (which is numerical) to a number and store it in the array. Add to S and SS. Increment counter. Check to see if data count is less than maximum permissible.

120-220 Print statistical data. Wait for a key press.

230 If A$ = "Y" then you wanted to use the graph (look at Line 40).

240-270 Do you want to enter more info?

280-380 Set up graph.

390-460 Press F to see data. Q to rerun.

```
Statistics Package

10 S=0:SS=0
20 CLS:PRINT"DO YOU WANT TO USE GRAPH ?"
30 A$=INKEY$
40 IFA$="Y"THENMX=25:GOTO70
50 IFA$="N"THENMX=1000:GOTO70
60 GOTO30
70 CLS:PRINT"ENTER DATA",,"----------":P
RINT:PRINT"Z-STOP":PRINT:C=0:ERASE:DIMD(
MX+1)
80 PRINTC;" ";
90 INPUT"DATA ";B$
100 IFB$="Z"THEN120
110 D(C)=VAL(B$):S=S+D(C):SS=SS+(D(C)*D(
C)):C=C+1:IFC<=MXTHEN80
120 CLS:PRINT"# OF DATA";TAB(20);C
130 PRINT"SUM OF DATA";TAB(20);S
140 PRINT"SUM OF DATA SQUARED";TAB(20);S
S
150 AV=S/C
160 SD=SQR((SS-((S*S)/C))/(C-1))
170 PD=SQR((SS-((S*S)/C))/C)
180 PRINT"ARITHMETIC MEAN";TAB(20);AV
190 PRINT"STANDARD DEVIATION";TAB(20);SD
200 PRINT"POP. STD. DEVIATION";TAB(20);P
D
210 PRINT:PRINT:PRINT"PRESS ANY KEY TO C
ONTINUE"
220 IFINKEY$=""THEN220
230 IFA$="Y"THEN280
240 PRINT:PRINT"MORE DATA..? (Y/N)"
250 IFINKEY$="Y"THEN10
260 IFINKEY$="N"THENEND
270 GOTO250
280 SCREEN 2,2:CLS:T=D(0):B=T
290 FORA=0TOC
300 IFD(A)>TTHENT=D(A)
310 IFD(A)<BTHENB=D(A)
320 NEXTA
330 LINE(25,10)-(25,190),8:LINE(1,180)-(
254,180)
340 CURSOR8,182:PRINTINT(B):CURSOR8,10:P
RINTINT(T):CURSOR8,86:PRINTINT(T/2+B/2)
350 ST=220/C:S=ST+18
360 FORA=0TOC-1:IFS+ST>255THENST=(S+ST)-
255
370 LINE(S,180-((D(A)-B)/(T-B)*180))-(S+
ST,180-((D(A+1)-B)/(T-B)*180)),4:S=S+ST
380 NEXTA
390 SCREEN 1,1:CLS
400 PRINT"PRESS F TO SEE DATA,Q TO RE-RU
N"
410 IFINKEY$="F"THENGOSUB440
420 IFINKEY$="Q"THEN10
430 GOTO410
440 SCREEN 2,2
450 IFINKEY$="F"THEN450
460 SCREEN 1,1:RETURN
```

Phase Change

The following short program is designed to show constructive and destructive interference between sinusoidal wares. It uses polychrome high-resolution graphs to great effect. Imagine saying that after a few to drink! Basically, all it does is plot the result of two sin wares, which are in or out of phase, and tells you the maximum point reached. If it sounds a mouthful, just run the program and all will become clear (hopefully!).

Notes:

10 Input distance between crests of the two waves (in radians).

20-50 Set up axis and print phase change. Note the "_" 's in Line 50 are in fact pi symbols found at the bottom right of the keyboard.

60 XF used to calculate maximum sum of waves.

70-110 PSET 3 waves. The 2 waves are red and blue. The third (the sum of the two waves) is green.

120 Every tenth point, plot a heavy dot.

140 Print maximum total.

150-170 Plot heavy dot.

When the program has finished the screen will revert to the text screen. To see the graph just press SHIFT and BREAK together.

