



CBM 64 PROGRAMS Volume 1



PROGRAMS BY
RICHARD FRANKLIN
EDITED BY NICK HAMPSHIRE

**CBM
64
PROGRAMS
Volume 1**

Programs
by

RICHARD FRANKLIN

Edited
by

NICK HAMPSHIRE



Duckworth

First published in 1984 by
Gerald Duckworth & Co Ltd,
The Old Piano Factory,
43 Gloucester Crescent, London NW1

© 1984 by Nick Hampshire

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publisher.

ISBN 0 7156 1882 2

British Library Cataloguing in Publication Data

Franklin, Richard

CBM 64 programs.

Vol. 1

1. Commodore 64 (Computer)—Programming

I. Title II. Hampshire, Nick

001.64'2

QA76.8.C64

ISBN 0-7156-1882-2

Printed in Great Britain by
Redwood Burn Ltd, Trowbridge

CONTENTS

Acknowledgment	4
Introduction	5
Star Trek	9
Hi-Res Graphics	
Principles of High Resolution Graphics.....	44
Multi-Colour Bar Chart	88
3 D Graph.....	91
Multi-Colour Pattern	93
Sine Graph	95
Demo of Hi-Res Routines 5 & 7	97
Digiclock	100
Disassembler	105
Hangman.....	109
Landmine	116
Fruit Machine	122
Car.....	131
Maze	135
Keyboard Synthesizer.....	139
Reserved Word Changer	146
+/- Lister	150
Large Characters	152
User Defined Characters.....	156
Machine Code to Data	159
Personal Information Retrieval.....	161
Will o' the Wisp	167
Index.....	190

ACKNOWLEDGMENT

Mr Richard Franklin, an avid Commodore user and professional programmer, is responsible for most of the programs reproduced in this book. Mr Franklin is active in the world of computing and is rapidly gaining a deserved reputation as a programmer of note.

INTRODUCTION

GRAPHICS CHARACTERS

The Commodore 64 has a very wide ranging and versatile control character set. These characters appear as reversed characters when listed in a program. When printed they have effects like moving the cursor, changing colours, etc.

We will explain how to get these characters and what they do when printed. You will probably know where to find some of the characters, but as the size of the character matrix on printers is slightly less than the 8x8 matrix on the screen, all the characters will look slightly different.









It would be very tedious to explain how to get every character available, therefore we will just explain the colour characters, cursor control characters, and the function key characters.

The first group of characters are the colour characters. When printed, they have the effect of changing the colour of the text to the required colour.



Being pressed with the CONTROL key held down:

1	Gives:	■	Which is for the colour black
2		□	Which is for the colour white
3		■	Which is for the colour red
4		■	Which is for the colour cyan
5		■	Which is for the colour purple
6		■	Which is for the colour green
7		■	Which is for the colour blue
8		■	Which is for the colour yellow




Being pressed with the CBM (bottom left) key held down:

- 1 Gives:  Which is for the colour orange
- 2  Which is for the colour brown
- 3  Which is for the colour pink
- 4  Which is for the colour dark grey
- 5  Which is for the colour med. grey
- 6  Which is for the colour lt. green
- 7  Which is for the colour lt. blue
- 8  Which is for the colour lt. grey




Two characters control whether the print is reverse or normal. Both of these are obtained by holding down the CONTROL key and pressing:

- 9 Gives:  Which turns reverse on
- 0  Which turns reverse off

There are six cursor control keys. These on their own:

-  By pressing the cursor down key
-  By pressing the cursor right key
-  By pressing the home key

And these with the SHIFT key held down:

-  By pressing the cursor down key (up)
-  By pressing the cursor right key (left)
-  By pressing the cursor home key (clear screen)

The last group of characters that appear in the listings are those characters that represent the function keys. The first four are from the key on its own:

```
F1 Gives: █
F3       █
F5       █
F7       █
```

And from the key being pressed with the SHIFT key held down:

```
F1 Gives: █ (F2)
F3       █ (F4)
F5       █ (F6)
F7       █ (F8)
```

MACHINE CODE ROUTINES

In this book there is a large section on hi-resolution graphics. This section explains how to plot points etc. on the 64. Examples are given in Basic but a collection of machine code routines accompanies them.

The machine code routines have been presented in two different formats, the first being a fully commented assembly listing (done using the Commodore Assembler Development System). This is there so that those who wish to understand how the routines are done, and who understand machine code, may look at it.

If you have the above mentioned assembler you could put them in using it. The majority of owners, however, will not understand assembly language and will not have the assembler. For those people, a loader in the form of a Basic program with the machine code in data has been included. To enter the machine code, type

in the Basic program and type RUN.

Once the routines have been entered, there are some sample programs to show how to use these routines.

STAR TREK

STAR TREK

The adventure game listed on the next few pages must be considered the ultimate experience for all you 'trekkies' out there.

The horrific news brought to all manner of living beings in the known galaxy by the interstellar video news system, has set men's (and other things'!) minds in turmoil. The Klingons have reneged on the peace treaty and are even now conducting raids upon the Federation's outlying bases. The Federation, in the face of the despicable behaviour of the Klingon Emperor, has had no alternative but to go to WAR! Captain James T. Kirk has been conspicuous only by his absence in this debacle and you!, yes, you the player have been chosen to lead the defense, captaining the Star Ship Enterprise, aided by the faithful retainer and trusty, half human, pointy eared First Officer Spock.

After a familiarisation sortie, you progressively take on more and more difficult tasks until almost unaided you are defending the Federation against the evil might of the Klingons.

RUNNING THE PROGRAM

Following the intense and pressurized familiarisation course that you followed at the Starfleet command, you realise that the task before you is immense and that the galaxy is a awesome battle ground, full of demonic Klingons and their powerful command ships. The most brilliant brains in the universe have instructed you, in the course of your studies, that to fight the battle the galaxy must be conceptualised as a 2-dimensional matrix, similar to a chequer board, and where each square represents one quadrant. Each quadrant, in turn, is divided into 100 sectors (10x10).

The Enterprise has been fitted with the most expensive, up-to-date technological equipment available in these war-torn times. To utilise these amazing examples of electronic wizardry, you must memorize the following commands and apply them intelligently to each exciting, do or die, battle that you encounter in your search for victory.

- 0 = Set course and move
- 1 = Short range sensor scan
- 2 = Long range sensor scan
- 3 = Fire phasers
- 4 = Fire photon torpedoes
- 5 = Damage control report
- 6 = Self destruct
- 7 = Abandon ship
- 8 = Activate energy shields
- 9 = Deactivate energy shields
- 10 = Fire experimental death ray
- 11 = Activate teleporter unit
- 12 = Engage impulse drives
- 13 = Maintain position and rest for one star date
- 14 = Print current position and stardate
- 15 = Print starchart of entire galaxy

Among the choice array of formidable space technology installed into the Enterprise, are several ingenious gadgets designed to make life easier when you find yourself in a sticky situation. For instance, the Star Chart, the brainchild of a particularly gifted space person, can be used to obtain a overall picture of the entire galaxy in which you struggle. Contained in it is all the information currently known about the Universe, painstakingly extracted by an allied team of space spies. It is, of course, in symbols, but even a Moon Moth could work them out. A symbol .01 means nothing is known. A .11 tells you that there is a star base, a very useful bit of information, BUT, there is also an unknown number of nasty, horrible Klingons and stars lurking about. Each digit means something if it is a whole number.

Now we get to the important information – the weapons and devices that will help you outwit the evil Klingon forces, in each of the tactical galactical battles that you will have to face in order to reach your desired goal of victory. However, I give you fair warning, the Klingon intelligence services have not been idle in their absence from galactical daily life and their weapon armoury, fighting skills and devilish cunning are not to be underestimated.

1. WARP ENGINES

COURSE = A real number from 1 to 8.999. Numbers indicate the direction in which you will go, starting at the right and progressing counter clockwise:

4 3 2
5-* -1
6 7 8

FACTOR = A real number from 0 to 12.

DISTANCE TRAVELLED = Integer (warp factor) quadrants:

Warp .25 = Travels 2 sectors
.5 = 4 sectors
1.25 = 1 quadrant
2.5 = 2 quadrants

For example, if you travel from quadrant 1-1, sector 1-1, in direction 1 at warp 1 you would stop at quadrant 1-1, sector 1-9, in the next stardate. However, when you change quadrants, the position of all the objects in the new quadrant (including the Enterprise) are randomised to simulate 3-d travel.

Carefully consider these notes, as they contain helpful hints that might just give you the edge in a confrontation with the enemy.

***NOTE: Every time you use the warp engines, precious time is used up. The amount of time used up is proportional to the warp factor used. For example, a warp factor of 4 uses 1 stardate. If the Enterprise is blocked by something during intra-quadrant travel, it will stop in front of it (wasting more of your time).

*** NOTE: There is a time portal available when you travel at warp 12. You may be carried either forward or backward in time, BUT there is always a .40 probability that you will be destroyed in a nuclear implosion, a nasty way to go. However, a warp factor of 12 may only be used after 9 solar years, so you will have killed off a large number of Klingons before you go.

***** NOTE:** If the Enterprise rams a Klingon vessel, either deliberately or accidentally, the Klingon will be destroyed and the Enterprise heavily damaged, making life very difficult for you.

2. MAINTAINING POSITION

There is a command available which can be used to maintain position. It uses .5 stardate and 50 units of energy. Repairs are also made to damaged devices, an invaluable service, as a damaged Enterprise can not operate at its optimum.

3. IMPULSE DRIVES

Impulse drives are a supplement to the warp engines and cost 400 units of energy to engage. There is a slight possibility of implosion.

4. TELEPORTATION UNIT

The teleporter instantaneously transports you to your base. It operates on power from the base. It can be used an infinite number of times, but not until the fifth solar year. Sometimes it fails!!!

5. SHORT RANGE SENSORS

These are invaluable and are the devices that give you, the astute Commander, the edge over the dreadful enemy. When activated, they provide a detailed view of the quadrant you are currently in. Like the Star Chart, the display is in symbols and has to be translated. The Enterprise is represented by an 'E' on the screen. The Klingon battle cruisers appear as a 'K', the Starbases, a 'B', the stars, a '*' and the occasional ultra powerful Klingon command ship may make an appearance, represented by a 'C'. So before taking any action, activation of the short range sensors will inform you of your position in relation to the surrounding possible dangers.

***** NOTE:** Docking is a two step procedure. First you have to move to an adjacent position to the base. Then conduct a short range scan. While docked, starbase shields protect the starship from any vicious attacks.

***** NOTE:** Condition yellow means you are dangerously low on energy and your advisors at base strongly recommend that you head back, at super speed. It would be false bravery for you to try and carry on.

6. EXPERIMENTAL DEATH RAY

The experimental death ray, as its name implies, is wholly unpredictable. However you can safely predict that in a given situation, it will do exactly as it likes. There is a good chance it will destroy some of the Klingons, but the side-effects could be interesting. It can probably only be fired once, but of course nothing is certain in outer space. It cannot be used until 8 solar years have passed.

7. LONG RANGE SENSORS

The long range sensors of the Enterprise display the number of objects in the 9 closest quadrants, with the Enterprise in the central one. Each digit of the number in each number means something.

The digits on ones are the number of Stars:

The tens—starbases.

The hundreds—Klingons.

The thousands – Supernovas.

For example:

319 means 3 Klingons, 1 base, and 9 stars.

206 means 2 Klingons, 0 bases, and 6 stars.

8. ENERGY SHIELDS

These can be invaluable when the situation becomes alarmingly one-sided. While they are activated, Klingon attacks and some space storms are nullified. Each time you move with your shields up, energy will be used in proportion to the warp factor. Shield strength is a measure of how much energy the shields can ward off during an attack. It will be printed after the shield status in a S.R. Scan. When attacked, the shield strength is lowered, and the effect is cumulative. When strength reaches 0, the shields are disabled. The ship's strength can be regained by docking.

9. PHASERS

These are part of the ship's active defense system. Any portion of the energy available can be fired, and your battle computer divides this amount among the Klingon cruisers in the quadrant. It also determines the various directions of fire. The effectiveness of a hit depends mostly on the distance of the target. A Klingon battle cruiser starts with 200 units of energy and it can fire an amount equal to whatever it has left. A double burst fires the same amount twice. If any Klingons are destroyed on the first burst, they will not be fired at again with the second burst, and the remaining surplus energy will be directed at any remaining Klingons.

10. PHOTON TORPEDOES

Use these carefully, as initially the supply is limited to 13 torpedoes. They are the most important weapons the Enterprise has and you should master their use in order to be successful in your mission. One torpedo usually destroys whatever it hits. The range of photon torpedoes (like phasers) is limited to the current quadrant. The course of the photon torpedo is set the same way as that of the Enterprise. You may, if you wish, fire a spread of three torpedoes at once. There is a time lag in launching them, and they are fired in counter clockwise sequence. For example, a torpedo course of 3.49 with a spread angle of 0.24 would fire torpedoes at directions 3.25, 3.49, and 3.73. A double burst fires two torpedoes in the same direction. A double burst with a spread of three each, can be fired with the spread angle varying for each burst.

11. DAMAGE CONTROL REPORT

The damage control report lists the main devices and their state of repair. A negative state of repair indicates a disabled device. Devices can be damaged in a large number of ways, usually as a result of collision or a particularly heavy Klingon attack. They can be repaired by a space storm or a truce, and are also repaired by 2

units every stardate.

With all these devices under your spacesuit belt you should prove a formidable opponent.

GOOD LUCK!!!

```

10 POKE53281,7:POKE53280,6:PRINT"☐";
20 PRINT"                3STAR TREK☐"
30 OPEN2,0:I9=RND(0)
40 PRINT"☐ BE CAREFUL AND HAVE A GOOD TIME"
50 DIM Q$(10),F$(12)
60 DIM G(8,8),Q(10,10),D(8),K(10,3),N(3),C(9,2)
70 DIM C0(8,8)
80 GOSUB 11200
90 PRINT"☐TO PLAY SHORT VERSION,";
95 PRINT"        TYPE 1 ----"
100 PRINT"☐TO PLAY REGULAR VERSION,        TYPE 2 ----"
110 PRINT"☐TO PLAY CHALLENGE VERSION, TYPE 3 ----"
120 INPUT#2,V$:PRINT:V=VAL(V$)
130 I=V*100
140 IF V>0 AND V<4 AND V=INT(V) THEN 160
150 GOTO 90
160 E0=5000:PRINT
170 E=E0:P0=13:P=13:S9=200:N1=1:R8=1:B7=0
180 DEF FNO(F)=SQR((K(F,1)-S1)^2+(K(F,2)-S2)^2)
190 GOSUB 7210
200 F=1:T0=INT(RND(1)*50+20)*100:T=T0
210 T9=25+INT(RND(1)*11)
220 FORI=1T07:D(I)=0:NEXT
230 S1=INT(RND(1)*10+1)
240 S2=INT(RND(1)*10+1)
250 REM SET UP GALAXY
260 B9=0:E4=0:W3=0:C5=0
270 Z$="DOWN":A9=3000:K9=0:L8=8
280 IF V<1THEN310
290 PRINT"☐ YOUR SHORTENED GALAXY IS 6 BY 6"
300 L8=6:T9=14+INT(RND(1)*11)
310 FORI=1T0L8
320 FORJ=1T0L8
330 IF V=2 THEN 470
340 IF V=3 THEN 460.
350 R=RND(1)*36
360 IF R<1.25 THEN K3=5:GOTO440
370 IF R<2.5 THEN K3=4:GOTO440
380 IF R<3.75 THEN K3=3:GOTO 440
390 IF R<5.0 THEN K3=4:GOTO 440
400 IF R<6.25 THEN K3=0:GOTO 440
410 IF R<7.5 THEN K3=2:GOTO 440
420 IF R<9.0 THEN K3=1:GOTO 440
430 K3=0
440 K9=K9+K3
450 GOTO 620
460 R8=1.85
470 REM DETERMINE NUMBER OF KLINGONS,
480 R=RND(1)*64 :REM BASES AND STARS
490 IF R<.10*R8 THEN K3=9:GOTO 590
500 IF R<.25*R8 THEN K3=8:GOTO 590

```

```

510 IF R<.5*R8 THEN K3=7:GOTO 590
520 IF R<1.00*R8 THEN K3=6:GOTO 590
530 IF R<1.5*R8 THEN K3=5:GOTO 590
540 IF R<3.00*R8 THEN K3=4:GOTO 590
550 IF R<6.28*R8 THEN K3=3:GOTO 590
560 IF R<9.00*R8 THEN K3=2:GOTO 590
570 IF R<14.28 THEN K3=1:GOTO 590
580 K3=0
590 K9=K9+K3
600 B3=RND(1)
610 IF B3>.97 THEN B3=1:C0(I,J)=.11:GOTO 630
620 B3=0:C0(I,J)=.01
630 B9=B9+B3
640 S3=INT(RND(1)*8+1)
650 G(I,J)=K3*100+B3*10+S3
660 NEXT J,I
670 K0=K9
680 IF B9>0 THEN 740
690 I=INT(RND(1)*L8+1)
700 J=INT(RND(1)*L8+1)
710 G(I,J)=G(I,J)+10
720 C0(I,J)=.11
730 B9=1
740 P2=1
750 Q1=INT(RND(1)*L8+1)
760 Q2=INT(RND(1)*L8+1)
770 PRINT"Q1PLEASE TYPE A SECRET PASSWORD :Q";
780 INPUT#2,X#:PRINT
790 N1=0:N9=0:N7=0:A7=0
800 REM DETERMINE KLINGON COMMANDER
810 J=INT(4.5*RND(1)*R8+1)
820 J1=INT(J/2)
830 IF J1=0 THEN 920
840 FORI=1TOJ1
850 R1=INT(RND(1)*L8+1)
860 R2=INT(RND(1)*L8+1)
870 M7=G(R1,R2)+100
880 IF M7>999 THEN 850
890 C(I,1)=R1:C(I,2)=R2
900 G(R1,R2)=G(R1,R2)+100
910 K9=K9+1:K0=K9:NEXT I
920 FORI=J1+1TOJ
930 R1=INT(RND(1)*L8+1)
940 R2=INT(RND(1)*L8+1)
950 IF G(R1,R2)<100 THEN 930
960 C(I,1)=R1:C(I,2)=R2
970 M7=G(R1,R2)+100
980 IF M7>999 THEN 930
990 G(R1,R2)=G(R1,R2)+100
1000 K9=K9+1:K0=K9:NEXT I