Some examples of phase change are:

0 phase change: This will cause only 2 waves to be plotted as the red & blue waves are on top of one another. This is called "in phase".

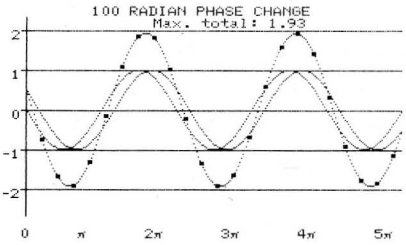
1.5 phase change: This is called "slightly out of phase".

2.6 phase change: Nearly out of phase.

Note that as the phase change gets close to Pi (3.1415926 ...) the resulting green wave gets flatter.

```
Phase Change

10 INPUT"INPUT PHASE CHANGE ";PH
20 SCREEN 2,2:CLS
30 FORF=0TO4:LINE(17,20+30*F)-(255,20+30
*F),8:NEXT:LINE(20,10)-(20,160)
40 T=2:FORF=0TO4:CURSOR5,20+30*F:PRINTT:
T=T-1:NEXTF
50 COLOR4:CURSOR55,2:PRINTPH;" RADIAN PH
ASE CHANGE":COLOR3:CURSOR17,170:PRINT"0
          2_        3_        4_        5_"
60 X4=0
70 FORF=1TOPI*75
```

```
80  X1=80+SIN(F/15)*30
90  X2=80+SIN(F/15+PH)*30
100  X3=X2+X1-80:IFX3>X4THENX4=X3
110  PSET(F+20,X1),8:PSET(F+20,X2),4:PSET
(F+20,X3),3
120  IFF/10=INT(F/10)  THEN150
130  NEXT  F
140  COLOR13:CURSOR100,11:PRINT"Max.  tota
l:";INT((X4-80)/3*100+.5)/1000:END
150  FORG=-1TO1:FORH=-1TO1
160  PSET(F+20+G,X3+H):NEXTH,G
170  GOTO130
```



Morse Code Trainer

The following short program allows you to create morse by using the keyboard. The dots and dashes are represented in binary as 0 and 1 respectively. For example, L is dot, dash, dot, dot which is 0100 in binary. This is then reversed to get 0010 and an extra bit is placed at the front, 10010, which is 18 in decimal.

Notes:

10	Clear screen. The CHR$(20) sets lower case.
20-30	Read in data to Z-array.
40-60	Wait for a key to be pressed. If it is not 0-9 or a-z then goto 40.
70	Print letter (or numeral) on screen and find data.
80	Perform binary division. This says whether a dot or dash is next.
90-100	Make a sound. D is the delay which is set at 10 if there is a dot, otherwise 38 for a dash.
110-120	Do some more division if N < 2, then the letter has been finished.

When it comes to learning morse make sure you don't learn dots and dashes. Instead learn the way each letter sounds. R is di, dah, dit. It is much easier, and you will gain speed much quicker.

```
Morse Code Trainer

10 CLS:PRINT"Morse trainer";CHR$(20)
20 DATA6,17,21,9,2,20,11,16,4,30,13,18,7
,5,15,22,27,10,8,3,12,24,14,25,29,19,63,
62,60,56,48,32,33,35,39,47,63
30 RESTORE:DIMZ(35):FORA=0TO35:READZ(A):
NEXT
40 A$=INKEY$:IFA$=""THEN40
50 P=ASC(A$)-96:IFP>=-48ANDP<=-39THENP=P
+75:GOTO70
60 IFP<10RP>122THEN40
70 PRINTA$;:N=Z(P-1)
80 X=N-INT(N/2)*2
90 SOUND1,1000,15:D=10:IFX=1THEND=38
100 FORI=0TOD:NEXT:SOUND1,,10:SOUND0
110 N=INT(N/2)
120 IFN<2THEN40
130 GOTO80
```

Machine-Code Programs

I am not going to go into the intricasies of M-code programming when I introduce the programs — I will only say that they work! They should be of use to those who want to produce games that involve scrolling the text screen in any of 4 directions.

To run them, all you do is enter them, run them — if no error message occurs SAVE them to tape. If an error does occur, check the data statements. Once the data is correct, new the program. The program (machine-code) is safe, only the BASIC goes. When you are ready to execute the machine code type in CALL&HF000 and the program will do what it's supposed to do!

For the more adventurous amongst you, you can actually get the screen to scroll in any diagonal direction! To do this you must load in 2 scroll routines. For example, we want to scroll diagonally towards the top left. So we firstly load the Up Scroll, run it, new it. Now load in the left scroll and change Line 20 to:

```
20  T = 0 : FOR A=&HF03B TO &HF069 : etc.
```

Run the program, then new it. From now on when you want diagonal scrolling just type:

```
CALL &HF000 : CALL &HF03B
```

This will give the effect of going towards the top left! This can be done with any combination giving diagonal scrolling (except for exact opposites like up & down).

The sound routines gives you the ability to create very fast, decreasing or increasing sound. The results can be quite spectacular. Anyone who has played "The House" will recognise the sounds, as this is the same routine I used. Enter the program, run it — if no errors occur save it to tape. From now on you can have either increasing or decreasing sound, with control of speed, tone and decay. To operate the "up" sound you call &HF036, "down" sound call &HF001. To take control of decay etc. you have to poke the data in. Here are the poke locations:

	Up	Down	Range of Values
Speed	&HF04F	&HF01A	0-100
Tone	&HF03B	&HF006	0-63 (anything higher gives weird
Decay	&HF02B	&HF02B	0-63 results)

The lower the speed, the quicker it runs, as with decay.

If this all seems a bit loopy, run the demo programs.

The last routine, which is loaded and run in the same way as the scroll routines, changes all the lower case letters. Exactly how I'm not going to say. Just load it, run it, type new and call &HF000 and look at the lower case letters.

```
M-Code Down Scroll

10 DATAF3,DB,BF,21,97,3F,1,98,3,11,BF,3F
,C5,E5,D5,CD,32,2C,DB,BE,8,E1,E5,CD,44,2
C,8,D3,BE,D1,E1,C1,1B,2B,B,78,B1,20,E5,2
1,2,3C,6,26,C5,E5,CD,44,2C,3E,20,D3,BE,E
1,23,C1,10,F2,C9
20 T=0:FORA=&HF000TO&HF03A:READA$:V=VAL(
"&h"+A$):T=T+V:POKEA,V:NEXT:IFT<>7417THE
NPRINT"ERROR"


M-Code Up Scroll

10 DATAF3,DB,BF,21,28,3C,11,0,3C,1,98,3,
C5,E5,D5,CD,32,2C,DB,BE,8,E1,E5,CD,44,2C
,8,D3,BE,D1,E1,C1,23,13,B,78,B1,20,E5,21
,9A,3F,6,26,C5,E5,CD,44,2C,3E,20,D3,BE,E
1,23,C1,10,F2,C9
20 T=0:FORA=&HF000TO&HF03A:READA$:V=VAL(
"&h"+A$):T=T+V:POKEA,V:NEXT:IFT<>7248THE
NPRINT"ERROR"

Machine Code - Left Scroll

10 DATAF3,DB,BF,21,3,3C,11,2,3C,6,17,C5,
6,26,C5,E5,D5,CD,32,2C,DB,BE,8,E1,E5,CD,
44,2C,8,D3,BE,D1,E1,13,23,C1,10,E8,23,23
,13,13,C1,10,DE,C9
20 T=0:FORA=&HF000TO&HF02D:READA$:V=VAL(
"&h"+A$):T=T+V:POKEA,V:NEXT:IFT<>5302THE
NPRINT"ERROR"
```

Machine Code - Right Scroll

```
10 DATAF3,DB,BE,21,BE,3F,11,BF,3F,6,18,C
5,6,26,C5,E5,D5,CD,32,2C,DB,BE,8,E1,E5,C
D,44,2C,8,D3,BE,D1,E1,1B,2B,C1,10,E8,2B,
2B,1B,1B,C1,10,DE,C9
20 T=0:FORA=&HF000TO&HF02D:READA$:V=VAL(
"&h"+A$):T=T+V:POKEA,V:NEXT:IFT<>5732THE
NPRINT"ERROR"
```