```

```

1010 REM CONTINUE
1020 H1=0:H2=INT(RND(1)*5)
1030 GOTO 12390
1040 REM SET UP QUADRANT
1050 K3=0:B3=0:S3=0
1060 IF Q1<1 THEN 1150
1070 IF Q1>8 THEN 1150
1080 IF Q2<1 THEN 1150
1090 IF Q2>8 THEN 1150
1100 X=G(Q1,Q2)/99.99
1110 K3=INT(X)
1120 B3=INT((X-K3)*10)
1130 S3=G(Q1,Q2)-INT(G(Q1,Q2)/9.999)*10
1140 FORI=0TO10:FORJ=0TO3:K(I,J)=0:NEXTJ,I
1150 FORI=0TO10:FORJ=0TO10:Q(I,J)=0:NEXTJ,I
1160 IF N1<>1 THEN 1190
1170 Q(S1,S2)=5
1180 GOTO 1200
1190 Q(S1,S2)=1
1200 IF K3<1 THEN 1400
1210 K5=K3
1220 FORI=1TO9
1230 IF Q1<>C(I,1) THEN 1310
1240 IF Q2<>C(I,2) THEN 1310
1250 K(K5,3)=600
1260 GOSUB 12340
1270 Q(R1,R2)=6
1280 K(K5,1)=R1
1290 K(K5,2)=R2
1300 K5=K5-1
1310 NEXT
1320 IF K5<=0 THEN 1400
1330 FORI=1TOK5
1340 GOSUB 12340
1350 Q(R1,R2)=2
1360 K(I,1)=R1
1370 K(I,2)=R2
1380 K(I,3)=S9
1390 NEXT I
1400 IF B3<1 THEN 1470
1410 FORI=1TOB3
1420 GOSUB 12340
1430 Q(R1,R2)=3
1440 B1=R1
1450 B2=R2
1460 NEXT I
1470 IF S3<1 THEN 1520
1480 FORI=1TOS3
1490 GOSUB 12340
1500 Q(R1,R2)=4
1510 NEXT I

```

```

1520 A=-1
1530 IF P2=1 THEN 1570
1540 GOSUB 6020
1550 IF E<=0 THEN 6430
1560 GOTO 2100
1570 P2=P2+1
1580 REM SHORT RANGE SENSOR SCAN
1590 T8=S1-1
1600 U=S1+1
1610 M=S2-1
1620 N=S2+1
1630 IF T8<1 THEN T8=1
1640 IF U>10 THEN U=10
1650 IF M<1 THEN M=1
1660 IF N>10 THEN N=10
1670 FOR I=T8 TO U
1680 IF J=0 THEN 1700
1690 FOR J=M TO N
1700 IF Q(I,J)<>3 THEN 1740
1710 C$="DOCKED":POKE53280,1
1720 GOSUB 10280
1730 GOTO 1830
1740 NEXT J
1750 NEXT I
1760 IF K3>0 THEN 1800
1770 IF E<E0*.1 THEN 1820
1780 C$="GREEN":POKE53280,5
1790 GOTO 1830
1800 C$="RED":POKE53280,2
1810 GOTO 1830
1820 C$="YELLOW":POKE53280,7
1830 IF A>=0 THEN 1850
1840 GOSUB 6020
1850 IF E<=0 THEN 6430
1860 IF D(2)>=0 THEN 1900
1870 PRINT"SHORT RANGE SENSORS ARE OUT"
1880 PRINT
1890 GOTO 2100
1900 PRINT"-----"
1910 C0(Q1,Q2)=G(Q1,Q2)
1920 FOR I=1 TO 10
1930 FOR J=1 TO 10
1940 PRINTQ$(Q(I,J))" ";
1950 NEXT J
1960 GOSUB 2000
1970 NEXT I
1980 PRINT"-----"
1990 GOTO 2100
2000 IF I=1 THEN PRINT"STARDATE "INT(T):RETURN
2010 IF I=2 THEN PRINT"CONDITION "C$:RETURN

```

```

2020 IFI=3THENPRINT"QUADRANT "Q1"- "Q2:RETURN
2030 IFI=4THENPRINT"SECTOR  "S1"- "S2:RETURN
2040 IFI=5THENPRINT"ENERGY  "INT(E):RETURN
2050 IFI=6THENPRINT"TORPEDOES"P:RETURN
2060 IFI=7THENPRINT"KLINGONS "K9:RETURN
2070 IFI=8THENPRINT"SHIELDS "Z$, "INT(A9):RETURN
2080 PRINT
2090 RETURN
2100 IF N9<>1 THEN 2140
2110 GOSUB 10790
2120 IF N7=1 THEN 5540
2130 N9=2
2140 PRINT"COMMAND :";
2150 INPUT#2,A$:PRINT:A=VAL(A$)
2160 IF C$<>"DOCKED" THEN 2180
2170 GOSUB 10280
2180 REM COMMAND LINK
2190 IF A>15 THEN 2250
2200 IF A<0 THEN 2250
2210 IF A>10 THEN 2240
2220 IF A=1 THEN PRINT"1";
2225 IF A>5 THEN 2235
2230 ON (A+1) GOTO 2280,1580,3650,3900,4320,5760
2235 ON (A-5) GOTO 11250,9040,10430,10600,8270
2240 ON (A-10) GOTO 11910,11640,7300,8050,8110
2250 PRINT"INVALID COMMAND !"
2260 PRINT
2270 GOTO 2100
2280 REM WARP DRIVE
2290 PRINT"DCOURSE :";
2300 INPUT#2,C1$:PRINT:C1=VAL(C1$)
2310 IF C1=0 THEN 2100
2320 IF C1<1 THEN 2140
2330 IF C1>=9 THEN PRINT"9":GOTO 2290
2340 PRINT"WARP FACTOR (0-12) :";
2350 INPUT#2,W1$:PRINT:W1=VAL(W1$)
2360 IF W1<=0 OR W1>12 THEN 2140
2370 IF W1<=.25 OR D(1)>=0 THEN 2410
2380 PRINT"WARP ENGINES ARE DAMAGED, MAXIMUM"
2390 PRINT"SPEED = WARP .25"
2400 GOTO 2290
2410 C$="MOVING":POKE53280,12
2420 GOTO 9370
2430 E=E-W1*C5/4-C5/4
2440 IF E<=0 THEN 6430
2450 IF W1<.2 THEN 2470
2460 W9=RND(1)
2470 IF W1=12 THEN 5360
2480 REM
2490 GOSUB 6020
2500 IF N9=1 THEN 2100

```

```

2510 IF E<=0 THEN 6450
2520 FOR I=1 TO 7
2530 IF D(I)=0 THEN 2570
2540 D(I)=D(I)+W1/2
2550 IF D(I)<0 THEN 2570
2560 D(I)=0
2570 NEXT I
2580 IF RND(1)>.25 THEN 3080
2590 R1=INT(RND(1)*7+1)
2600 IF RND(1)>.5 THEN 2940
2610 IF C#<>"DOCKED" THEN 2680
2620 PRINT"*** SPACE STORM, ";
2630 PRINT"STARBASE ENVIRONMENTAL"
2640 PRINT"PROTECTORS ACTIVATED. ALL HARMFUL"
2650 PRINT"RADIATION AND ENERGY FACTORS ARE "
2660 PRINT"ABSORBED."
2670 GOTO 3080
2680 IF Z#="UP" THEN 2760
2690 D(R1)=D(R1)-(RND(1)*5+1)
2700 PRINT"*** SPACE STORM , "F$(R1)" DAMAGED."
2710 IF F$(R1)<>"ENERGY SHLDS" THEN 3080
2720 C5=0
2730 PRINT"*** ENERGY SHIELDS ARE DOWN ***"
2740 Z#="DOWN"
2750 GOTO 3080
2760 IF R1=7 THEN 2830
2770 PRINT"*** SPACE STORM ,ENERGY SHIELDS";
2780 PRINT"WARD OFFDAMAGING EFFECTS"
2790 A9=A9-(1044*RND(1)+50)
2800 IF A9>0 THEN 3080
2810 PRINT"*** SHIELD ENERGY EXHAUSTED ***"
2820 GOTO 2720
2830 PRINT"***** SEVERE SPACE STORM , "F$(?)
2840 PRINT"DAMAGED *****"
2850 PRINT"*** ENERGY SHIELDS ARE DOWN ***"
2860 Z#="DOWN"
2870 C5=0
2880 A9=A9-1000
2890 D(?)=D(?) - 3*RND(1)
2900 R1=INT(RND(1)*6+1)
2910 D(R1)=D(R1)-3*RND(1)
2920 PRINT"*** "F$(R1)" ALSO DAMAGED ***"
2930 GOTO 3080
2940 FOR I=R1 TO 7
2950 IF D(I)<0 THEN 3020
2960 NEXT I
2970 IF R1<2 THEN 3010
2980 FOR I=1 TO R1-1
2990 IF D(I)<0 THEN 3020
3000 NEXT I
3010 GOTO 3080

```



```

3020 R1=I
3030 D(I)=D(I)+RND(1)*5+1
3040 IF D(I)<=0 THEN 3060
3050 D(I)=0
3060 PRINT"*** TEMPORARY TRUCE, "F$(I)" STATE"
3070 PRINT"OF REPAIR IMPROVED ***"
3080 N=INT(W1*8)
3090 T=T+(INT(W1*2.5+.5))/10
3100 IF V<>1 THEN 3120
3110 T=T+(INT(W1*2.5+.5)/23)
3120 E=E-2*N
3130 Q(S1,S2)=0
3140 X=S1
3150 Y=S2
3160 IF T>T0+T9 THEN 6430
3170 GOSUB 5330
3180 IF N<1 THEN 3270
3190 FOR I=1 TO N
3200 S1=S1+X1
3210 S2=S2+X2
3220 X3=INT(S1+.5)
3230 Y3=INT(S2+.5)
3240 IFS1<.50RS2<.50RS1>=10.5ORS2>=10.5THEN3460
3250 IF Q(X3,Y3)<>0 THEN 3280
3260 NEXT I
3270 GOTO 3420
3280 Z5=Q(X3,Y3)
3290 IF Z5<>6 THEN 3340
3300 Z3=01
3310 Z4=02
3320 GOSUB 11820
3330 GOTO 7450
3340 IF Z5<>2 THEN 3360
3350 GOTO 7450
3360 PRINT
3370 GOSUB 10710
3380 PRINTN$" BLOCKED BY OBJECT AT"
3390 PRINT"SECTOR: "INT(S1+.5)"-"INT(S2+.5)". "
3400 S1=S1-X1
3410 S2=S2-X2
3420 S1=INT(S1+.5)
3430 S2=INT(S2+.5)
3440 Q(S1,S2)=A5
3450 GOTO 2100
3460 Q1=INT(Q1+W1*X1+(X-.5)/10)
3470 Q2=INT(Q2+W1*X2+(Y-.5)/10)
3480 IF Q1<1 THEN Q1=1:Z2=1
3490 IF Q1>L8 THEN Q1=L8:Z2=1
3500 IF Q2<1 THEN Q2=1:Z2=1
3510 IF Q2>L8 THEN Q2=L8:Z2=1

```

```

3520 IF Z2=1 THEN GOSUB 12200
3530 PRINT
3540 IF Q1<>M1 OR Q2<>M2 THEN 3600
3550 Q1=INT(L8*RND(1)+1)
3560 Q2=INT(L8*RND(1)+1)
3570 IF G(Q1,Q2)>99 THEN 3550
3580 PRINT"☉SUPERNOVA SHOCK WAVE PUTS YOU IN"
3590 PRINT"QUADRANT"Q1"-"Q2"."
3600 IF G(Q1,Q2)>999 THEN 10230
3610 Z2=0
3620 S1=INT(RND(1)*10+1)
3630 S2=INT(RND(1)*10+1)
3640 GOTO 1040
3650 REM LONG RANGE SENSOR SCAN
3660 IF D(3)>=0 THEN 3700
3670 PRINT"☉LONG RANGE SENSORS ARE INOPERABLE"
3680 PRINT
3690 GOTO 2100
3700 PRINT"☐LONG RANGE SENSOR SCAN FOR QUADRANT"
3710 PRINTTAB(18)Q1"-"Q2"."
3720 PRINTTAB(10)"☐"
3730 FOR I=Q1-1 TO Q1+1
3740 PRINTTAB(10)" | "
3750 N$(1)=" -1";N$(2)=" -1";N$(3)=" -1"
3760 FOR J=Q2-1 TO Q2+1
3770 IF I<1 THEN 3830
3780 IF I>L8 THEN 3830
3790 IF J<1 THEN 3830
3800 IF J>L8 THEN 3830
3810 N$(J-Q2+2)=RIGHT$(" "+STR$(G(I,J)),4)
3820 C0(I,J)=G(I,J)
3830 NEXT J
3840 PRINTTAB(10)" |:"N$(1):"N$(2):"N$(3):" |"
3850 PRINTTAB(10)" | "
3860 PRINTTAB(10)" | "
3870 NEXT I
3880 PRINTTAB(10)"☐"
3890 GOTO 2100
3900 REM PHASER CONTROL
3910 IF D(4)>=0 THEN 3950
3920 PRINT"☉PHASER CONTROL IS DISABLED"
3930 PRINT
3940 GOTO 2100
3950 IF C5=0 THEN 3990
3960 PRINT"☉YOU MUST LOWER YOUR ENERGY SHIELDS ";
3970 PRINT"TO FIRE.☉"
3980 GOTO 2140
3990 H9=1
4000 PRINT"☉DO YOU WANT A DOUBLE BURST ?";
4010 INPUT#2,H1$:PRINT
4020 PRINT"PHASERS LOCKED ON TARGET. ENERGY"

```

```

4030 PRINT"AVAILABLE ="E
4040 PRINT"NO. OF UNITS TO FIRE ";
4050 IF H1#="YES" THEN 4070
4060 IF H1#<>"Y" THEN PRINT"?";GOTO 4090
4070 PRINT"WITH EACH BURST ?"
4080 H9=2
4090 INPUT#2,XX#:PRINT:X=VAL(XX#)
4100 IF X<=0 THEN 2100
4110 IF E-X*H9<=0 THEN 4020
4120 FOR H1=1 TO H9
4130 E=E-X
4140 IF K3=0 THEN 4300
4150 W2=K3
4160 FOR I=1 TO 10
4170 IF K(I,3)<=0 THEN 4260
4180 H=(X)/(FND(I))*(3.0#RND(1))
4190 K(I,3)=K(I,3)-(H/W2)
4200 PRINT" ]"H/W2"UNIT HIT ON KLINGON AT"
4210 PRINT"SECTOR"K(I,1)"-"K(I,2)
4220 PRINT"("<K(I,3)"LEFT)"
4230 IF K(I,3)>0 THEN 4260
4240 GOSUB 5880
4250 IF K9<=0 THEN 6610
4260 NEXT I
4270 NEXT H1
4280 IF K3<=0 THEN 4310
4290 GOSUB 6020
4300 IF E<=0 THEN 6430
4310 GOTO 2100
4320 REM PHOTON TORPEDOES
4330 H9=1:E4=0
4340 IF D(5)>=0 THEN 4370
4350 PRINT" ]PHOTON TUBES ARE NOT OPERATIONAL."
4360 GOTO 2100
4370 IF P>0 THEN 4400
4380 PRINT" ]ALL PHOTON TORPEDOES EXPENDED."
4390 GOTO 2100
4400 IF C5=0 THEN 4430
4410 PRINT" ]YOU MUST LOWER YOUR ENERGY SHIELDS ";
4415 PRINT"TO FIRE."
4420 GOTO 2140
4430 PRINT" ]DO YOU WANT A DOUBLE BURST ?";
4440 INPUT#2,H1#:PRINT
4450 IF H1#="YES" THEN 4470
4460 IF H1#<>"Y" THEN 4490
4470 IF P<2 THEN 4510
4480 H9=2
4490 PRINT" ]DO YOU WISH TO FIRE A SPREAD ?";
4500 INPUT#2,M#:PRINT
4510 PRINT" ]TORPEDO COURSE (1-8.9999) :";
4520 INPUT#2,H0#:PRINT:H0=VAL(H0#)

```

```

4530 FOR H1=1 TO H9
4540 IF H9<>2 THEN 4560
4550 PRINT"BURST NUMBER"H1
4560 C1=H0
4570 IF C1=0 THEN 2100
4580 IF C1<1 THEN 4510
4590 IF C1>9 THEN 4510
4600 IF P<3*H9 THEN 4630
4610 IF M$="YES" THEN 7860
4620 IF M$="Y" THEN 7860
4630 IF RND(1)<.92 THEN 4710
4640 PRINT"TORPEDO MISFIRES."
4650 C1=C1+RND(1)*8+1
4660 IF C1<9.0 THEN 4680
4670 C1=C1-8.0
4680 IF E4=0 THEN 4710
4690 PRINT"REMAINDER OF SPREAD ABORTED."
4700 E4=0
4710 GOSUB 5330
4720 X=S1:Y=S2
4730 P=P-1:U2=-1
4740 PRINT"TORPEDO TRACK  "
4750 X=X+X1
4760 Y=Y+X2
4770 IF X<.5 THEN 5220
4780 IF Y<.5 THEN 5220
4790 IF X>10.5 THEN 5220
4800 IF Y>10.5 THEN 5220
4810 X5=INT(X+.5)
4820 Y5=INT(Y+.5)
4830 PRINTX$TAB(19)"-"Y5
4840 U2=U2+1
4850 IF Q(X5,Y5)<>0 THEN 4870
4860 GOTO 4750
4870 F7=Q(X5,Y5)
4880 IF F7=1 THEN 4750
4890 IF F7=2 THEN 6800
4900 IF F7=3 THEN 5030
4910 IF F7=4 THEN 5030
4920 IF F7=5 THEN 4750
4930 IF F7=6 THEN 6790
4940 PRINT"*** KLINGON DESTROYED ***"
4950 K3=K3-1
4960 K9=K9-1
4970 IF K9<=0 THEN 6610
4980 FOR I=1 TO 9
4990 IF K(I,1)<>X5 THEN 5020
5000 IF K(I,2)<>Y5 THEN 5020
5010 GOTO 5140
5020 NEXT I
5030 Q(X5,Y5)=0

```

```

5040 PRINT"***STARBASE DESTROYED...";
5045 PRINT"CONGRATULATIONS**";
5050 B3=B3-1
5060 C$="RIDICULOUS":POKE53280,4
5070 GOTO 5190
5080 Q(X5,Y5)=0
5090 PRINT"*** STAR DESTROYED ***"
5100 S3=S3-1
5110 IF S3>=0 THEN 5190
5120 S3=0
5130 GOTO 5190
5140 K(I,3)=0
5150 Q(X5,Y5)=0
5160 IF F7<>6 THEN 5190
5170 Z3=Q1:Z4=Q2
5180 GOSUB 11820
5190 G(Q1,Q2)=K3*100+B3*10+S3
5200 C0(Q1,Q2)=G(Q1,Q2)
5210 GOTO 5230
5220 PRINT"TORPEDO MISSED."
5230 IF E4=0 THEN 5280
5240 E4=E4+1
5250 IF E4>3.5 THEN 5270
5260 GOTO 7930
5270 IF K3<=0 THEN 5310
5280 NEXT H1
5290 GOSUB 6020
5300 IF E<=0 THEN 6430
5310 E4=0
5320 GOTO 2100
5330 X2=COS((C1-1)*.785398)
5340 X1=-SIN((C1-1)*.785398)
5350 RETURN
5360 W9=RND(1)
5370 IF T>T0+8.99 THEN 5410
5380 PRINT"WARP 12 CANNOT BE USED UNTIL";
5390 PRINT"STARDATE"CHR$(13)T0+9
5400 GOTO 2290
5410 GOSUB 6020
5420 IF E<=0 THEN 65535
5430 PRINT"2SPEED APPROACHING WARP 12. ENGINES"
5440 PRINT"OPERATING ABOVE DANGER LEVEL. ALL ";
5450 PRINT"SAFETYDEVICES DISFUNCTIONAL. CRITICAL"
5460 PRINT"OVERHEATING--SPONTANEOUS INPLOSION"
5470 PRINT"IMMINENT."
5480 IF W9>=.40 THEN 5560
5490 FOR I=1 TO 4
5500 PRINT"*****";
5510 NEXT I
5520 GOSUB 10710

```

```

5530 PRINT " ** ENTROPY OF "N#" MAXIMIZED **"
5540 PRINT
5550 GOTO 12590
5560 IF W9>=.55 THEN 5620
5570 PRINT
5580 PRINT"YOU ARE TRAVELLING BACK IN TIME."
5590 T=T-9
5600 PRINT"*** STARDATE ="T
5610 GOTO 2480
5620 PRINT"YOU ARE TRAVELLING BACK IN TIME."
5630 T=T-14
5640 PRINT"*** STARDATE ="T
5650 GOTO 2480
5660 IF W9>=.70 THEN 5720
5670 PRINT"YOU ARE TRAVELLING FORWARD IN TIME."
5680 T=T+2
5690 IF T>T0+T9 THEN 6430
5700 PRINT"*** STARDATE ="T
5710 GOTO 2480
5720 PRINT
5730 PRINT"NUCLEAR REACTOR CHAIN REACTION ";
5735 PRINT"QUENCHED."
5740 PRINT"*** TIME PORTAL MISSED ***"
5750 GOTO 2480
5760 REM DAMAGE CONTROL REPORT
5770 IF D(6)>=0 THEN 5810
5780 PRINT"DAMAGE CONTROL REPORT IS NOT";
5785 PRINT"AVAILABLE."
5790 PRINT
5800 GOTO 2100
5810 PRINT"DEVICE"TAB(20)"STATE OF REPAIR"
5820 FOR I=1 TO 7
5830 PRINTF$(I);TAB(20)D(I)
5840 NEXT I
5850 PRINT
5860 GOTO 2100
5870 REM
5880 PRINT"*** KLINGON AT SECTOR"K(I,1)"-";
5890 PRINTK(I,2);"DESTROYED."
5900 X5=K(I,1)
5910 Y5=K(I,2)
5920 IF Q(X5,Y5)<>6 THEN 5960
5930 Z3=Q1
5940 Z4=Q2
5950 GOSUB 11820
5960 K3=K3-1
5970 K9=K9-1
5980 Q(X5,Y5)=0
5990 G(Q1,Q2)=K3*100+B3*10+S3
6000 C0(Q1,Q2)=G(Q1,Q2)

```

```

6010 RETURN
6020 REM KLINGON ATTACK
6030 GOSUB 10710
6040 IF K3<=0 THEN 6420
6050 IF C#<>"DOCKED" THEN 6080
6060 PRINT"STARBASE SHIELDS PROTECT THE "
6065 PRINTN$
6070 RETURN
6080 IF C5<>0 THEN 7040
6090 J=0
6100 FORI=1TO10
6110 J=J+K(I,3)
6120 NEXT I
6130 IF (J+200)<E THEN 6260
6140 IF N1<>0 THEN 6260
6150 PRINT"-----KLINGON COMMANDER TO ";
6155 PRINT"ENTERPRISE-----";
6160 PRINT"SO YOU WELL KNOW, YOU ARE ";
6165 PRINT"DANGEROUSLY"
6170 PRINT"LOW ON ENERGY AND WILL PROBABLY BE"
6180 PRINT"DESTROYED BY MY NEXT ATTACK. AS AN"
6190 PRINT"OFFICER AND A GENTLEMAN, ";
6195 PRINT"I OFFER YOU THE";
6200 PRINT"CHANCE TO SAVE YOURSELF AND CREW---"
6210 PRINT"DO YOU SURRENDER?";
6220 INPUT#2,S#:PRINT
6230 IF S#="YES" THEN 6250
6240 IF S#<>"Y" THEN 6260
6250 A7=1:N9=1:GOTO 6420
6260 U3=1
6270 PRINT
6280 FOR I=U3 TO 10
6290 IF K(I,3)<=0 THEN 6400
6300 H=(K(I,3)/FND(I))*(2+RND(1))
6310 E=E-H
6320 PRINT"10"H"UNIT HIT ON "N$" FROM"
6330 PRINT"KLINGON, SECTOR:"K(I,1)"-"K(I,2)
6340 IF H<=100 THEN 6390
6350 R5=INT(RND(1)*7+1)
6360 D(R5)=D(R5)-(H/200)*(1.5*RND(1))
6370 PRINT"11** CRITICAL HIT, ";
6380 PRINTF$(R5);" DAMAGED **";
6390 PRINT "("E"LEFT)"
6400 NEXT I
6410 PRINT
6420 RETURN
6430 REM LOSE
6440 PRINT"120 IT IS STARDATE"
6450 PRINT
6460 IF T<T0+T9 THEN 6490
6470 PRINT" TIME HAS RUN OUT AND "

```

```

6480 GOTO 6560
6490 GOSUB 10710
6500 PRINT" THE "N#" HAS BEEN DESTROYED"
6510 PRINT"      (ENERGY LEFT ="E"UNITS.)"
6520 IF K9>1 THEN 6560
6530 PRINT" ] THERE IS STILL ONE KLINGON BATTLE"
6540 PRINT"      CRUISER LEFT."
6550 GOTO 6570
6560 PRINT" ] THERE ARE STILL "K9"KLINGON BATTLE"
6570 PRINT" CRUISERS, THE FEDERATION WILL BE"
6580 PRINT"CONQUERED! YOU HAVE BEEN DESTROYED ";
6590 PRINT"- YOU      MISCALCULATED."
6600 GOTO 12590
6610 REM WIN
6620 PRINT" ]";
6630 PRINT"      IT IS STARDATE" T
6640 PRINT" ] THE LAST KLINGON BATTLE CRUISER IN"
6650 PRINT" THE GALAXY HAS BEEN DESTROYED. THE"
6660 PRINT" FEDERATION HAS BEEN SAVED.";
6665 PRINT" YOU HAVE BEEN";
6670 PRINT"      PROMOTED TO ]COMMANDER EMERITUS]. "
6680 IF N1<0 THEN 6720
6690 PRINT" ] ] "K0"KLINGONS IN" T-T0"YEARS."
6700 PRINT"      RATING ="INT(K0/(T-T0)*1000)
6710 GOTO 12590
6720 PRINT" ] ] "K0"KLINGONS IN" T-T0"YEARS."
6730 IF A7=5 THEN 6760
6740 PRINT" ]      ENTERPRISE CAPTURED."
6750 GOTO 12590
6760 PRINT" ] ENTERPRISE DESTROYED INTENTIONALLY."
6770 PRINT"      RATING ="INT(K0/(T-T0)*1000-5000)
6780 GOTO 12590
6790 REM HIT OR MISS COMMANDER
6800 IF RND(1)<(U2*4)/100 THEN 7010
6810 IF RND(1)>.4 THEN 4940
6820 FOR I=1 TO 10
6830 IF K(I,1)<>5 THEN 6870
6840 IF K(I,2)<>5 THEN 6870
6850 K(I,3)=K(I,3)-(580/(SQR(U2+1)))
6860 GOTO 6950
6870 NEXT I
6880 REM HIT OR MISS
6890 IF RND(1)<(U2*3)/100 THEN 7010
6900 IF RND(1)>.15 THEN 4940
6910 FOR I=1 TO 10
6920 IF K(I,1)<>5 THEN 6990
6930 IF K(I,2)<>5 THEN 6990
6940 K(I,3)=K(I,3)-0.4*K(I,3)-120*RND(1)
6950 IF K(I,3)<=0 THEN 4940
6960 PRINT" ]** KLINGON EVASIVE ACTION ---"
6970 PRINT"* KLINGON DAMAGED BUT NOT DESTROYED *"

```



```

6980 GOTO 5230
6990 PRINT
7000 GOTO 5230
7010 PRINT"0** KLINGON EVASIVE ACTION ---"
7020 PRINT"* TORPEDO MISSED *"
7030 GOTO 5230
7040 REM SHIELDS
7050 FORI=1TO10
7060 IF K(I,3)<=0 THEN 7170
7070 H=(K(I,3)/FNO(I))*(2*RND(1))
7080 A9=A9-H
7090 IF A9>0 THEN 7170
7100 Z$="DOWN"
7110 U3=I+1
7120 PRINT"0**KLINGONS ATTACK-ENERGY ";
7130 PRINT"SHIELDS KNOCKEDOUT**"
7140 PRINT
7150 C5=0
7160 GOTO 6280
7170 NEXT I
7180 PRINT"0**KLINGONS ATTACK-ENERGY ";
7190 PRINT"SHIELDS PROTECTTHE "N$" **"
7200 GOTO 6410
7210 REM SET UP DEVICE LIST
7220 F$(1)="WARP ENGINES"
7230 F$(2)="S.R. SENSORS"
7240 F$(3)="L.R. SENSORS"
7250 F$(4)="PHASER CNTRL"
7260 F$(5)="PHOTON TUBES"
7270 F$(6)="DAMAGE CNTRL"
7280 F$(7)="ENERGY SHLDS"
7290 RETURN
7300 REM MAINTAIN POSITION
7310 J=1
7320 W1=0
7330 W9=RND(1)
7340 C1=3
7350 T=T+.5
7360 IF C#<>"DOCKED" THEN 7380
7370 J=4
7380 FOR I=1 TO 7
7390 D(I)=D(I)+2*J
7400 IF D(I)<0 THEN 7420
7410 D(I)=0
7420 NEXT I
7430 E=E-50
7440 GOTO 9370
7450 REM RAMMING KLINGON
7460 PRINT
7470 Q(S1,S2)=0
7480 K3=K3-1

```

```

7490 K9=K9-1
7500 Z$="DOWN"
7510 FOR I=1 TO 9
7520 IF K(I,1)<>X3 THEN 7570
7530 IF K(I,2)<>Y3 THEN 7570
7540 Z6=K(I,1)
7550 Z7=K(I,2)
7560 GOTO 7580
7570 NEXT I
7580 K(I,3)=0
7590 GOSUB 10710
7600 Q(X3,Y3)=A5
7610 G(Q1,Q2)=K3*100+B3*10+S3
7620 PRINT"*** EMERGENCY ALERT ***"
7630 PRINT"*** COLLISION IMMINENT ***"
7640 PRINT
7650 GOSUB 10710
7660 PRINT"█"N$" RAMS KLINGON AT SECTOR"
7670 PRINTZ6"-Z7"."
7680 PRINT"*** KLINGON DESTROYED, "N$
7690 PRINT"HEAVILY DAMAGED ***"
7700 T1=VAL(TI$)
7710 IF T1+6>VAL(TI$) THEN 7710
7720 FOR I=1 TO 7
7730 D(I)=D(I)-(RND(1)*4+1)
7740 NEXT I
7750 C5=0
7760 E=E-500
7770 IF K9<=0 THEN 6610
7780 GOSUB 6020
7790 IF E<=0 THEN 6450
7800 S1=Z6
7810 S2=Z7
7820 A=2
7830 T1=VAL(TI$)
7840 IF T1+6>VAL(TI$) THEN 7840
7850 GOTO 5810
7860 REM TORPEDO SPREAD FORMATION
7870 PRINT"█SPREAD ANGLE (0.05-0.30) ?";
7880 INPUT#2,E5$:PRINT:E5=VAL(E5$)
7890 IF E5=0 THEN 2140
7900 IF E5<.05 THEN 7870
7910 IF E5>.30 THEN 7870
7920 E4=1
7930 PRINT"█ NUMBER"E4
7940 IF E4>1.5 THEN 7970
7950 C1=C1-E5
7960 GOTO 7980
7970 C1=C1+E5
7980 IF C1>=1.0 THEN 8010
7990 C1=C1+8.0

```

```

8000 GOTO 4630
8010 IF C1<9.0 THEN 4630
8020 C1=C1-8.0
8030 GOTO 4630
8040 REM PRINT POSITION
8050 PRINT"CURRENT POSITION:■"
8060 PRINT"QUADRANT"Q1"- "Q2TAB(20)"SECTOR"S1"- "S2
8070 PRINT"STARDATE : "T
8080 IF B3<1 THEN 2140
8090 PRINT"STARBASE AT SECTOR"B1"- "B2"."
8100 GOTO 2140
8110 REM STARCHART
8120 PRINT"STAR CHART:■"
8130 PRINT
8140 C0(Q1,Q2)=G(Q1,Q2)
8150 FOR J1=1 TO L8
8160 PRINTTAB(5*J1-1)"■";RIGHT$(STR$(J1),1);
8170 NEXT J1
8180 IF L8=6 THEN PRINTTAB(39)"■ ";
8190 FOR I=1 TO L8
8200 PRINT:PRINT"■";RIGHT$(STR$(I),1);"■";
8210 FOR J1=1 TO L8
8215 IF I=Q1 AND J1=Q2 THEN PRINT"■";
8220 PRINTTAB(5*J1-LEN(STR$(C0(I,J1)))));
8225 PRINTSTR$(C0(I,J1));"■";
8230 NEXT J1
8240 IF L8=6 THEN PRINTTAB(39);" ";
8250 NEXT I
8260 GOTO 8050
8270 REM DEATH RAY
8280 IF T>T0+7.99 THEN 8320
8290 PRINT"DEATH RAY CANNOT BE USED UNTIL "
8300 PRINT"STARDATE" T0+8
8310 GOTO 2140
8320 IF W3<.05 THEN 8350
8330 PRINT"DEATH RAY PERMANENTLY OUT OF ORDER."
8340 GOTO 2140
8350 IF C5=0 THEN 8380
8360 PRINT"YOU MUST LOWER YOUR SHIELDS TO FIRE."
8370 GOTO 2140
8380 W3=RND(1)
8390 PRINT
8400 IF W3>.35 THEN 8520
8410 PRINT" *ANTI-MATTER OVERLOAD, TEMPORARY"
8420 PRINT" PSEUDO-STARS CREATED FROM ENERGY."
8430 FOR I=1 TO 10
8440 FOR J=1 TO 10
8450 IF Q(I,J)<>0 THEN 8470
8460 Q(I,J)=4
8470 NEXT J
8480 NEXT I

```

```

8490 W3=RND(1)
8500 PRINT
8510 GOTO 1900
8520 IF W3>.20 THEN 8560
8530 PRINT"*** DEATH RAY BACKFIRES ***"
8540 PRINT"*** BETTER LUCK NEXT TIME ***"
8550 GOTO 5490
8560 IF W3>.80 THEN 8760
8570 PRINT"*** DEATH RAY ANNIHILATES ALL ";
8580 PRINT"KLINGONS IN QUADRANT"Q1"- "Q2"."
8590 K9=K9-K3
8600 IF K9<=0 THEN 6610
8610 FOR I=1 TO 10
8620 FOR J=1 TO 10
8630 IF Q(I,J)<>6 THEN 8650
8640 Q(I,J)=0
8650 IF Q(I,J)<>2 THEN 8670
8660 Q(I,J)=0
8670 NEXT J
8680 NEXT I
8690 K3=0
8700 G(Q1,Q2)=INT(B3*10+S3)
8710 Z3=Q1
8720 Z4=Q2
8730 GOSUB 11820
8740 W3=RND(1)
8750 GOTO 2140
8760 IF W3>.75 THEN 8810
8770 PRINT"***THERMAL REACTOR ENERGY DRAIN."
8780 PRINTINT(.95*E)"UNITS OF ENERGY BURNED."
8790 E=0.5*E
8800 GOTO 8570
8810 IF W3>.95 THEN 8960
8820 PRINT"***MATTER/ANTI-MATTER TRANSFORMATION."
8830 IF K3>0 THEN 8850
8840 GOTO 8420
8850 PRINT"***KLINGONS HIT BY MAIN BEAM ";
8855 PRINT"DUPLICATED."
8860 P8=K3
8870 PRINT"ENTIRE QUADRANT DISORDERED."
8880 FOR I=1 TO P8
8890 IF K3>8 THEN 8930
8900 K3=K3+1
8910 K9=K9+1
8920 NEXT I
8930 G(Q1,Q2)=K3*100+B3*10+S3
8940 W3=RND(1)
8950 GOTO 3620
8960 PRINT"*** DEATH RAY GOES CRAZY ***"
8970 PRINT"***DESTROYS EVERY KLINGON IN THE ";
8975 PRINT"GALAXY***";

```

```

8980 PRINT " *** ALSO DESTROYS THE "N$" ***"
8990 T1=VAL(TI$)
9000 IF T1+10>VAL(TI$) THEN 9000
9010 IF N1<>0 THEN 9030
9020 A7=5:N1=5
9030 GOTO 6610
9040 REM ABANDON SHIP
9050 IF N1=0 THEN 9090
9060 PRINT"THE STINKING GARBAGE SCOW CANNOT BE"
9070 PRINT"ABANDONED."
9080 GOTO 2140
9090 PRINT
9100 IF A7<>1 THEN 9140
9110 PRINT"YOU MAY NOT ABANDON SHIP AFTER YOU ";
9120 PRINT"HAVE BEEN ASKED TO SURRENDER."
9130 GOTO 2140
9140 PRINT"WOULD YOU LIKE TO RECONSIDER ?";
9150 INPUT#2,P$:PRINT
9160 IF P$="YES" THEN 2140
9170 IF P$="Y" THEN 2140
9180 PRINT"Y";
9190 PRINT"          3***** ABANDON SHIP *****"
9200 PRINT"          3***** ABANDON SHIP *****"
9210 PRINT"@SERIES ENTER-BLITZ INITIATED."
9220 PRINT"FAIL-SAFE AUTO-OVERRIDE INSERTED."
9230 PRINT"@COMMANDER AND SELECT FEMALE CREW ";
9240 PRINT"ESCAPE IN SHUTTLE CRAFT."
9250 PRINT"@***ENTERPRISE REDUCED TO SUB-ATOMIC"
9260 PRINT"          PARTICLES***"
9270 A7=5:N1=1:N9=1:T=T+1
9280 PRINT"@ABANDONING SHIP COSTS 1 STARDATE."
9290 T1=VAL(TI$)
9300 IF T1+12>VAL(TI$) THEN 9300
9310 IF T<T0+T9 THEN 9360
9320 PRINT"@    *** SO LONG, SPORT ***"
9330 N1=0
9340 PRINT
9350 GOTO 6440
9360 GOTO 2100
9370 REM SUPERNOVA
9380 IF B7<>0 THEN 9610
9390 IF V>=2 THEN 9420
9400 IF RND(1)>.15 THEN 9920
9410 GOTO 9430
9420 IF RND(1)>.10 THEN 9920
9430 B8=2+INT(2.5*RND(1))
9440 M1=0
9450 M2=0
9460 Z3=INT(RND(1)*L8+1)
9470 Z4=INT(RND(1)*L8+1)