Machine Code - Sound

```
10 DATA0,3E,90,D3,7F,6,3F,AF,32,0,F0,3E,
8F,D3,7F,3A,0,F0,3C,32,0,F0,D3,7F,21,1,3
,2B,7C,B5,20,FB,10,E9,6,10,3E,90,D3,7F,F
5,21,1,F,2B,7C,B5,20,FB,F1,3C,10,F1,C9
20 DATA3E,90,D3,7F,6,3F,78,32,0,F0,3E,8F
,D3,7F,3A,0,F0,3D,32,0,F0,D3,7F,21,1,3,2
B,7C,B5,20,FB,10,E9,C3,22,F0
30 T=0:FORA=&HF000TO&HF059:READA$:V=VAL(
"&h"+A$):POKEA,V:T=T+V:NEXT:IFT<>9713THE
NPRINT"ERROR"
```

Invert

```
10 DATAF3,DB,BF,21,8,1A,11,8,1B,6,D0,C5,
E5,D5,CD,32,2C,DB,BE,8,E1,E5,CD,44,2C,8,
2F,D3,BE,D1,E1,C1,13,23,10,E7,C9
20 T=0:FORA=&HF000TO&HF024:READA$:C=VAL(
"&H"+A$):T=T+C:POKEA,C:NEXT:IFT<>4697THE
NPRINT"ERROR"
```

Mastermind

I know that you know how to play Mastermind, and I don't mean sitting in a black chair and getting a quick volley of brain shattering questions fired at you! I mean the peg game where you have to guess a code by using trial and error and deduction. Well this program is a faithful reproduction of that game. To play, all you have to do is guess the 4 digit code, enter your code and press the Carriage Return key, and the Sega will then evaluate your entry. If you get a result of say 2 black, 1 white, this means you have 2 of your digits in the right place, and 1 of them is the right number but in the wrong place. From this you evaluate that the other digit is completely wrong.

Notes:

10 Print header.

20-30 Set W and B to 0. These are the numbers of Whites and Blacks. Dimension arrays. Set up random code. When played the game produces numbers in the range 0-5. To increase the range, alter Line 30 to ... INT(RND[8]*10) ... this will be a much harder game, producing numbers in the range 0-9.

40 You have 18 chances at guessing (sorry, deducing!) the code.

50-70 Enter 4 digit code. If "R" is pressed reveal code. If your guess is not 4 digits long, re-enter.

80-200 Evaluate blacks and whites.

210 Print results.

220 If B = 4 (all black) jump to 260.

230 Loop back to 50.

240 Reveal code.

260-280 Smart aren't you? Press a key.

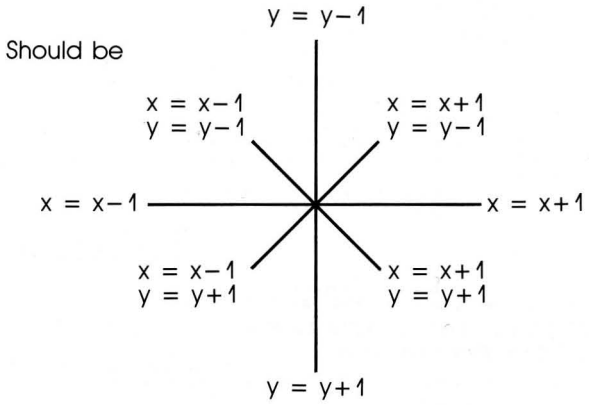
```
Master Mind...(?)

10 CLS:PRINT"Mastermind..",,,,"Code  Goe
s   Black   White"
20 W=0:B=0:ERASE:DIMC$(3),D$(3),T$(3)
30 FORA=0TO3:C$(A)=RIGHT$(STR$(INT(RND(8
)*6)),1):NEXT
40 FORN=1TO18
50 CURSOR0,21:INPUT"Enter a 4 digit code
 or R to reveal     ";R$
60 IFR$="R"THEN240
70 IFLEN(R$)<>4THEN50
80 FORA=1TO4:T$(A-1)=MID$(R$,A,1):NEXTA
90 FORA=0TO3:D$(A)=C$(A):NEXTA
100 W=0:B=0
110 FORA=0TO3:IFT$(A)<>D$(A)THEN140
120 D$(A)="C":G$(A)="D"
130 B=B+1
140 NEXTA
150 FORX=0TO3
160 FORZ=0TO3
170 IFD$(X)<>T$(Z)THEN200
180 D$(X)="A":T$(Z)="B"
190 W=W+1
200 NEXTZ,X
210 CURSOR0,21:PRINT"
                                    ":CURSOR0,N+2:P
RINTR$;TAB(7);N;TAB(13);B;TAB(20);W
220 IFB=4THEN260
230 NEXTN
240 CURSOR0,21:PRINT"Code  ";:FORA=0TO3:P
RINTC$(A);:NEXTA
250 GOTO270
260 CURSOR0,21:PRINT"Well  done"
270 IFINKEY$=""THEN270
280 GOTO10
```