```

```

9480 IF G(Z3,Z4)>999 THEN 9920
9490 S4=G(Z3,Z4)-INT(G(Z3,Z4)/9.999)*10
9500 IF S4<=0 THEN 9920
9510 REM STARBASE RESCUE
9520 GG=G(Z3,Z4)
9525 B4=INT(GG/9.999)-(INT(GG/99.99)*10)
9530 IF B4<>1 THEN 9680
9540 PRINT"*** EMERGENCY SUB-SPACE COMMUNICATION";
9550 PRINT" TO THE "N#" FROM STARBASE IN ";
9555 PRINT"QUADRANT"
9560 PRINTZ3"-Z4". **"
9570 PRINT"SUPERNOVA IMMINENT UNLESS STAR ";
9580 PRINT"BY PHOTON TORPEDO."
9590 B5=Z3
9600 B6=Z4
9610 B7=B7+1
9620 S4=G(B5,B6)-INT(G(B5,B6)/9.999)*10
9630 IF S4<=0 THEN 9910
9640 Z3=B5
9650 Z4=B6
9660 IF B7>B8 THEN 9680
9670 GOTO 9920
9680 PRINT"*****";
9690 PRINT"***** SUPERNOVA IN QUADRANT"Z3"-Z4
9700 PRINT"ALL THE MATTER IN THIS QUADRANT"
9710 PRINT"CONVERTED TO ENERGY."
9720 IF Q1<>Z3 OR Q2<>Z4 THEN 9780
9730 PRINT
9740 GOSUB 10710
9750 PRINT"UNFORTUNATELY, THE "N#" WAS IN"
9760 PRINT"THIS QUADRANT."
9770 GOTO 5540
9780 PRINT"FROM NOW ON, ANY VESSEL STOPPING IN ";
9790 PRINT"THISQUADRANT WILL BE TOTALLY ";
9795 PRINT"ENERGIZED."
9800 PRINT"*****";
9805 PRINT"*****"
9810 K4=INT(G(Z3,Z4)/99.99)
9820 B4=INT((G(Z3,Z4)/99.99-K4)*10.0)
9830 B9=B9-B4
9840 K9=K9-K4
9850 M1=Z3
9860 M2=Z4
9870 IF K9<=0 THEN 6610
9880 G(Z3,Z4)=1000
9890 C0(Z3,Z4)=G(Z3,Z4)
9900 GOSUB 11820
9910 B7=0
9920 REM MAGNETIC K-CAPTURE
9930 IF RND(1)>.10 THEN 10220
9940 FOR I=1 TO 9

```

```

9950 IF C(I,1) <> Q2 THEN 9970
9960 IF C(I,2) = Q2 THEN 2430
9970 NEXT I
9980 I = INT(RND(1)*4*R8+1)
9990 IF C(I,1) > .5 THEN 10020
10000 FOR I = 1 TO 9
10010 IF C(I,1) <= .5 THEN 10210
10020 IF C# <> "DOCKED" THEN 10060
10030 PRINT "MAGNETIC TRACTOR BEAM FROM QUADRANT"
10040 PRINT C(I,1) "-" C(I,2);
10045 PRINT "DEFLECTED BY STARBASE SHIELDS."
10050 GOTO 10220
10060 Q1 = C(I,1)
10070 Q2 = C(I,2)
10080 S1 = INT(RND(1)*8+1)
10090 S2 = INT(RND(1)*8+1)
10100 PRINT "M#N#" CAUGHT IN MAGNETIC TRACTOR"
10110 PRINT "BEAM."
10120 PRINT
10130 T = T + .5
10140 W9 = RND(1)
10150 FOR J = 1 TO 7
10160 D(J) = D(J) + 1
10170 IF D(J) <= 0 THEN 10190
10180 D(J) = 0
10190 NEXT J
10200 GOTO 1040
10210 NEXT I
10220 GOTO 2430
10230 REM STARSHIP ENCOUNTERS OLD SUPERNOVA
10240 PRINT "*** HEY STUPID.";
10245 PRINT "YOU WERE WARNED NOT TO"
10250 PRINT "STOP IN A QUADRANT ";
10255 PRINT "WHERE A SUPERNOVA HAD";
10260 PRINT "OCCURRED."
10270 GOTO 12310
10280 REM VARIABLE RESUPPLY
10290 IF N1 = 1 THEN 10340
10300 E = E0
10310 A9 = 3000
10320 P = 13
10330 GOTO 10370
10340 E = 2000
10350 A9 = 2000
10360 P = 8
10370 A7 = 0
10380 FOR I = 1 TO 7
10390 D(I) = D(I) + 2
10400 IF D(I) > 0 THEN D(I) = 0
10410 NEXT I
10420 RETURN

```

```

10430 REM ACTIVATE ENERGY SHIELDS
10440 IF D(7)>=0 THEN 10470
10450 PRINT"ENERGY SHIELDS DISABLED."
10460 GOTO 2140
10470 IF C5=0 THEN 10500
10480 PRINT"ENERGY SHIELDS ALREADY ACTIVATED."
10490 GOTO 2140
10500 IF A9>=200 THEN 10530
10510 PRINT"SHIELD ENERGY DRAINED."
10520 GOTO 2140
10530 PRINT"ENERGY SHIELDS ACTIVATED."
10540 C5=200
10550 Z$="UP"
10560 A9=A9-C5
10570 IF E<=0 THEN 6430
10580 GOSUB 6020
10590 GOTO 2140
10600 REM DEACTIVATE ENERGY SHIELDS
10610 PRINT
10620 IF C5>0 THEN 10650
10630 PRINT"ENERGY SHIELDS ARE ALREADY DOWN."
10640 GOTO 10700
10650 PRINT"ENERGY SHIELDS DEACTIVATED."
10660 C5=0
10670 Z$="DOWN"
10680 GOSUB 6020
10690 IF E<=0 THEN 6430
10700 GOTO 2140
10710 REM DETERMINE NAME
10720 IF N1<>1 THEN 10760
10730 A5=5
10740 N$="GARBAGE SCOW"
10750 GOTO 10780
10760 N$="ENTERPRISE"
10770 A5=1
10780 RETURN
10790 REM
10800 PRINT
10810 Z$="DOWN"
10820 A9=2000
10830 FORI=1TO8
10840 FORJ=1TO8
10850 K3=INT((G(I,J)/99.99)
10860 B3=INT(((G(I,J)/99.99-K3)*10.0)
10870 IF B3>0 THEN 10900
10880 NEXT J
10890 NEXT I
10900 PRINT" YOU HAVE BEEN TAKEN TO KLINGON"
10910 PRINT"HEADQUARTERS. IF YOU HAD A STARBASE,"
10920 PRINT"YOU WOULD BE REPATRIATED";
10925 PRINT" AND GIVEN A NEW";

```



```

10930 PRINT"STARSHIP TO COMMAND.";
10935 PRINT" SINCE YOU HAVE NO"
10940 PRINT"STARBASE, YOU WILL BE MERCILESSLY"
10950 PRINT"TORTURED TO DEATH."
10960 N7=1
10970 GOTO 11190
10980 N9=2
10990 T=T+1
11000 Q1=I
11010 Q2=J
11020 E=2000
11030 P=8
11040 N1=1
11050 PRINT"  YOU HAVE BEEN SENT ";
11055 PRINT" BACK TO STARFLEET"
11060 PRINT"COMMAND. ";
11065 PRINT"YOU ARE NOW IN COMMAND OF A NEW";
11070 PRINT"STARSHIP, ";
11075 PRINT"THE FEDERATION'S ONE AND ONLY"
11080 PRINT"SPACE JUNK COLLECTOR ----"
11090 PRINT"  THE GARBAGE SCOW,"
11100 PRINT"WHICH LOOKS LIKE /G/ ON THE SCAN."
11110 PRINT"  THE SMELLY GARBAGE";
11115 PRINT" SCOW IS CURRENTLY"
11120 PRINT"IN QUADRANT"Q1"-"Q2"."
11130 PRINT"TRY TO BE MORE CAREFUL THIS TIME."
11140 FOR I=1 TO 7
11150 D(I)=0
11160 NEXT I
11170 PRINT
11180 GOTO 3620
11190 RETURN
11200 Q$(0)="G." : Q$(1)="SE"
11210 Q$(2)="K" : Q$(3)="B"
11220 Q$(4)="F" : Q$(5)="G"
11230 Q$(6)="C"
11240 RETURN
11250 REM SELF DESTRUCT
11260 PRINT"  "
11270 PRINT"   SERIES ENTER-BLITZ INITIATED."
11280 PRINT"IN TEN SECONDS THIS VESSEL WILL SELF-"
11290 PRINT"           DESTRUCT."
11300 PRINT" 10"
11310 GOSUB 12660
11320 PRINT" 20"
11330 GOSUB 12660
11340 PRINT" 30"
11350 GOSUB 12660
11360 PRINT" 40"
11370 GOSUB 12660
11380 PRINT" 50"

```

```

11390 GOSUB 12660
11400 PRINT"                35"
11410 PRINT"                AT MINUS 5 SECONDS AND HOLDING."
11420 PRINT"FAIL-SAFE MECHANISM ENGAGED.ONLY THE"
11430 PRINT"COMMANDER OF THE STARSHIP CAN ";
11435 PRINT"OVERRIDE"
11440 PRINT"    WITH HIS SECRET PASSWORD. SCIENCE"
11450 PRINT"OFFICER SPOCK HAS ALREADY ENTERED HIS"
11460 PRINT"                CORRECT COUNTER-PASSWORD."
11470 PRINT"XTO CONTINUE SERIES ENTER-BLITZ,TYPE"
11480 PRINT"    YOUR SECRET PASSWORD :";
11490 INPUT#2,V$:PRINT
11500 IF V$<>"X" THEN 11600
11510 PRINT"                34"
11520 GOSUB 12660
11530 PRINT"                33"
11540 GOSUB 12660
11550 PRINT"                32"
11560 GOSUB 12660
11570 PRINT"                31"
11580 GOSUB 12660
11590 GOTO 5490
11600 PRINT
11610 PRINT"X    SERIES ENTER-BLITZ TERMINATED."
11620 PRINT
11630 GOTO 2100
11640 REM IMPULSE ENGINES
11650 PRINT"XIMPULSE ENGINES ON STANDBY."
11660 PRINT
11670 PRINT"XCOURSE (1-8,9999)  :";
11680 INPUT#2,C1$:PRINT:C1=VAL(C1$)
11690 IF C1<1 THEN 2140
11700 IF C1>=9 THEN 11670
11710 PRINT"XWARP FACTOR (1-5)  :";
11720 INPUT#2,W1$:PRINT:W1=VAL(W1$)
11730 IF W1<1 THEN 2140
11740 IF W1>5 THEN 2140
11750 PRINT"XIMPULSE ENGINES ENGAGED."
11760 C$="MOVING":POKE53280,12
11770 E=E-400
11780 IF E<=0 THEN 6440
11790 IF RND(1)<.85 THEN 9370
11800 PRINT"XIMPULSE ENGINES IMplode."
11810 GOTO 5540
11820 REM COMMANDER DIES
11830 FOR C2=1 TO 9
11840 IF C<(C2,1)=0 THEN 11890
11850 IF C<(C2,1)<>23 THEN 11890
11860 IF C<(C2,1)<>24 THEN 11890
11870 C<(C2,1)=0
11880 GOTO 11900

```

```

11890 NEXT C2
11900 RETURN
11910 REM TELEPORTER
11920 IF T>T0+4.99 THEN 11960
11930 PRINT"TELEPORTER CANNOT BE USED UNTIL"
11940 PRINT"STARDATE" T0+5
11950 GOTO 2140
11960 IF W9>.25 THEN 12000
11970 PRINT"COMMUNICATIONS WITH";
11975 PRINT" BASE TEMPORARILY"
11980 PRINT"LOST--TELEPORTER, SHE NO WORK."
11990 GOTO 2140
12000 H1=H1+1
12010 IF H1<=H2 THEN 12040
12020 PRINT"TELEPORTER CAPABILITY";
12025 PRINT" PERMANENTLY LOST."
12030 GOTO 2140
12040 PRINT"TELEPORTER ACTIVATED."
12050 FOR I=1 TO 8
12060 FOR J=1 TO 8
12070 K3=INT(G(I,J)/99.99)
12080 B3=INT((G(I,J)/99.99-K3)*10.0)
12090 IF B3>0 THEN 12130
12100 NEXT J
12110 NEXT I
12120 GOTO 11970
12130 Q1=I:Q2=J
12140 IF RND(1)>.15 THEN 12190
12150 PRINT"DISINTEGRATOR/INTEGRATOR";
12155 PRINT" MALFUNCTION--"
12160 PRINT"STARSHIP REORIENTATED UNUSABLE."
12170 PRINT"YOU ATE THAT ONE!"
12180 GOTO 5540
12190 GOTO 3620
12200 REM LEAVING GALAXY
12210 IF F>2 THEN 12290
12220 PRINT"STARSHIP ENCOUNTERS END OF GALAXY."
12230 PRINT"FOOLISH MORTAL,";
12235 PRINT" YOU HAVE ATTEMPTED TO"
12240 PRINT"VENTURE OUT OF ";
12245 PRINT" THE GALAXY INTO SUB-SPACE";
12250 PRINT" ON THE THIRD TIME";
12255 PRINT" YOU TRY THIS, THE "
12260 PRINT"LIVING GOD, KORP WILL DESTROY YOU."
12270 F=F+1
12280 GOTO 12330
12290 PRINT" YOU HAVE DARED TO ";
12295 PRINT"ENTER SUB-SPACE ";
12300 PRINT" A THIRD TIME."
12310 PRINT"FOR THIS,"
12315 PRINT" *****YOU DIE*****"

```

```

12320 GOTO 5490
12330 RETURN
12340 REM RANDOM NUMBER GENERATOR
12350 R1=INT(10*RND(1)+1)
12360 R2=INT(10*RND(1)+1)
12370 IF Q(R1,R2)<>0 THEN 12350
12380 RETURN
12390 PRINT"70,72ORDERS:           STARDATE=█"T
12400 PRINT"█ AS CAPTAIN OF THE";
12405 PRINT" UNITED STARSHIP"
12410 PRINT"ENTERPRISE, ";
12415 PRINT"YOUR MISSION IS TO DESTROY"
12420 PRINT"THE KLINGON INVASION ";
12425 PRINT"FORCE OF"K9"BATTLE"
12430 PRINT"CRUISERS. YOU HAVE"T9"SOLAR YEARS TO"
12440 PRINT"COMPLETE YOUR MISSION."
12450 PRINT"█(I.E. UNTIL STARDATE"TO+T9">)."
12460 IF B9>1 THEN 12490
12470 PRINT"█YOU WILL HAVE ONE";
12475 PRINT" SUPPORTING STARBASE."
12480 GOTO 12500
12490 PRINT"█YOU WILL HAVE"B9;
12495 PRINT"SUPPORTING STARBASES."
12500 PRINT"█CURRENT POSITION: QUADRANT"Q1"--"Q2
12510 PRINT"           SECTOR"S1"--"S2
12520 PRINT"█           █HIT ANY KEY TO CONTINUE█";
12530 WAIT 197,64,64:POKE198,0
12540 PRINT"█";
12550 GOSUB 12570
12560 GOTO 1040
12570 PRINT"█"
12580 RETURN
12590 PRINT"█ DO YOU WISH TO PLAY AGAIN ?";
12600 INPUT#2,A#:PRINT
12610 IF A#="YES" THEN RUN
12620 IF A#="Y" THEN RUN
12630 PRINT
12640 PRINT"█           █*** AVE ATQUE VALE ***"
12650 END
12660 TT=VAL(TI#)
12670 IF TT+1>VAL(TI#) THEN 12670
12680 RETURN
READY.

```

HI-RES GRAPHICS

THE PRINCIPLES OF HIGH RESOLUTION GRAPHICS

There are some books on the market that explain how to produce Hi-Res Graphics on the Commodore 64; a good example being the 64 Programmer's Reference Guide. The only setback to these books are that they are very sparse on detail and only explain how to implement Hi-Res in Basic.

In this part of this book, we will try to explain the production of Hi-Res routines and provide machine code routines to use the principles.

The Hi-Res screen on the 64 takes up 8000 bytes of memory and is bit mapped onto the screen by the command `POKE53265,PEEK(53265)OR32`. This selects bit mapped mode. If you did this in direct mode, you would see the first 8000 bytes of the 64's memory displayed on the screen. Some of the locations at the top of the screen will be constantly changing, this is because it is the zero page where different registers such as the key pressed location are. If you type anything with the screen set up like this, the middle area of the screen will change as the video screen is bit mapped on to the screen rather than directly displayed. To get out of this mode, type the command `POKE53265,PEEK(53265)AND223`.

It is obvious that this area cannot be used as the bit mapped screen as parts of it change without any prompt from the user. To select a Hi-Res area to plot onto, the command is `POKE53272,-PEEK(53272)OR8`. This will start your bit mapped area in the top half of the first 16K block of memory. As the video chip can only access 16K at a time, this is the best area to put a Hi-Res screen in the first bank. Unfortunately, using bank 0 for your bit mapped screen means that the Hi-Res screen will block off a lot of the user's Basic programming memory and thus leave the user only 6K of memory to use for Basic programs and variables.

To rectify this problem, the best way is either to select a bank that is not in the Basic programming area at all (bank 3) or to select a bank at the top of the Basic programming memory and lower the pointers so that variables do not corrupt this area of memory.

In the routines that follow, the second bank mentioned in the last paragraph (bank 2) is used and the bit mapped screen is stored in the RAM behind the Basic ROM.

Using bank 2, there is still 30K of memory to program in and there is still room to put sprites and user defined characters in there as well.

THE ROUTINES

There are eight callable routines in all included in the Hi-Res routines. These are as follows:

1. Set up Hi-Res screen and enter Hi-Res mode.
2. Plot a point on the screen.
3. Draw a line on the screen.
4. Display a character on the screen.
5. Change from normal screen to Hi-Res screen.
6. Clear the Hi-Res screen.
7. Change from Hi-Res screen to normal screen.
8. Set colour memory to zeros.

1. SET UP HI-RES SCREEN

This routine is called by:
SYS(49152),,mode,colour

Mode: 0 for standard Hi-Res,
1 for multi colour mode.

Colour: 0 to 15 for both border and background colour.