Experiment

The following program is meant for 1 purpose only ... to give you some ideas for your own programs. If you're anything like me, it just takes a small idea to create a program. The hardest part is getting that original idea. So hopefully this will help you.

All the program is, is a little space ship in the middle of the screen. You can rotate it by using the "P" & "Q" keys, and move it by pressing the space bar.

You may need to consult the following bit of data. If your ship is at the point x,y then to move it in any of 8 directions the addition/subtraction to/from x/y is:



Notes:

10 Data for all 8 directions ship will face.

20 Read in data into A$. Set x & y.

30 Set direction (A = 0).

40 Set shape of ship. The shape is governed by A which is it's direction.

50 If B$ = "P", then B = −1. If B$ = "Q", then B = 1. If space bar is pressed then go to the subroutines from Lines 100-170. The subroutine jumped to is also governed by A.

60 Change direction (if any direction change).

70-90 Make sure that the ship goes "full circle" if made to.

100-170 Direction movement subroutines.

Experiment

```
10 DATA"030F3EFCFC3E0F03","10103383C7F7CF
0C0","C3E77E7E3C3C1818","08081C3CFE3E0F0
3","C0F07C3F3F7CF0C0","030F3EFE3C1C0808"
,"18183C3C7E7EE7C3","C0F07C7F3C381010"
20 RESTORE:DIMA$(7):FORA=0TO7:READA$(A):
NEXT:X=128:Y=96
30 A=0:SCREEN 2,2:CLS
40 PATTERNS#0,A$(A):SPRITE0,(X,Y),0,8
50 B$=INKEY$:B=(B$="P")-(B$="Q"):IFB$=""
"THENSOUND3,3000:SOUND4,3,8:ONA+1GOSUB10
0,110,120,130,140,150,160,170
60 A=A+B
70 IFA>7THENA=0
80 IFA<0THENA=7
90 SOUND0:GOTO40
100 X=X-1:RETURN
110 X=X-1:Y=Y+1:RETURN
120 Y=Y+1:RETURN
130 X=X+1:Y=Y+1:RETURN
140 X=X+1:RETURN
150 X=X+1:Y=Y-1:RETURN
160 Y=Y-1:RETURN
170 X=X-1:Y=Y-1:RETURN
```

Reverse

You don't need a PhD to play this game — it's dead easy! The object is to sort a string of randomly organised digits into the form 1 2 3 4 5 6 7 8 9 in as few moves as possible. To make a move, enter a single digit in the range 1-9. The computer will then reverse that number of digits, counting from the left.

Notes:

10 Clear screen and print heading.

20-70 Generate a random set of digits, making sure that each number appears only once.

80 Print out number 1-9.

90 Print out random arrangement.

100 Checks to see if game is completed.

110-120 Await a number in range 1-9. Change to a number.

130-150 Flip n number of numbers around.