The routine takes these parameters from the input line and uses them to set a flag for the mode and colour settings. The operations explained earlier are used to enter Hi-Res mode (and Multi-colour if it is so wished). Bank 2 is then selected and a video screen is chosen for plotting colours. Then the 8K Hi-Res screen is cleared and return to Basic.

2. PLOT A POINT

This routine is called by:
SYS(49347),X,Y,colour,brush

X: x co-ordinate,

Y: y co-ordinate,

colour: 0 to 15 for plotting colour,

brush: 0 or 1 in standard Hi-Res for unplot or plot,

0,1,2, or 3 in Multi-colour for unplot, or one of three brushes.

In this routine, the location on the screen is calculated for the point and the bit on that byte. The equivalent in Basic is:

```

10 BYTE=INT(X/8)*8+INT(Y/8)*320+(YAND7)+40960
20 IF MODE=0THENBIT=7-(XAND7):GOTO40
30 BIT=INT((7-(XAND7))/2)*2

```

This routine also checks for multi-colour or standard Hi-Res mode.

Next the machine code routine decides which brush the point is plotted using, and plots the point as:

```
40 POKEBYTE,PEEK(BYTE)OR 2(to the power of)BIT
```

Finally the colour is sent to either the video RAM or the colour nybble RAM depending on the brush required.

The bit combinations to correspond with the colours are as follows:

In standard Hi-Res mode, if a bit is set to a '1' (as 01), it is plotted and if it is set to a '0' (as 00) it is off.

In Multi-colour mode, two bits are used for plotting a point:

```

Bits Colour From
00 Point is off,
01 Upper 4 bits of video RAM,
10 Lower 4 bits of video RAM,
11 Colour Nybble RAM.

```

3. DRAW A LINE

The method for calculating and drawing a line was taken from the quarter square line drawing routine in Pet Graphics by Nick Hampshire.

The equivalent routine for Hi-Res on the 64 is given below and the SYS call is to the machine code plot point routine.

```

1000 XD=X2-X1
1010 YD=Y2-Y1
1020 REM **NEAREST DIAGONAL**
1030 A0=1:A1=
1040 IF YD<0 THEN A0=-1
1050 IF XD<0 THEN A1=-1
1060 REM **NEAREST HORIZ/VERT**
1070 XE=ABS(XD):YE=ABS(YD):D1=XE-YE
1080 IF D1>=0 THEN 1120
1090 SO=-1:S1=0:LG=YE:SH=XE
1100 IF YD>=0 THEN SO=1
1110 GOTO 1140
1120 S0=0:S1=-1:LG=XE:SH=YE

```



```

1130 IF XD>=0 THEN S1=1
1140 REM **SET UP**
1150 TT=LG:TS=SH:UD=LG-SH:CT=SH-LG/2
1160 D=0
1170 REM **WHILE MORE POINTS DO**
1180 SYS(49347),X1,Y1,colour,brush
1190 IF CT>=0 THEN 1220
1200 CT=CT+TS:X1=X1+S1:Y1=Y1+S0
1210 GOTO 1230
1220 CT=CT-UD:X1=X1+A1:Y1=Y1+A0
1230 TT=TT-1
1240 IF TT<=0 THEN RETURN
1250 GOTO 1180

```

The routine would be called by GOSUB 1000 with the start and end points of the line set in X1,Y1 and X2,Y2. The test on the coordinates is made within the point plot routine and if it is out of bounds, the point is just not plotted.

4. DISPLAY A CHARACTER

This routine uses the Character Generator to get the points to be plotted for the character.

The routine is called by:

```
SYS(50487),X,Y,colour,brush,rvorn,ASC(char)
```

X: The top left x co-ordinate of the character
Y: The top left y co-ordinate of the character
colour: The colour that the character is to be plotted in
brush: The brush that is to be used
rvorn: Whether you require the character to be plotted in reverse or normal
0 – Plot in normal
1 – Plot in reverse
char: The actual character in quotes

The character plot routine first takes the ASCII value of the character and converts it to the CBM poke value for that character. If the character is out of bounds, it is just not plotted. If the character is OK, each of the eight bytes of the character is loaded in and then each bit is tested to see if it is on. If the bit is on, the point is plotted but if the bit is off the next point is tested. If you requested the character to be plotted in reverse, then each byte of the character is exclusive or'd with 255 to invert it before testing the byte.

5. NORMAL TO HI-RES SCREEN

Called by: SYS(49182) this routine allows the user to switch from the normal (text) screen to the Hi-Res screen without clearing the Hi-Res screen. This routine is the part of the first routine that sets up the Hi-Res screen.

6. CLEAR THE HI-RES SCREEN

This routine is one of the subroutines called by the first routine. It is called by SYS(49231) and just loops through the Hi-Res screen setting each of the 8000 bytes to zero. This is a very slow process in Basic, but in machine code it is done in a split second.

The equivalent Basic routine would be:

```
FORI=40960TO48959:POKEI,0:NEXT
```

7. HI-RES TO NORMAL SCREEN

When you wish to return to the normal (text) screen from the Hi-Res screen, use SYS(49821) and not STOP/RESTORE as the latter would not work. If the latter is used, POKE648,4 will restore the correct screen but it would have to be typed in blind.

8. SET COLOUR MEMORY TO ZEROS

This routine was, like routines 5 and 6, a product of the first routine and just sets each location in the colour memory to black. This is very useful if Multi-Colour mode is used as the colour memory is used for plotting colours and will affect the colour of text displayed.

NOTE

Because Bank 2 was used for the routines and the Hi-Res screen is behind the Basic ROM, to plot a point, the byte to be plotted in must be read in first. This cannot be done from Basic because if you peek a location of the Basic ROM area, the value from the Basic ROM is returned, but POKEing to the same location would store it in the RAM behind the ROM. Therefore to plot a point, the Basic ROM must be switched out while plotting.

All of the following demonstrations require that the machine code routines following are 'resident' in memory. The routines are given in the form of an Assembler listing so that those users interested in the routines can study them. The routines are also provided, for those without an assembler, in the form of a Basic loader which can be entered and then RUN.

LINE#	LOC	CODE	LINE
00001	0000		; COMBINED MULTI-COLOUR AND STANDARD
00002	0000		;HIRES GRAPHICS PACKAGE, CONSISTING OF
00003	0000		;EIGHT SYS CALLS. THE CALLS ARE AS FOLLOWS:
00004	0000		;
00005	0000		;SYS 49152,MODE,COLOUR
00006	0000		; SETS UP THE HIRES SCREEN IN THE REQUIRED
00007	0000		; MODE:- 0=STANDARD,1=MULTI-COLOUR.
00008	0000		; COLOUR IS FOR BOTH SCREEN AND BORDER.
00009	0000		;
00010	0000		;SYS 49182
00011	0000		; EQUIVALENT TO THE GRAPH FUNCTION
00012	0000		; IN MOST HIRES PACKAGES. IT SWITCHES
00013	0000		; FROM THE NORMAL SCREEN TO THE HIRES
00014	0000		; SCREEN WITHOUT CLEARING THE SCREEN.
00015	0000		;
00016	0000		;SYS 49231
00017	0000		; CLEARS THE 8K GRAPHICS SCREEN.
00018	0000		;
00019	0000		;SYS 49308
00020	0000		; CLEARS THE COLOUR MEMORY TO ZERO'S.
00021	0000		;
00022	0000		;SYS 49347,X,Y,COLOUR,BRUSH
00023	0000		; PLOTS A POINT AT X,Y WITH COLOUR
00024	0000		; (0-15) AND BRUSH:-
00025	0000		; (0-3) IN MULTI OR
00026	0000		; 0 OR 1 IN STANDARD.
00027	0000		; IN BOTH CASES BRUSH 0 IS UNPLOT.

```

00028 0000 ;
00029 0000 ;SYS 49821
00030 0000 ; SWITCH FROM HIRES SCREEN TO NORMAL
00031 0000 ; SCREEN (EQUIVALENT TO NORM).
00032 0000 ;
00033 0000 ;SYS 49850,X1,Y1,X2,Y2,COLOUR,BRUSH
00034 0000 ; DRAWS A LINE BETWEEN X1,Y1 AND X2,Y2
00035 0000 ; WITH COLOUR AND BRUSH AS ABOVE.
00036 0000 ;
00037 0000 ;SYS 50487,X,Y,COLOUR,BRUSH,RVORN,ASC(CHAR)
00038 0000 ; DISPLAY A CHARACTER AT COORDINATES
00039 0000 ; X,Y WITH COLOUR AND BRUSH AS ABOVE.
00040 0000 ; RVORN IS:-
00041 0000 ; 0 FOR NORMAL CHARACTER,
00042 0000 ; 1 FOR REVERSE CHARACTER.
00043 0000 ; NOTE THAT CBM GRAPHICS CHARACTERS
00044 0000 ; ARE NOT AVAILABLE WITH THIS ROUTINE.
00045 0000 ;
00046 0000 ;
00047 0000 ;
00048 0000 ;DECLARE ALL VARIABLES.
00049 0000 ;
00050 0000 ;
00051 0000 ; PNTR =#02 ; POINTER TO CHARACTER ROM
00052 0000 ; T1 =#57 ; POINTER TO HIRES SCREEN
00053 0000 ; T2 =#59 ; X COORDINATE
00054 0000 ; T3 =#58 ; Y COORDINATE
00055 0000 ;

```

LINE#	LOC	CODE	LINE
00056	0000		=#50
00057	0000		=#5E
00058	0000		=#5F
00059	0000	MODE	=#FB
00060	0000	PBR	=#FC
00061	0000	COL	=#FD
00062	0000		* =#033C
00063	033C	00 00	.WDR 0
00064	033E	00 00	.WDR 0
00065	0340	00 00	.WDR 0
00066	0342	00 00	.WDR 0
00067	0344	00 00	.WDR 0
00068	0346	00 00	.WDR 0
00069	0348	00 00	.WDR 0
00070	034A	00 00	.WDR 0
00071	034C	00 00	.WDR 0
00072	034E	00 00	.WDR 0
00073	0350	00 00	.WDR 0
00074	0352	00 00	.WDR 0
00075	0354	00 00	.WDR 0
00076	0356	00 00	.WDR 0
00077	0358	00 00	.WDR 0
00078	035A	00 00	.WDR 0
00079	035C	00 00	.WDR 0
00080	035E	00 00	.WDR 0
00081	0360	00 00	.WDR 0

```

; Y AND Z
; 2*(7-(X AND 7))
; USED AS SECOND BIT IN MULTI
; FLAG FOR HIRES OR MULTI MODE
; 'PAINTERUSH'
; POINT COLOUR
;
; COORDINATES FOR LINE
; PLOT ROUTINE
; AS X1,Y1
; AND X2,Y2
; DIFFERENCE OF X1 AND X2
; DIFFERENCE OF Y1 AND Y2
; ABS(XD)
; ABS(YD)
; THE NEXT 11 ARE
; VARIABLES USED IN THE
; LINE PLOT ROUTINE

```

```

00082 0362 00 00 .WOR 0 ; TOP LEFT X OF CHARACTER
00083 0364 00 00 .WOR 0 ; TOP LEFT Y OF CHARACTER
00084 0366 00 .BYT 0 ; CHARACTER
00085 0367 00 .BYT 0 ; REVERSE OR NORMAL
00086 0368 00 00 .WOR 0 ; TEMP X LOCATION
00087 036A 00 CNTR1 .BYT 0 ; COUNTER FOR CHAR PLOT
00088 036B 00 POINT .BYT 0 ; BIT OF CHAR BEING PLOTTED
00089 036C CHKCOM =#A0F0 ; SCAN FIRST COMMA
00090 036C PARAMS =#B7EB ; PICK OF 2 PARAMETERS AFTER SYS
00091 036C BORDER =#D020 ; BORDER COLOUR
00092 036C SCREEN =#D021 ; SCREEN COLOUR

* =#C000
;
; ROUTINE TO SET UP HIRES SCREEN
;
00094 036C JSR CHKCOM ; READ OF PARAMETERS
00095 C000 JSR PARAMS ; FOR MODE AND COLOUR
00096 C000 TXA ; X STORES COLOUR
00097 C000 STA BORDER
00098 C000 STA SCREEN
00099 C003 LDA #14 ; MODE IN LOC #14
00100 C006 STA MODE
00101 C007 BEQ MULTI
00102 C00A LDX #00
00103 C00D JSR CLRMEM ; CLEAR HIRES SCREEN
00104 C00F JSR CLRSCN ; CLEAR VIDEO SCREEN
00105 C011 JSR CLRCLC ; CLEAR COLOUR MEMORY
00106 C013 MULTI
00107 C015 20 4F C0
00108 C018 20 76 C0
00109 C01B 20 9C C0

```

LINE#	LOC	CODE	LINE
00110	C01E	A9 38	LDA #38
00111	C020	8D 11 00	STA \$D011
00112	C023	A9 10	LDA #310
00113	C025	8D 18 00	STA \$D018
00114	C028	A5 FB	LDA MODE
00115	C02A	F0 05	BEQ DONE
00116	C02C	A9 08	LDA #308
00117	C02E	8D 16 00	STA \$D016
00118	C031	A9 80	LDA #380
00119	C033	85 38	STA #38
00120	C035	85 34	STA #34
00121	C037	AD 02 00	LDA \$D002
00122	C03A	09 03	ORA #303
00123	C03C	8D 02 00	STA \$D002
00124	C03F	AD 00 00	LDA \$D000
00125	C042	29 FC	AND #3FC
00126	C044	09 01	ORA #301
00127	C046	8D 00 00	STA \$D000
00128	C049	A9 84	LDA #384
00129	C04B	8D 88 02	STA \$D288
00130	C04E	60	RTS
00131	C04F	A0 00	CLRMEM
00132	C051	A9 40	LDA #340
00133	C053	85 57	STA T1
00134	C055	A9 BF	LDA #3BF
00135	C057	85 58	STA T1+1

; SELECT BIT MAP MODE
; CHOOSE HIRES SCREEN
; IF MODE=0
; ELSE SET MULTI MODE
; PROTECT HIRES SCREEN MEMORY
; TOP OF MEMORY
; BASIC VARIABLES
; SELECT BANK 2 FOR HIRES
; SCREEN
; POINTER TO VIDEO SCREEN
; LOOP TO CLEAR HIRES
; SCREEN
; THIS ROUTINE CALLABLE
; SEPARATELY AS ABOVE

```

00136 C059 A9 00 LOOP LDA #80
00137 C05B 91 57 STA (T1),Y
00138 C05D A5 57 LDA T1
00139 C05F F0 05 BEQ L0001
00140 C061 C6 57 DEC T1
00141 C063 4C 59 C0 JMP LOOP
00142 C066 C6 58 L0001 DEC T1+1
00143 C068 A5 58 LDA T1+1
00144 C06A C9 9F CMP #9F
00145 C06C F0 07 BEQ OUT1
00146 C06E A9 FF LDA #FF
00147 C070 85 57 STA T1
00148 C072 4C 59 C0 JMP LOOP
00149 C075 60 OUT1 RTS
00150 C076 A0 00 CLRSCN LDY #80
00151 C078 A9 E7 LDA #E7
00152 C07A 85 57 STA T1
00153 C07C A9 87 LDA #87
00154 C07E 85 58 STA T1+1
00155 C080 8A LOOP1 TXA
00156 C081 91 57 STA (T1),Y
00157 C083 A5 57 LDA T1
00158 C085 F0 05 BEQ L0002
00159 C087 C6 57 DEC T1
00160 C089 4C 80 C0 JMP LOOP1
00161 C08C C6 58 L0002 DEC T1+1
00162 C08E A5 58 LDA T1+1
00163 C090 C9 93 CMP #83
00164 C092 F0 07 BEQ OUT2
; LOOP TO CLEAR
; VIDEO SCREEN

```


LINE#	LOC	CODE	LINE
00165	C094	A9 FF	LDA #FF
00166	C096	85 57	STA T1
00167	C098	4C 80 C0	JMP LOOP1
00168	C09B	60	RTS
00169	C09C	A0 00	LDY #00
00170	C09E	A9 E7	LDA #E7
00171	C0A0	85 57	STA T1
00172	C0A2	A9 DB	LDA #DB
00173	C0A4	85 58	STA T1+1
00174	C0A6	A9 00	LDA #00
00175	C0A8	91 57	STA <T1>,Y
00176	C0AA	A5 57	LDA T1
00177	C0AC	F0 05	BEQ L0003
00178	C0AE	C6 57	DEC T1
00179	C0B0	4C A6 C0	JMP LOOP2
00180	C0B3	C6 58	DEC T1+1
00181	C0B5	A5 58	LDA T1+1
00182	C0B7	C9 07	CMP #07
00183	C0B9	F0 07	BEQ OUT3
00184	C0BB	A9 FF	LDA #FF
00185	C0BD	85 57	STA T1
00186	C0BF	4C A6 C0	JMP LOOP2
00187	C0C2	60	RTS
00189	C0C3		;
00190	C0C3		; ROUTINE TO PLOT A POINT

; LOOP TO CLEAR
; COLOUR MEMORY.
;
; THIS ROUTINE CALLABLE
; SEPARATELY AS MENTIONED
; ABOVE.

```

00191 C0C3      20 FD AE      JSR CHKCOM      ; GET X AND Y
00192 C0C3      20 EB B7     JSR PARAMS     ; COORDINATES
00193 C0C6      A5 14      LDA #14        ; X IS STORED AS A DOUBLE
00194 C0C9      85 59      STA T2        ; BYTE VALUE
00195 C0C8      A5 15      LDA #15
00196 C0CD      85 5A      STA T2+1
00197 C0CF      8A        TXA
00198 C0D1      8A        STA T3
00199 C0D2      85 58      JSR CHKCOM     ; Y IS STORED AS A SINGLE
00200 C0D4      20 FD AE     JSR PARAMS     ; BYTE VALUE
00201 C0D7      20 EB B7     JSR PARAMS     ; GET COLOUR AND BRUSH
00202 C0DA      8A        TXA
00203 C0DB      85 FC      STA PBR
00204 C0DD      A5 14      LDA #14
00205 C0DF      85 FD      STA COL
00206 C0E1      A5 5A      LDA T2+1
00207 C0E3      C9 00      CMP #00
00208 C0E5      F0 0B      BEQ XOK
00209 C0E7      C9 01      CMP #01
00210 C0E9      D0 06      BNE XER
00211 C0EB      A5 59      LDA T2
00212 C0ED      C9 40      CMP #40
00213 C0EF      90 01      BCC XOK
00214 C0F1      60        RTS
00215 C0F2      A5 58      LDA T3
00216 C0F4      C9 C8      CMP #C8
00217 C0F6      90 01      BCC YOK
00218 C0F8      60        RTS
00219 C0F9      A5 59      LDA T2

```

;