160 Dunnit!

Reverse

```
10 CLS:PRINT"Reverse...",,"----------"
20 DIMA(9),B(9):N=0
30 FORI=1TO9
40 DI=INT(RND(8)*9)+1
50 IFA(DI)=0THENA(DI)=I:GOTO70
60 GOTO40
70 NEXTI
80 CURSOR3,5:FORI=1TO9:PRINTI;:NEXT
90 CURSOR3,3:FORI=1TO9:PRINTA(I);:NEXTI
100 FORB=1TO8:IFA(B)<A(B+1)THENNEXTB:GOT
O160
110 A$=INKEY$:IFA$<"1"ORA$>"9"THEN110
120 T=VAL(A$)
130 FORQ=1TOT:B(Q)=A(Q):NEXTQ
140 Z=T:FORQ=1TOINT(T/2):P=A(Q):A(Q)=A(Z
):A(Z)=P:Z=Z-1:NEXTQ:N=N+1:BEEP
150 GOTO90
160 PRINT:PRINT:PRINT:PRINT:PRINT"Well d
one you did it in";N;" goes"
```

Sprites

Listed are over 20 sprites to use in your games. They will serve only as ideas. For other ideas I suggest you purchase the following programs because they are riddled with sprites!

The House!
Cube-It
Mars Mobile
Munch Man

Here is how the data works. Let's say you want to define a Purple People Eater. You would write a computer program as follows:

```
10 PATTERNS#0, "C021...BF87"
20 PATTERNS#1, "1FA7...4F2F"
30 PATTERNS#2, "IC22...F0CS"
40 PATTERNS#3, "FA...E8"
50 MAG1: SCREEN 2, 2:CLS
```

Don't forget also that PPE's are always colour 13! The above program is only an example. I suggest you read pages 118-122 of the operator's manual.

Spider

```
0000001028479720
2F5C272828680818
0000000814E2E9B4
F43AE41414161018
```

Darth Vader's ship

```
0C1830306063677C
6763603030180C00
C0603030181898F8
981818303060C000
```

Pac ghost

030F0F1F1119111F
1F1F1F1F1F1B1111
C0F0F0F888C888F8
F8F8F8F8F8D88888

Kamikaze Combat Caterpillar

00E0B0F0311F2E0A
0000000000000000
003078CC96468B01
0000000000000000

Pac-man

00030F1F1F3F3F3E
3E3F3F1F1F0F0300
0080E0B0F8C00000
0000C0F8F0E08000

Skull & Crossbones

071F1F39393F1D07
67F2FC1F07FFFC60
E0F8F89C9CFCB8E0
E6AF3FF8E07F3F06

Crocodile

0000000030F3FFF
0000000000000000
0003E6ACF8F0E5FF
0000000000000000

Car

0304097F7FFF1408
0000000000000000
C0A010F8FCFC5020
0000000000000000

Dopy looking baddy

6090C864670F1919
1F1B0C0704081070
06091326E6F09898
F8D830E02010080E

Stupid Knight

0001061D2A2A2A1F
4CF7F01807023EFE
188F6511C9A9B1F3
7F9F31418181F9FD

Dopy looking goody

030F1F312D69717F
3E3F1F0000000000
80E0F018682C1CFC
F8F8F00000000000

Dalek

0307040C081F101F
103F203F607F40FF
0084FCC440E021E2
3CF310F018F808FC

Not-so-happy face

01070D1B3F332D69
313F3C1B0F070100
80E0B0D8FCCCB496
8CFC3CD8F0E08000

Fuzzzzzy

48449F7F3FB36D2D
73BF3F1B24494820
42543DFEFCCEB7B6
CCFEFCA6910A924C

Smiling Face

071F3F635DD1D1E1
FFFE5E67381C0700
E0F8FCC6BA8B8B87
FF7F7AE61C38E000

Boot

FE828E828E828E82
8E81828480F8F7F1
0000000000000000
788402010101FEFC

Purple People Eater

C02143271D45BF87
1FA74F176F8F4F2F
1C228CD1614EF0C5
FAC0FEC1E6E8E4E8

The Ghost

0103070587E7FFFF
FF8F8703030100000
80C0E0A0E1E7FFFF
FFF1E1C0C0C0E23C

Daft alien

01F3D79D0F0F0D06
0707060606 1E3E00
80CFEBB9F0F0B060
E0E0606060787C00

Swooping alien

80623B0F05070705
0C1C181010000000
028CB8E040C0C040
6070301010000000

Alf the alien

C06033171F98DB78
3F041C0838000000
0C1830A0E0646C78
F080E04070000000

Tank

017F031FFF402A1F
0000000000000000
80E0F0F8FF04A8F0
0000000000000000

Glossary

ALGORITHM — The series of steps the computer follows to solve a problem.

ALU — Arithmetic/logic unit, the part of the computer which does maths and where decisions are made.

ASCII — American Standard Code Information Exchange. 128 upper and lower case letters, digits and 31 special characters — literally the alphabet.

BASIC — Beginners All-purpose Symbolic Instruction Code. The most widely used computer language in use on microcomputers.

BAUD — Named after Baudot, a pioneer of telegraphic communication. Baud measures the rate of data transfer from tapes, disc drives, printers, etc. 1 Baud is 1 bit per second.

BENCHMARK — A measure of speed of a computer.

BINARY — A numbering system based on "0" 's and "1" 's.

BIT — BInary digiT. The smallest unit of data a computer can recognise.

BOOLEAN LOGIC — Use of AND, OR, NOT and XOR. Developed by George Boole.

BUG — An error in a program.

BUS — A number of conductors inside a chip or computer, used for sending and receiving data.

BYTE — In Sega's case 8 bits, capable of holding a number in the range of 0-255.

CAI — Computer Aided Instruction.

CAD — Computer Aided Design.

CAL — Computer Aided Learning.

CHIP — The general term for a small black box, with lots of little metal legs!

CPU — Central Processing Unit. The heart of a computer.

DATA — Information.

DEBUG — The removal of bugs from a program.

DYNAMIC MEMORY — A memory unit within a computer which loses its contents when the power is turned off.

FLIP-FLOP — A circuit which maintains one electrical condition until an input signal is received, when it then becomes the opposite condition. Also called Bi-Stable Vibrator.

GRAPHICS — Pictures as opposed to words.

HEXADECIMAL — Hex, a numbering system to base 16. Digits 0-9 are used as well as A, B, C, D, E, F. A = 10, B = 11 ... F = 15.

INTERFACE — Usually two "objects" (say a printer and a computer) can't "talk" to one another, so the interface acts as "interpreter". The most common interfaces are RS-232 and Centronics.

MACHINE CODE — An operation code which a processor can understand. All Basic programs are converted into machine code. A program written in M/C need not be "worked out" so it runs much quicker than a BASIC program.

MAINFRAME — Computers come in 3 sizes: Micro-computers — such as the Sega; Mini-computers — say a PDP-11; and the really big Mainframes — say CRAY-1 (which costs about US$15 million!).

PERIPHERAL — Anything which is joined onto a computer, and is controlled by the latter e.g. disc drives, printer, etc.

PORT — A socket through which data can be fed out of or into a computer e.g. joystick ports, TV port, etc.

PROGRAMMER — A degenerate race of social dropouts — tend to disappear for days on end, to perform a ritual called "programming". They tend to talk only to others of their species (the language, so far undeciphered by scholars, sounds like shorthand!). They keep the coffee companies in business and baffle psychologists.

PURPLE PEOPLE EATERS — PPE's. Dangerous, evil allies of Vanessa the Vampire. They are the sworn enemies of Humankind. They can be found in The House!, but only a few remain, others have been seen in a place called "The Crazy Crypt" seeking revenge. They have 8 arms and tend towards unneeded violence.

ROUTINE — A section of a program.

SEMI-CONDUCTOR — A material that is usually an electrical insulator but under specific conditions becomes a conductor e.g. silicon and germanium.

STATIC MEMORY — A memory which preserves its contents so long as power is on, but does not require additional boosts of power to keep its memory, unlike Dynamic memory which needs "refreshing".

VDU — Video Display Unit.

VOLATILE — Refers to memory which "forgets" its contents when power is off.