PLOT

XER

XOK

YOK

LINE#	LOC	CODE	LINE
00220	C0FB	29 07	AND #07
00221	C0FD	85 5E	STA T5
00222	C0FF	A9 07	LDA #07
00223	C101	38	SEC T5
00224	C102	E5 5E	SBC T5
00225	C104	85 5E	STA T5
00226	C106	A5 FB	LDA MODE
00227	C108	F0 08	BEQ BITOK
00228	C10A	46 5E	LSR T5
00229	C10C	06 5E	ASL T5
00230	C10E	A5 5E	LDA T5
00231	C110	18	CLC
00232	C111	69 01	ADC #01
00233	C113	85 5F	STA T6
00234	C115	A5 5E	LDA T5
00235	C117	F0 09	BEQ BITST1
00236	C119	A8	TAY
00237	C11A	A9 01	LDA #01
00238	C11C	0A	ASL A
00239	C11D	88	DEY
00240	C11E	D0 FC	BNE LOOP10
00241	C120	F0 02	BEQ BIT1
00242	C122	A9 01	LDA #01
00243	C124	85 5E	STA T5
00244	C126	A5 5F	LDA T6
00245	C128	F0 09	BEQ BITST2

; BE PLOTTED AS
; 7-(X AND 7)

; IF NOT MULTI, VALUE
; IS O.K.
; IN MULTI CALCULATE
; INT((7-(X AND 7))/2)*2

; AND THE SECOND BIT IS
; ONE GREATER
; CALCULATE 2¹ OF THE
; VALUE(S) PREVIOUSLY
; CALCULATED

00246	C12A	A8	TAY	
00247	C12B	A9 01	LDA #01	
00248	C12D	0A	LOOP11 ASL A	
00249	C12E	88	DEY	
00250	C12F	D0 FC	BNE LOOP11	
00251	C131	F0 02	BEQ BIT2	
00252	C133	A9 01	BITST2 LDA #01	
00253	C135	85 5F	BIT2 STA T6	
00254	C137	A9 00	LDA #00	
00255	C139	85 5C	STA T3+1	
00256	C13B	85 58	STA T1+1	
00257	C13D	85 57	STA T1	
00258	C13F	A5 58	LDA T3	; CALCULATE Y AND 7
00259	C141	29 07	AND #07	
00260	C143	85 5D	STA T4	; STORE IN T4
00261	C145	A5 58	LDA T3	
00262	C147	4A	LSR A	
00263	C148	4A	LSR A	
00264	C149	4A	LSR A	
00265	C14A	85 58	STA T3	
00266	C14C	A0 05	LDY #05	
00267	C14E	18	LOOP20 CLC	
00268	C14F	0A	ASL A	
00269	C150	26 58	ROL T1+1	
00270	C152	88	DEY	
00271	C153	D0 F9	BNE LOOP20	
00272	C155	85 57	STA T1	
00273	C157	A5 58	LDA T3	
00274	C159	A0 03	LDY #03	; INT(Y/8)*32

LINE#	LOC	CODE	LINE
00275	C15B	18	LOOP21 CLC
00276	C15C	0A	ASL A
00277	C15D	88	DEY
00278	C15E	00 FB	BNE LOOP21
00279	C160	85 58	STA T3
00280	C162	18	CLC
00281	C163	65 57	ADC T1
00282	C165	85 58	STA T3
00283	C167	A5 58	LDA T1+1
00284	C169	69 00	ADC #00
00285	C16B	85 5C	STA T3+1
00286	C16D	A0 03	LDY #03
00287	C16F	18	MULT80 CLC
00288	C170	46 5A	LSR T2+1
00289	C172	66 59	ROR T2
00290	C174	88	DEY
00291	C175	D0 F8	BNE MULT80
00292	C177	A5 59	LDA T2
00293	C179	85 57	STA T1
00294	C17B	A5 5A	LDA T2+1
00295	C17D	85 58	STA T1+1
00296	C17F	A0 03	LDY #03
00297	C181	18	MULT81 CLC
00298	C182	06 57	ASL T1
00299	C184	26 58	ROL T1+1
00300	C186	88	DEY

; INT(Y/8)*8

; INT(Y/8)*40

; INT(X/8)
; STORE IN T1

```

00301 C187 D0 F8 BNE MULT81
00302 C189 A0 08 LDY #08
00303 C18B 18 ADD1
00304 C18C A5 58 LDR T3
00305 C18E 65 57 ADC T1
00306 C190 85 57 STA T1
00307 C192 A5 5C LDR T3+1
00308 C194 65 58 ADC T1+1
00309 C196 85 58 STA T1+1
00310 C198 88 DEY
00311 C199 D0 F0 BNE ADD1
00312 C19B 18 CLC
00313 C19C A5 5D LDR T4
00314 C19E 65 57 ADC T1
00315 C1A0 85 57 STA T1
00316 C1A2 A9 00 LDR #00
00317 C1A4 65 58 ADC T1+1
00318 C1A6 18 CLC
00319 C1A7 A9 A0 LDR #A0
00320 C1A9 65 58 ADC T1+1
00321 C1AB 85 58 STA T1+1
00322 C1AD 18 CLC
00323 C1AE A5 59 LDR T2
00324 C1B0 65 58 ADC T3
00325 C1B2 85 58 STA T3
00326 C1B4 A9 00 LDR #00
00327 C1B6 65 5C ADC T3+1
00328 C1B8 85 5C STA T3+1
00329 C1BA A9 36 LDR #36

```

```

; INT(X/8)*8
; ADD 8 TIMES T3 <INT(Y/8)*40>
; INTO T1

; ADD T4 <Y AND 7> INTO T1

; ADD THE START OF GRAPHICS
; SCREEN INTO T1

; STORE IN T3,
; INT(X/8)+INT(Y/8)*40

; DISABLE BASIC ROM FOR

```

LINE#	LOC	CODE	LINE
00330	C18C	85 01	STA #01
00331	C18E	A5 FB	LDA MODE
00332	C1C0	00 03	BNE MULTII
00333	C1C2	4C 63 C2	JMP HIRES
00334	C1C5	A5 FC	LDA PBR
00335	C1C7	C9 00	CMP #00
00336	C1C9	F0 11	BEQ BRUSH0
00337	C1CB	C9 01	CMP #01
00338	C1CD	F0 28	BEQ BRUSH1
00339	C1CF	C9 02	CMP #02
00340	C1D1	F0 51	BEQ BRUSH2
00341	C1D3	C9 03	CMP #03
00342	C1D5	F0 72	BEQ BRUSH3
00343	C1D7	A9 37	LDA #37
00344	C1D9	85 01	STA #01
00345	C1DB	60	RTS
00346	C1DC	A0 00	BRUSH0 LDY #00
00347	C1DE	A5 5E	LDA T5
00348	C1E0	49 FF	EOR #FF
00349	C1E2	85 5E	STA T5
00350	C1E4	A5 5F	LDA T6
00351	C1E6	49 FF	EOR #FF
00352	C1E8	85 5F	STA T6
00353	C1EA	B1 57	LDA (T1),Y
00354	C1EC	25 5E	AND T5
00355	C1EE	25 5F	AND T6

; PLOTTING ROUTINE
; CHOOSE BRUSH 0
; CHOOSE BRUSH 1
; CHOOSE BRUSH 2
; CHOOSE BRUSH 3
; ERROR IN BRUSH NUMBER
; RESTORE BASIC ROM AND
; RETURN TO BASIC PROG
; UNPLOT BOTH POINTS IN
; MULTI-COLOUR MODE
; STORE ON HIRES SCREEN

```

00356 C1F0 91 57 STA (T1),Y
00357 C1F2 A9 37 LDA #37
00358 C1F4 85 01 STA #01
00359 C1F6 60 RTS
00360 C1F7 A0 00 BRUSH1 LDY #00
00361 C1F9 A5 5F LDA T6
00362 C1FB 49 FF EOR #FF
00363 C1FD 85 5F STA T6
00364 C1FF B1 57 LDA (T1),Y
00365 C201 05 5E ORA T5
00366 C203 25 5F AND T6
00367 C205 91 57 STA (T1),Y
00368 C207 A9 84 LDA #84
00369 C209 18 CLC
00370 C20A 65 5C ADC T3+1
00371 C20C 85 5C STA T3+1
00372 C20E 06 FD ASL COL
00373 C210 06 FD ASL COL
00374 C212 06 FD ASL COL
00375 C214 06 FD ASL COL
00376 C216 B1 58 LDA (T3),Y
00377 C218 29 0F AND #0F
00378 C21A 18 CLC
00379 C21B 65 FD ADC COL
00380 C21D 91 58 STA (T3),Y
00381 C21F A9 37 LDA #37
00382 C221 85 01 STA #01
00383 C223 60 RTS
00384 C224 A0 00 BRUSH2 LDY #00
; ENABLE BASIC ROM
; STORE COMBINATION 01 IN
; THE BYTES FOR MULTI
; POINT.
; STORE ON HIRES SCREEN
; ADD START OF VIDEO RAM
; TO T3
; COLOUR TIMES 16
; MASK OFF TOP 4 BITS
; ADD COLOUR TIMES 16
; STORE IN VIDEO RAM
; ENABLE BASIC ROM

```


LINE#	LOC	CODE	LINE
00385	C226	A5 5E	
00386	C228	49 FF	
00387	C22A	85 5E	
00388	C22C	B1 57	
00389	C22E	25 5E	
00390	C230	05 5F	
00391	C232	91 57	
00392	C234	18	
00393	C235	A9 84	
00394	C237	65 5C	
00395	C239	85 5C	
00396	C23B	B1 58	
00397	C23D	29 F0	
00398	C23F	18	
00399	C240	65 FD	
00400	C242	91 58	
00401	C244	A9 37	
00402	C246	85 01	
00403	C248	60	
00404	C249	A0 00	
00405	C24B	B1 57	
00406	C24D	05 5E	
00407	C24F	05 5F	
00408	C251	91 57	
00409	C253	A9 08	
00410	C255	18	


```

LDA T5
EOR #FF
STA T5
LDA (T1),Y
AND T5
ORA T6
STA (T1),Y
CLC
LDA #84
ADC T3+1
STA T3+1
LDA (T3),Y
AND #F0
CLC
ADC COL
STA (T3),Y
LDA #37
STA #01
RTS
BRUSH3 LDY #80
LDA (T1),Y
ORA T5
ORA T6
STA (T1),Y
LDA #08
CLC

```

; PLOT POINTS FOR BRUSH 2
; STORE ON HIRES SCREEN
; ADD START OF VIDEO RAM
; TO T3
; MASK OFF BOTTOM 4 BITS
; ADD IN THE COLOUR
; STORE ON VIDEO RAM
; ENABLE BASIC ROM
; PLOT POINTS FOR BRUSH 3
; STORE ON HIRES SCREEN
; ADD START OF COLOUR RAM
; TO T3

```

00411 C256 65 5C ADC T3+1
00412 C258 85 5C STA T3+1
00413 C25A A5 FD LDR COL
00414 C25C 91 58 STA (T3),Y
00415 C25E A9 37 LDR #£37
00416 C260 85 01 STA #01
00417 C262 60 RTS
00418 C263 A0 00 LDY #£00
00419 C265 A5 FC LDR PBR
00420 C267 F0 23 BEQ UNPLOT
00421 C269 B1 57 LDR (T1),Y
00422 C26B 05 5E ORA T5
00423 C26D 91 57 STA (T1),Y
00424 C26F A9 84 LDR #£84
00425 C271 18 CLC
00426 C272 65 5C ADC T3+1
00427 C274 85 5C STA T3+1
00428 C276 A5 FD LDR COL
00429 C278 0A ASL A
00430 C279 0A ASL A
00431 C27A 0A ASL A
00432 C27B 0A ASL A
00433 C27C 85 5F STA T6
00434 C27E B1 58 LDR (T3),Y
00435 C280 29 0F AND #£0F
00436 C282 18 CLC
00437 C283 65 5F ADC T6
00438 C285 91 58 STA (T3),Y
00439 C287 A9 37 LDR #£37

HIRES
; STORE COLOUR IN COLOUR RAM
; ENABLE BASIC ROM

; IF BRUSH=0 THEN UNPLOT
; IN STANDARD MODE
; OTHERWISE PLOT POINT

; ADD START OF VIDEO RAM
; TO T3

; COLOUR TIMES 16
; MASK OFF TOP 4 BITS

; ADD COLOUR
; STORE IN VIDEO RAM
; ENABLE BASIC ROM

```

LINE#	LOC	CODE	LINE
00440	C289	85 01	STA #01
00441	C288	60	RTS
00442	C28C	A5 5E	UNPLOT LDA T5 ; UNPLOT IN STANDARD MODE
00443	C28E	49 FF	EOR #FF
00444	C290	85 5E	STA T5
00445	C292	B1 57	LDA (T1),Y
00446	C294	25 5E	AND T5
00447	C296	91 57	STA (T1),Y
00448	C298	A9 37	LDA #37
00449	C29A	85 01	STA #01 ; ENABLE BASIC ROM
00450	C29C	60	RTS
00451	C29D		
00452	C29D		; ROUTINE TO RETURN TO NORMAL SCREEN
00453	C29D		;
00454	C29D	A9 04	LDA #04
00455	C29F	80 88 02	STA \$0288 ; NORMAL VIDEO SCREEN
00456	C2A2	AD 02 00	LDA \$0002
00457	C2A5	29 FC	AND #FC
00458	C2A7	80 02 00	STA \$0002 ; BACK TO BANK 0
00459	C2AA	A9 1B	LDA #1B
00460	C2AC	80 11 00	STA \$0011 ; BIT MAP MODE OFF
00461	C2AF	A9 C8	LDA #C8
00462	C2B1	80 16 00	STA \$0016 ; MULTI-COLOUR OFF
00463	C2B4	A9 15	LDA #15
00464	C2B6	80 13 00	STA \$0013 ; NORMAL SCREEN
00465	C2B9	60	RTS

```

00466 C2BA          20 FD AE          JSR CHKCOM          ; X1 AND Y1
00467 C2BA          20 EB B7          JSR PARAMS
00468 C2BA          8A              TXA
00469 C2BA          8D 3E 03          STA Y1
00470 C2BA          A9 00          LDA #00
00471 C2BA          8D 3F 03          STA Y1+1
00472 C2BA          8A              TXA
00473 C2BA          8D 3E 03          STA Y1
00474 C2BA          A9 00          LDA #00
00475 C2BA          8D 3F 03          STA Y1+1
00476 C2BA          8A              TXA
00477 C2BA          8D 3E 03          STA Y1
00478 C2BA          A9 00          LDA #00
00479 C2BA          8D 3F 03          STA Y1+1
00480 C2BA          8A              TXA
00481 C2BA          8D 3E 03          STA Y1
00482 C2BA          A9 00          LDA #00
00483 C2BA          8D 3F 03          STA Y1+1
00484 C2BA          8A              TXA
00485 C2BA          8D 3E 03          STA Y1
00486 C2BA          A9 00          LDA #00
00487 C2BA          8D 3F 03          STA Y1+1
00488 C2BA          8A              TXA
00489 C2BA          8D 3E 03          STA Y1
00490 C2BA          A9 00          LDA #00
00491 C2BA          8D 3F 03          STA Y1+1
00492 C2BA          8A              TXA
00493 C2BA          8D 3E 03          STA Y1
00494 C2BA          A9 00          LDA #00

```

LINE#	LOC	CODE	LTNE
00495	C2F5	R5 14	LDA #14
00496	C2F7	R5 FE	STA COL+1
00497	C2F9	RD 40 03	LDA X2
00498	C2FC	38	SEC
00499	C2FD	ED 3C 03	SBC X1
00500	C300	RD 44 03	STA XD
00501	C303	RD 41 03	LDA X2+1
00502	C306	ED 3D 03	SBC X1+1
00503	C309	RD 45 03	STA XD+1
00504	C30C	RD 42 03	LDA Y2
00505	C30F	38	SEC
00506	C310	ED 3E 03	SBC Y1
00507	C313	RD 46 03	STA YD
00508	C316	RD 3F 03	LDA Y1+1
00509	C319	ED 43 03	SBC Y2+1
00510	C31C	RD 47 03	STA YD+1
00511	C31F		;
00512	C31F		NEAREST DIAGONAL
00513	C31F		;
00514	C31F	R9 01	LDA #01
00515	C321	RD 5E 03	STA R0
00516	C324	RD 60 03	STA R1
00517	C327	R9 00	LDA #00
00518	C329	RD 5F 03	STA R0+1
00519	C32C	RD 61 03	STA R1+1
00520	C32F	RD 47 03	LDA YD+1 IF YD<0 THEN CHECKX

```

00521 C332 29 80 AND #80
00522 C334 F0 08 BEQ CHECKX
00523 C336 A9 FF LDA #FFF
00524 C338 80 5E 03 STA A0
00525 C33B 80 5F 03 STA A0+1
00526 C33E AD 45 03 CHECKX LDA XD+1
00527 C341 29 80 AND #80
00528 C343 F0 08 BEQ NRHOR
00529 C345 A9 FF LDA #FFF
00530 C347 80 60 03 STA A1
00531 C34A 80 61 03 STA A1+1
00532 C34D AD 45 03 LDA XD+1
00533 C350 29 80 AND #80
00534 C352 F0 1E BEQ POS1
00535 C354 AD 45 03 LDA XD+1
00536 C357 49 FF EOR #FFF
00537 C359 80 49 03 STA XE+1
00538 C35C 18 CLC
00539 C35D AD 44 03 LDA XD
00540 C360 49 FF EOR #FFF
00541 C362 69 01 ADC #01
00542 C364 80 48 03 STA XE
00543 C367 AD 49 03 LDA XE+1
00544 C36A 69 00 ADC #00
00545 C36C 80 49 03 STA XE+1
00546 C36F 4C 7E C3 JMP CHECKY
00547 C372 AD 44 03 LDA XD
00548 C375 80 48 03 STA XE
00549 C378 AD 45 03 LDA XD+1

; A0=-1
; IF XD<0 THEN NRHOR
; A1=-1
; XE=ABS(XD)

```

LINE#	LOC	CODE	LINE
00550	C37B	80 49 03	STA XE+1
00551	C37E	AD 47 03	LDA YD+1
00552	C381	29 80	CHECKY AND #F00
00553	C383	F0 1E	BEG POS2
00554	C385	AD 47 03	LDA YD+1
00555	C388	49 FF	EOR #FFF
00556	C38A	80 4B 03	STA YE+1
00557	C38D	18	CLC
00558	C38E	AD 46 03	LDA YD
00559	C391	49 FF	EOR #FFF
00560	C393	69 01	ADC #F01
00561	C395	80 4A 03	STA YE
00562	C398	AD 4B 03	LDA YE+1
00563	C39B	69 00	ADC #F00
00564	C39D	80 4B 03	STA YE+1
00565	C3A0	4C AF C3	JMP CALCD1
00566	C3A3	AD 46 03	POS2 LDA YD
00567	C3A6	80 4A 03	STA YE
00568	C3A9	AD 47 03	LDA YD+1
00569	C3AC	80 4B 03	STA YE+1
00570	C3AF		; NEAREST HORIZONTAL/VERTICAL
00571	C3AF		; NEAREST HORIZONTAL/VERTICAL
00572	C3AF		; NEAREST HORIZONTAL/VERTICAL
00573	C3AF	AD 48 03	CALCD1 LDA XE
00574	C3B2	38	SEC
00575	C3B3	ED 4A 03	SBC YE

; YE=ABS(YD)

00576	C386	80 58 03	STA D1
00577	C389	AD 49 03	LDA XE+1
00578	C38C	ED 48 03	SBC YE+1
00579	C38F	80 59 03	STA D1+1
00580	C3C2	29 80	AND #80
00581	C3C4	F0 3C	BEO L360
00582	C3C6	A9 FF	LDA #FF
00583	C3C8	80 5A 03	STA S0
00584	C3CB	80 5B 03	STA S0+1
00585	C3CE	A9 00	LDA #80
00586	C3D0	80 5C 03	STA S1
00587	C3D3	80 5D 03	STA S1+1
00588	C3D6	AD 4A 03	LDA YE
00589	C3D9	80 4C 03	STA LG
00590	C3DC	AD 4B 03	LDA YE+1
00591	C3DF	80 4D 03	STA LG+1
00592	C3E2	AD 48 03	LDA XE
00593	C3E5	80 4E 03	STA SH
00594	C3E8	AD 49 03	LDA XE+1
00595	C3EB	80 4F 03	STA SH+1
00596	C3EE	AD 47 03	LDA YD+1
00597	C3F1	29 80	AND #80
00598	C3F3	D0 46	BNE L380
00599	C3F5	A9 01	LDA #81
00600	C3F7	80 5A 03	STA S0
00601	C3FA	A9 00	LDA #80
00602	C3FC	80 5B 03	STA S0+1
00603	C3FF	4C 3B C4	JMP L380
00604	C402	A9 00	LDA #80

L360

LINE#	LOC	CODE	LINE
00605	C404	80 5A 03	STA S0
00606	C407	80 5B 03	STA S0+1
00607	C40A	A9 FF	LDA #FF
00608	C40C	80 5C 03	STA S1
00609	C40F	80 5D 03	STA S1+1
00610	C412	AD 48 03	LDA XE
00611	C415	80 4C 03	STA LG
00612	C418	AD 49 03	LDA XE+1
00613	C41B	80 4D 03	STA LG+1
00614	C41E	AD 4A 03	LDA YE
00615	C421	80 4E 03	STA SH
00616	C424	AD 4B 03	LDA YE+1
00617	C427	80 4F 03	STA SH+1
00618	C42A	AD 45 03	LDA XD+1
00619	C42D	29 80	AND #80
00620	C42F	00 0A	BNE L380 IF XD=0 THEN L380
00621	C431	A9 01	LDA #01
00622	C433	80 5C 03	STA S1
00623	C436	A9 00	LDA #00
00624	C438	80 5D 03	STA S1+1
00625	C43B		; S1=1
00626	C43B		; SET UP
00627	C43B		; L380
00628	C43B	AD 4C 03	LDA LG
00629	C43E	80 52 03	STA TT
00630	C441	AD 4D 03	LDA LG+1
			; TT=LG

```

00631 C444 8D 53 03 STA TT+1
00632 C447 AD 4E 03 LDA SH
00633 C44A 8D 50 03 STA TS
00634 C44D AD 4F 03 LDA SH+1
00635 C450 8D 51 03 STA TS+1
00636 C453 AD 4C 03 LDA LG
00637 C456 38 SEC
00638 C457 ED 4E 03 SBC SH
00639 C45A 8D 54 03 STA UO
00640 C45D AD 4D 03 LDA LG+1
00641 C460 ED 4F 03 SBC SH+1
00642 C463 8D 55 03 STA UO+1
00643 C466 4E 4D 03 LSR LG+1
00644 C469 6E 4C 03 ROR LG
00645 C46C AD 4E 03 LDA SH
00646 C46F 38 SEC
00647 C470 ED 4C 03 SBC LG
00648 C473 8D 56 03 STA CT
00649 C476 AD 4F 03 LDA SH+1
00650 C479 ED 4D 03 SBC LG+1
00651 C47C 8D 57 03 STA CT+1
00652 C47F
00653 C47F
00654 C47F
00655 C47F AD 3C 03 L420
00656 C482 85 59 LDR X1
00657 C484 AD 3D 03 LDR X2
00658 C487 85 5A LDR X1+1
00659 C489 AD 3E 03 LDR Y1

; WHILE MORE POINTS DO
;
;

```

LINE#	LOC	CODE	LINE
00660	C48C	85 56	STA T3
00661	C48E	A5 FE	LDA COL+1
00662	C490	85 FD	STA COL
00663	C492	20 E1 00	JSR PLOT
00664	C495	AD 57 03	LDA CT+1
00665	C498	29 90	RND #180
00666	C49A	F0 3C	BEQ L460
00667	C49C	AD 56 03	LDA CT
00668	C49F	18	CLC
00669	C4A0	6D 50 03	ADC TS
00670	C4A3	8D 56 03	STA CT
00671	C4A6	AD 57 03	LDA CT+1
00672	C4A9	6D 51 03	ADC TS+1
00673	C4AC	8D 57 03	STA CT+1
00674	C4AF	AD 3C 03	LDA X1
00675	C4B2	18	CLC
00676	C4B3	6D 5C 03	ADC S1
00677	C4B6	8D 3C 03	STA X1
00678	C4B9	AD 30 03	LDA X1+1
00679	C4BC	6D 5D 03	ADC S1+1
00680	C4BF	8D 3D 03	STA X1+1
00681	C4C2	AD 3E 03	LDA Y1
00682	C4C5	18	CLC
00683	C4C6	6D 5A 03	ADC S0
00684	C4C9	8D 3E 03	STA Y1
00685	C4CC	AD 3F 03	LDA Y1+1

; CALL PLOT POINT ROUTINE

; IF CT>=0 THEN L460

; CT=CT+TS

; X1=X1+S1

; Y1=Y1+S0

```

00696 C4CF 6D 5B 03 ADC S0+1
00697 C4D2 8D 3F 03 STA Y1+1
00698 C4D5 4C 11 C5 JMP L470
00699 C4D8 AD 56 03 LDA CT
00700 C4DB 38 L460
00701 C4DC ED 54 03 SEC
00702 C4DF 8D 56 03 SBC U0
00703 C4E2 AD 57 03 STA CT
00704 C4E5 ED 55 03 LDA CT+1
00705 C4E8 8D 57 03 SBC U0+1
00706 C4EB AD 3C 03 STA CT+1
00707 C4EE 18 LDA X1
00708 C4EF 6D 60 03 CLC
00709 C4F2 8D 3C 03 ADC A1
00710 C4F5 AD 3D 03 STA X1
00711 C4F8 6D 61 03 LDA X1+1
00712 C4FB 8D 3D 03 ADC A1+1
00713 C4FE AD 3E 03 STA X1+1
00714 C501 18 LDA Y1
00715 C502 6D 5E 03 CLC
00716 C505 8D 3E 03 ADC A0
00717 C508 AD 3F 03 STA Y1
00718 C50B 6D 5F 03 LDA Y1+1
00719 C50E 8D 3F 03 ADC A0+1
00720 C511 AD 52 03 STA Y1+1
00721 C514 38 LDA TT
00722 C515 E9 01 SEC #01
00723 C517 8D 52 03 STA TT
00724 C51A AD 53 03 LDA TT+1

```

; CT=CT-U0

; X1=X1+A1

; Y1=Y1+A0

; TT=TT-1

LINE#	LOC	CODE	LINE
00715	C51D	E9 00	SEC #00
00716	C51F	8D 53 03	STA TT+1
00717	C522	AD 53 03	LDA TT+1
00718	C525	F0 07	BEO L420A
00719	C527	C9 FF	CMP #FF
00720	C529	F0 0B	BEO RTN
00721	C52B	4C 7F C4	JMP L420
00722	C52E	AD 52 03	LDA TT
00723	C531	F0 03	BEO RTN
00724	C533	4C 7F C4	JMP L420
00725	C536	60	RTN
00726	C537		.FIL CHAR
00727	C537		;
00728	C537		; ROUTINE TO PLOT A CHARACTER
00729	C537		;
00730	C537	20 FD AE	JSR CHKCOM ; X AND Y
00731	C53A	20 EB B7	JSR PARAMS
00732	C53D	8A	TXA
00733	C53E	8D 64 03	STA YTL
00734	C541	A9 00	LDA #00
00735	C543	8D 65 03	STA YTL+1
00736	C546	A5 14	LDA #14
00737	C548	8D 62 03	STA XTL
00738	C54B	A5 15	LDA #15
00739	C54D	8D 63 03	STA XTL+1
00740	C550	20 FD AE	JSR CHKCOM ; COLOUR AND BRUSH

```

00741 C553 20 EB B7 JSR PARAMS
00742 C556 8A TXA
00743 C557 85 FC STA PER
00744 C559 A5 14 LDA #14
00745 C55B 85 FE STA COL+1
00746 C55D 20 FD AE JSR CHKCOM
00747 C560 20 EB B7 JSR PARAMS
00748 C563 8A TXA
00749 C564 80 66 03 STA CHAR
00750 C567 A5 14 LDA #14
00751 C569 80 67 03 STA RVORN
00752 C56C F0 05 BEQ MAIN
00753 C56E A9 FF LDA #FF
00754 C570 80 67 03 STA RVORN
00755 C573
00756 C573 ; CONVERT CHAR FROM ASCII TO POKE
00757 C573 ;
00758 C573 ;
00759 C576 AD 66 03 MAIN LDA CHAR
00760 C578 C9 1F CMP #1F
00761 C57A B0 03 BCS CONVCH
00762 C57D 4C 48 C6 JMP FIN
00763 C57F C9 40 CONVCH CMP #40
00764 C581 90 0E BCC CHAROK
00765 C582 18 CLC
00766 C584 C9 60 CMP #60
00767 C586 90 03 BCC CHCHAR
00768 C588 4C 48 C6 JMP FIN
00769 C589 38 CHCHAR SEC
00770 C58A E9 40 SBC #40

```

; RVORN AND CHAR

LINE#	LOC	CODE	LINE
00770	C58C	80 66 03	
00771	C58F	A9 00	STA CHAR
00772	C591	85 03	CHAROK LDA #\$00
00773	C593	A9 00	STA PNTR+1
00774	C595	85 02	LDA #\$00
00775	C597	AD 66 03	STA PNTR
00776	C59A	A0 03	LDA CHAR
00777	C59C	18	LDY #\$03
00778	C59D	0A	CLC
00779	C59E	8D 66 03	ASL A
00780	C5A1	A5 03	STA CHAR
00781	C5A3	69 00	LDA PNTR+1
00782	C5A5	85 03	ADC #\$00
00783	C5A7	AD 66 03	STA PNTR+1
00784	C5AA	88	LDA CHAR
00785	C5AB	D0 EF	DEY
00786	C5AD	85 02	BNE LOOP0
00787	C5AF	A9 08	STA PNTR
00788	C5B1	8D 6A 03	LDA #\$03
00789	C5B4	AD 0E DC	STA CNTR1
00790	C5B7	29 FE DC	LDA \$DC0E
00791	C5B9	8D 0E DC	AND #\$FE
00792	C5BC	A5 01	STA \$DC0E
00793	C5BE	29 FB	LDA \$01
00794	C5C0	85 01	AND #\$FB
00795	C5C2	A0 00	STA \$01
			LDY #\$00
			LOOP0
			LOOP01
			;
			START OF CHARACTER ROM
			;
			CHAR*8
			;
			FIRST LOCATION OF CHAR IN ROM
			;
			SWITCH IN CHARACTER ROM
			;

00796	C5C4	B1 02	LDA (PNTR),Y
00797	C5C6	80 66 03	STA CHAR
00798	C5C9	A5 01	LDA #01
00799	C5CB	09 04	ORA #04
00800	C5CD	85 01	STA #01
00801	C5CF	AD 0E DC	LDA \$DC0E
00802	C5D2	09 01	ORA #01
00803	C5D4	80 0E DC	STA \$DC0E
00804	C5D7	AD 67 03	LDA RYORN
00805	C5DA	F0 06	BEQ NORM
00806	C5DC	40 66 03	EOB CHAR
00807	C5DF	80 66 03	STA CHAR
00808	C5E2	AD 63 03	LDA XTL+1
00809	C5E5	80 69 03	STA XTEMP+1
00810	C5E8	AD 62 03	LDA XTL
00811	C5EB	80 68 03	STA XTEMP
00812	C5EE	A9 80	LDA #80
00813	C5F0	80 68 03	STA POINT
00814	C5F3	AD 66 03	LOOP02 LDA CHAR
00815	C5F6	20 68 03	AND POINT
00816	C5F9	F0 16	BEQ NXTPNT
00817	C5FB	AD 68 03	LDA XTEMP
00818	C5FE	85 59	STA T2
00819	C600	AD 69 03	LDA XTEMP+1
00820	C603	85 5A	STA T2+1
00821	C605	AD 64 03	LDA YTL
00822	C608	85 5B	STA T3
00823	C60A	A5 FE	LDA COL+1
00824	C60C	85 FD	STA COL

; GET BYTE

; SWITCH OUT CHARACTER ROM

; REVERSE BYTE

;X COORDINATE OF BIT

; LOOP FOR 8

; IF NOT SET, NEXT POINT

LINE#	LOC	CODE	LINE
00825	C60E	20 E1 C0	JSR PLOT
00826	C611	4E 68 03	NXTPNT LSR POINT
00827	C614	F0 17	BEG NXTLINE
00828	C616	A5 FB	LDA MODE
00829	C618	18	CLC
00830	C619	69 01	ADC #01
00831	C61B	18	CLC
00832	C61C	6D 68 03	ADC XTEMP
00833	C61F	8D 68 03	STA XTEMP
00834	C622	AD 69 03	LDA XTEMP+1
00835	C625	69 00	ADC #00
00836	C627	8D 69 03	STA XTEMP+1
00837	C62A	4C F3 C5	JMP LOOP02
00838	C62D	A5 02	NXTLINE LDA PNTR
00839	C62F	18	CLC
00840	C630	69 01	ADC #01
00841	C632	85 02	STA PNTR
00842	C634	A5 03	LDA PNTR+1
00843	C636	69 00	ADC #00
00844	C638	85 03	STA PNTR+1
00845	C63A	18	CLC
00846	C63B	EE 64 03	INC YTL
00847	C63E	80 08	BCS FIN
00848	C640	CE 6A 03	DEC CNTR1
00849	C643	F0 03	BEG FIN
00850	C645	4C B4 C5	JMP LOOP01

; PLOT POINT
; IF BYTE FINISHED, NEXT LINE
; INCREASE POINTER BY 1
; INCREASE Y COORD
; IF Y OUT OF BOUNDS, FINISH
; IF ALL 8 BYTES PLOTTED, FINISH

ERRORS = 00000

SYMBOL TABLE

SYMBOL	VALUE	A1	0360	ADD1	C188	BIT1	C124
A0	035E		0360				
BIT2	C135	BITOK	C115	BITST1	C122	BITST2	C133
BORDER	0920	BRUSH0	C10C	BRUSH1	C1F7	BRUSH2	C224
BRUSH3	C249	CALCD1	C3AF	CHAR	0366	CHAROK	C58F
CHCHAR	C589	CHECKX	C33E	CHECKY	C37E	CHKCOM	REFD
CLRCOL	C09C	CLRNEM	C04F	CLRSCN	C076	CNTR1	036A
COL	00FD	CONVCH	C57D	CT	0356	D1	0358
DONE	C031	FIN	C648	HIRES	C263	L0001	C066
L0002	C09C	L0003	C0B3	L360	C402	L380	C43B
L420	C47F	L420A	C52E	L460	C4D8	L470	C511
L6	034C	LINE	C2BA	LOOP	C059	LOOP0	C59C
LOOP01	C5B4	LOOP02	C5F3	LOOP1	C080	LOOP10	C11C
LOOP11	C12D	LOOP2	C0A6	LOOP20	C14E	LOOP21	C15B
MAIN	C573	MODE	00FB	MULT80	C16F	MULT81	C181
MULTI	C015	MULTI1	C1C5	NORM	C5E2	NRHOR	C34D
NXTLINE	C62D	NXTPNT	C611	OUT1	C075	OUT2	C09B
OUT3	C0C2	PARAMS	B7EB	PBR	00FC	PLOT	C0E1
PNTR	0002	POINT	036B	POS1	C372	POS2	C3A3

SYMBOL TABLE

RTN	C536	RVORN	0367	S0	035A	S1	035C
SCREEN	D021	SH	034E	T1	0057	T2	0059
T3	005B	T4	005D	T5	005E	T6	005F
T8	0350	TT	0352	UD	0354	UNPLOT	C20C
X1	033C	X2	0340	XD	0344	XE	0348
XER	C0F1	XOK	C0F2	XTEMP	0368	XTL	0362
Y1	033E	Y2	0342	YD	0346	YE	034A
YOK	C0F9	YTL	0364				

END OF ASSEMBLY

```

10000 I=49152:ER=0:PRINT"J"
10010 T=0
10020 READ A
10030 IF A>255 THEN 15000
10040 IF A=-1 THEN 16000
10050 POKE I,A:T=T+A
10060 I=I+1:GOTO 10020
15000 READ A$
15010 PRINT"J"A$ " ";
15020 IF A=T THEN PRINT" O.K.":GOTO 15500
15030 PRINTT,A;"M":ER=1
15500 GOTO 10010
16000 IF ER=1 THEN END
16010 POKE55,0:POKE56,128
16020 POKE51,0:POKE52,128
16030 CLR:NEW
20000 DATA32,253,174,32,235,183,138
20010 DATA141,32,208,141,33,208,165
20020 DATA20,133,251,240,2,162,0
20030 DATA32,79,192,32,118,192,32
20040 DATA156,192,169,59,141,17,208
20050 DATA169,29,141,24,208,165,251
20060 DATA240,5,169,216,141,22,208
20070 DATA169,128,133,56,133,52,173
20080 DATA2,221,9,3,141,2,221
20090 DATA173,0,221,41,252,9,1
20095 DATA8530,"LINES 20000-20090"
20100 DATA141,0,221,169,132,141,136
20110 DATA2,96,160,0,169,64,133
20120 DATA87,169,191,133,88,169,0
20130 DATA145,87,165,87,240,5,198
20140 DATA87,76,89,192,198,88,165
20150 DATA88,201,159,240,7,169,255
20160 DATA133,87,76,89,192,96,160
20170 DATA0,169,231,133,87,169,135
20180 DATA133,88,138,145,87,165,87
20190 DATA240,5,198,87,76,128,192
20195 DATA8868,"LINES 20100-20190"
20200 DATA198,88,165,88,201,131,240
20210 DATA7,169,255,133,87,76,128
20220 DATA192,96,160,0,169,231,133
20230 DATA87,169,219,133,88,169,0
20240 DATA145,87,165,87,240,5,198
20250 DATA87,76,166,192,198,88,165
20260 DATA88,201,215,240,7,169,255
20270 DATA133,87,76,166,192,96,32
20280 DATA253,174,32,235,183,165,20
20290 DATA133,89,165,21,133,90,138
20295 DATA9499,"LINES 20200-20290"
20300 DATA133,91,32,253,174,32,235

```

20310 DATA183,138,133,252,165,20,133
20320 DATA253,165,90,201,0,240,11
20330 DATA201,1,208,6,165,89,201
20340 DATA64,144,1,96,165,91,201
20350 DATA200,144,1,96,165,89,41
20360 DATA7,133,94,169,7,56,229
20370 DATA94,133,94,165,251,240,11
20380 DATA70,94,6,94,165,94,24
20390 DATA105,1,133,95,165,94,240
20395 DATA8366,"LINES 20300-20390"
20400 DATA9,168,169,1,10,136,208
20410 DATA252,240,2,169,1,133,94
20420 DATA165,95,240,9,168,169,1
20430 DATA10,136,208,252,240,2,169
20440 DATA1,133,95,169,0,133,92
20450 DATA133,88,133,87,165,91,41
20460 DATA7,133,93,165,91,74,74
20470 DATA74,133,91,160,5,24,10
20480 DATA38,88,136,208,249,133,87
20490 DATA165,91,160,3,24,10,136
20495 DATA7479,"LINES 20400-20490"
20500 DATA208,251,133,91,24,101,87
20510 DATA133,91,165,88,105,0,133
20520 DATA92,160,3,24,70,90,102
20530 DATA89,136,208,248,165,89,133
20540 DATA87,165,90,133,88,160,3
20550 DATA24,6,87,38,88,136,208
20560 DATA248,160,8,24,165,91,101
20570 DATA87,133,87,165,92,101,88
20580 DATA133,88,136,208,240,24,165
20590 DATA93,101,87,133,87,169,0
20595 DATA7746,"LINES 20500-20590"
20600 DATA101,88,24,169,160,101,88
20610 DATA133,88,24,165,89,101,91
20620 DATA133,91,169,0,101,92,133
20630 DATA92,169,54,133,1,165,251
20640 DATA208,3,76,99,194,165,252
20650 DATA201,0,240,17,201,1,240
20660 DATA40,201,2,240,81,201,3
20670 DATA240,114,169,55,133,1,96
20680 DATA160,0,165,94,73,255,133
20690 DATA94,165,95,73,255,133,95
20695 DATA8269,"LINES 20600-20690"
20700 DATA177,87,37,94,37,95,145
20710 DATA87,169,55,133,1,96,160
20720 DATA0,165,95,73,255,133,95
20730 DATA177,87,5,94,37,95,145
20740 DATA87,169,132,24,101,92,133
20750 DATA92,6,253,6,253,6,253
20760 DATA6,253,177,91,41,15,24

20770 DATA101,253,145,91,169,55,133
20780 DATA1,96,160,0,165,94,73
20790 DATA255,133,94,177,87,37,94
20795 DATA7456,"LINES 20700-20790"
20800 DATA5,95,145,87,24,169,132
20810 DATA101,92,133,92,177,91,41
20820 DATA240,24,101,253,145,91,169
20830 DATA55,133,1,96,160,0,177
20840 DATA87,5,94,5,95,145,87
20850 DATA169,216,24,101,92,133,92
20860 DATA165,253,145,91,169,55,133
20870 DATA1,96,160,0,165,252,240
20880 DATA35,177,87,5,94,145,87
20890 DATA169,132,24,101,92,133,92
20895 DATA7672,"LINES 20800-20890"
20900 DATA165,253,10,10,10,10,133
20910 DATA95,177,91,41,15,24,101
20920 DATA95,145,91,169,55,133,1
20930 DATA96,165,94,73,255,133,94
20940 DATA177,87,37,94,145,87,169
20950 DATA55,133,1,96,169,4,141
20960 DATA136,2,173,2,221,41,252
20970 DATA141,2,221,169,27,141,17
20980 DATA208,169,200,141,22,208,169
20990 DATA21,141,24,208,96,32,253
20995 DATA7566,"LINES 20900-20990"
21000 DATA174,32,235,183,138,141,62
21010 DATA3,169,0,141,63,3,165
21020 DATA20,141,60,3,165,21,141
21030 DATA61,3,32,253,174,32,235
21040 DATA183,138,141,66,3,169,0
21050 DATA141,67,3,165,20,141,64
21060 DATA3,165,21,141,65,3,32
21070 DATA253,174,32,235,183,138,133
21080 DATA252,165,20,133,254,173,64
21090 DATA3,56,237,60,3,141,68
21095 DATA7358,"LINES 21000-21090"
21100 DATA3,173,65,3,237,61,3
21110 DATA141,69,3,173,66,3,56
21120 DATA237,62,3,141,70,3,173
21130 DATA63,3,237,67,3,141,71
21140 DATA3,169,1,141,94,3,141
21150 DATA96,3,169,0,141,95,3
21160 DATA141,97,3,173,71,3,41
21170 DATA128,240,8,169,255,141,94
21180 DATA3,141,95,3,173,69,3
21190 DATA41,128,240,8,169,255,141
21195 DATA6422,"LINES 21100-21190"
21200 DATA96,3,141,97,3,173,69
21210 DATA3,41,128,240,30,173,69
21220 DATA3,73,255,141,73,3,24

21230 DATA173,68,3,73,255,105,1
21240 DATA141,72,3,173,73,3,105
21250 DATA0,141,73,3,76,126,195
21260 DATA173,68,3,141,72,3,173
21270 DATA69,3,141,73,3,173,71
21280 DATA3,41,128,240,30,173,71
21290 DATA3,73,255,141,75,3,24
21295 DATA6126,"LINES 21200-21290"
21300 DATA173,70,3,73,255,105,1
21310 DATA141,74,3,173,75,3,105
21320 DATA0,141,75,3,76,175,195
21330 DATA173,70,3,141,74,3,173
21340 DATA71,3,141,75,3,173,72
21350 DATA3,56,237,74,3,141,88
21360 DATA3,173,73,3,237,75,3
21370 DATA141,89,3,41,128,240,60
21380 DATA169,255,141,90,3,141,91
21390 DATA3,169,0,141,92,3,141
21395 DATA6404,"LINES 21300-21390"
21400 DATA93,3,173,74,3,141,76
21410 DATA3,173,75,3,141,77,3
21420 DATA173,72,3,141,78,3,173
21430 DATA73,3,141,79,3,173,71
21440 DATA3,41,128,208,70,169,1
21450 DATA141,90,3,169,0,141,91
21460 DATA3,76,59,196,169,0,141
21470 DATA90,3,141,91,3,169,255
21480 DATA141,92,3,141,93,3,173
21490 DATA72,3,141,76,3,173,73
21495 DATA6062,"LINES 21400-21490"
21500 DATA3,141,77,3,173,74,3
21510 DATA141,78,3,173,75,3,141
21520 DATA79,3,173,69,3,41,128
21530 DATA208,10,169,1,141,92,3
21540 DATA169,0,141,93,3,173,76
21550 DATA3,141,82,3,173,77,3
21560 DATA141,83,3,173,78,3,141
21570 DATA80,3,173,79,3,141,81
21580 DATA3,173,76,3,56,237,78
21590 DATA3,141,84,3,173,77,3
21595 DATA5637,"LINES 21500-21590"
21600 DATA237,79,3,141,85,3,78
21610 DATA77,3,110,76,3,173,78
21620 DATA3,56,237,76,3,141,86
21630 DATA3,173,79,3,237,77,3
21640 DATA141,87,3,173,60,3,133
21650 DATA89,173,61,3,133,90,173
21660 DATA62,3,133,91,165,254,133
21670 DATA253,32,225,192,173,87,3
21680 DATA41,128,240,60,173,86,3

21690 DATA24,109,80,3,141,86,3
21695 DATA6628,"LINES 21600-21690"
21700 DATA173,87,3,109,81,3,141
21710 DATA87,3,173,60,3,24,109
21720 DATA92,3,141,60,3,173,61
21730 DATA3,109,93,3,141,61,3
21740 DATA173,62,3,24,109,90,3
21750 DATA141,62,3,173,63,3,109
21760 DATA91,3,141,63,3,76,17
21770 DATA197,173,86,3,56,237,84
21780 DATA3,141,86,3,173,87,3
21790 DATA237,85,3,141,87,3,173
21795 DATA5475,"LINES 21700-21790"
21800 DATA60,3,24,109,96,3,141
21810 DATA60,3,173,61,3,109,97
21820 DATA3,141,61,3,173,62,3
21830 DATA24,109,94,3,141,62,3
21840 DATA173,63,3,109,95,3,141
21850 DATA63,3,173,82,3,56,233
21860 DATA1,141,82,3,173,83,3
21870 DATA233,0,141,83,3,173,83
21880 DATA3,240,7,201,255,240,11
21890 DATA76,127,196,173,82,3,240
21895 DATA6080,"LINES 21800-21890"
21900 DATA3,76,127,196,96,32,253
21910 DATA174,32,235,183,138,141,100
21920 DATA3,169,0,141,101,3,165
21930 DATA20,141,98,3,165,21,141
21940 DATA99,3,32,253,174,32,235
21950 DATA183,138,133,252,165,20,133
21960 DATA254,32,253,174,32,235,183
21970 DATA138,141,102,3,165,20,141
21980 DATA103,3,240,5,169,255,141
21990 DATA103,3,173,102,3,201,31
21995 DATA8214,"LINES 21900-21990"
22000 DATA176,3,76,72,198,201,64
22010 DATA144,14,24,201,96,144,3
22020 DATA76,72,198,56,233,64,141
22030 DATA102,3,169,208,133,3,169
22040 DATA0,133,2,173,102,3,160
22050 DATA3,24,10,141,102,3,165
22060 DATA3,105,0,133,3,173,102
22070 DATA3,136,208,239,133,2,169
22080 DATA8,141,106,3,173,14,220
22090 DATA41,254,141,14,220,165,1
22095 DATA6974,"LINES 22000-22090"
22100 DATA41,251,133,1,160,0,177
22110 DATA2,141,102,3,165,1,9
22120 DATA4,133,1,173,14,220,9
22130 DATA1,141,14,220,173,103,3

22140 DATA240,6,77,102,3,141,102
22150 DATA3,173,99,3,141,105,3
22160 DATA173,98,3,141,104,3,169
22170 DATA128,141,107,3,173,102,3
22180 DATA45,107,3,240,22,173,104
22190 DATA3,133,89,173,105,3,133
22195 DATA6274,"LINES 22100-22190"
22200 DATA90,173,100,3,133,91,165
22210 DATA254,133,253,32,225,192,78
22220 DATA107,3,240,23,165,251,24
22230 DATA105,1,24,109,104,3,141
22240 DATA104,3,173,105,3,105,0
22250 DATA141,105,3,76,243,197,165
22260 DATA2,24,105,1,133,2,165
22270 DATA3,105,0,133,3,24,238
22280 DATA100,3,176,8,206,106,3
22290 DATA240,3,76,180,197,96
22295 DATA6977,"LINES 22200-22290",-1
READY.

MULTI-COLOUR BAR CHART

This program uses the Hi-Resolution graphics routines to plot bar charts in multi-colour. The bar charts are given a three dimensional effect to make the display look nicer and more interesting.

The character plot routine is used to display a title and a reference for each bar (first character of each month) to plot twelve random bars.

RUNNING THE PROGRAM

There are no input parameters required for this program. Just run the program and see the display that appears.

This program can be modified to use data for the heights by changing line 130 to READ H and inserting the value of H before each of the month letters in line 1000.

PROGRAM STRUCTURE

- 10-30 Give the SYS call numbers variable names.
- 40 Enter Multi-Colour mode with the screen colour white.
- 50-90 Display 'EXAMPLE BAR-CHART' at the bottom of the screen.
- 100 Draw a line at the bottom of the graph area.
- 110 Set up plotting variables.
- 120-250 Main loop.
- 130-140 Get value of H and test for zero.
- 150-180 Plot front and side of bar using lines.
- 190-210 Plot top of bar.
- 220-240 Display character at bottom of bar.
- 250 Repeat until all 12 bars are drawn.
- 260 Wait for a key press.
- 70 Restore original screen and set character colours to black.
- 1000 Character data for bar titles.

```

1 REM ROUTINE TO PLOT A MULTI COLOUR
2 REM BAR CHART
3 REM
8 REM VARIABLES FOR EACH OF THE CALLS
9 REM
10 HIRES=49152:GRAPH=49182:CLEAR=49231
20 MEM=49308:PLOT=49347:NRM=49821
30 LINE=49850:CHAR=50487
34 REM
35 REM INTO MULTI MODE
36 REM
40 SYS HIRES,1,1
44 REM
45 REM DISPLAY A$ AT BOTTOM OF SCREEN
46 REM
50 A$=" EXAMPLE BAR-CHART ":X=0:Y=192
60 FOR I=1 TO LEN(A$)
70 SYS CHAR,X,Y,8,1,1,ASC(MID$(A$,I,1))
80 X=X+16
90 NEXT I
94 REM
95 REM DRAW A LINE AT BOTTOM OF GRAPH
96 REM
100 SYS LINE,0,181,319,181,14,1
104 REM
105 REM WIDTH,DEPTH,OFFSET
106 REM
110 W=17:D=10:O=W+4*D/5
114 REM
115 REM DISPLAY 12 RANDOM HEIGHT BARS
116 REM
120 FORI=12 TO 1 STEP -1
130 H=INT(RND(1)*170)
140 IF H=0 THEN Y=180:GOTO 190
144 REM
145 REM FRONT AND SIDE OF BAR
146 REM
150 FORY=180 TO 180-H STEP -1
160 SYS LINE,I*O,Y,I*O+W,Y,14,1
170 SYS LINE,I*O-2,Y,I*O-2-D,Y-D,6,2
180 NEXT Y
184 REM
185 REM TOP OF BAR
186 REM
190 FORJ=D-2 TO 0 STEP -1
200 SYS LINE,I*O-1-J,Y-J,I*O-1+W-J,Y-J,0,3
210 NEXT
214 REM
215 REM CHARACTERS UNDER EACH BAR
216 REM
220 READ A$

```

```
230 SYS CHAR,I*0,184,2,3,1,32
240 SYS CHAR,I*0,184,7,2,0,ASC(A$)
250 NEXT
254 REM
255 REM WAIT FOR A KEY PRESS
256 REM
260 GETA$:IFA$="" THEN 260
264 REM
265 REM NORMAL SCREEN,CLEAR COLOUR MEM
266 REM AND END
267 REM
270 SYS NRM:SYS MEM:END
994 REM
995 REM CHARACTER DATA
996 REM
1000 DATA D,N,O,S,A,J,J,M,A,M,F,J
READY.
```

3 DIMENSIONAL GRAPH

This program uses the Hi-Res routines given to display a three dimensional graph in standard Hi-Res. The function of the graph to be plotted is given in line 40 and the graph is plotted by lines 60-150.

RUNNING THE PROGRAM

As with the last program, there are no input parameters required. Just type RUN and see the display appear. The function is a symmetrical one and as it takes such a long time to plot up (approx 5 minutes for the display to be completed), the reflected image is displayed as well. The speed of the plotting is so slow because of the complicated calculations required to set the X and Y co-ordinates.

PROGRAM STRUCTURE

- 10-30 Give the SYS call numbers variable names.
- 40 Define the function to be plotted.
- 50 Enter standard Hi-Res mode with screen colour light grey.
- 60 Start of main loop.
- 70 Set up plotting variables.
- 80 Calculate y values.
- 90-150 Plot one line up.
- 140 If reflection not plotted, plot it.
- 150 Repeat until finished.
- 160 Wait for key press.
- 170 Normal screen and end.

```

1 REM 3D GRAPH IN STANDARD HIRES
2 REM *****
3 REM
8 REM VARIABLES FOR EACH OF THE CALLS
9 REM
10 HIRES=49152:GRAPH=49182:CLEAR=49231
20 MEM=49308:PLOT=49347:NRM=49821
30 LINE=49850:CHAR=50487
34 REM
35 REM FUNCTION OF GRAPH
36 REM
40 DEF FNA(Z)=38*(SIN(Z/24)+.48*SIN(3*Z/24))+20
44 REM
45 REMHIRES MODE
46 REM
50 SYS HIRES,0,12
54 REM
55 REM LOOP FOR HALF OF X VALUES
56 REM
60 FOR X=-100 TO 0 STEP 1
70 K=6:L=0:P=1:Z1=0
80 Y1=K*INT(SQR(10000-X*X)/K)
90 FOR Y=Y1 TO -Y1 STEP -K
100 Z=INT(80+FNA(SQR(X*X+Y*Y))-.707106*Y)
104 REM
105 REM IF OUT OF SIGHT, DO NOT PLOT
106 REM
110 IF Z<L THEN 150
120 M=1:L=Z
130 SYS PLOT,160+M*X,190-Z,0,1
134 REM
135 REM SYMMETRICAL GRAPH, PLOT OTHER
136 REM HALF OF X COORDINATES
137 REM
140 IF M=1 THEN M=-1:GOTO 130
150 NEXT Y:NEXT X
154 REM
155 REM WAIT FOR KEYPRESS
156 REM
160 GETA#:IFA#=""THEN160
164 REM
165 REM NORMAL SCREEN,CLEAR COLOUR MEM,END
166 REM
170 SYS NRM:SYS MEM:END
READY.

```

MULTI-COLOUR PATTERN

This program draws a pattern using two colours in multi-colour mode. Lines are drawn from a point at the top of the screen to the bottom line and from a point on the bottom of the screen to the top line. An interesting pattern appears from the cross over of the two sets of lines.

RUNNING THE PROGRAM

Just type RUN and watch the pattern appear. For a different pattern, change the first x and y co-ordinates in lines 60 and 70. An even more interesting pattern would appear if there was an exclusive OR plotting mode where if a point is off, turn it on and if a point is on, turn it off.

PROGRAM STRUCTURE

- 10-30 Give SYS call numbers variable names.
- 40 Enter Multi-colour mode with screen colour light grey.
- 50 Start of loop.
- 60 Draw line from top to bottom in red.
- 70 Draw line from bottom to top in white.
- 80 Repeat until complete.
- 90 Wait for key press.
- 100 Return to normal (text) screen and end.

```
1 REM PATTERN USING LINE PLOT IN MULTI
2 REM COLOUR MODE
3 REM
8 REM VARIABLES FOR EACH OF THE CALLS
9 REM
10 HIRES=49152:GRAPH=49182:CLEAR=49231
20 MEM=49308:PLOT=49347:NRM=49821
30 LINE=49850:CHAR=50487
34 REM
35 REM MULTI COLOUR MODE
36 REM
40 SYS HIRES,1,12
50 FOR I=0 TO 319 STEP 4
54 REM
55 REM DRAW LINE FROM 159,0 TO BOTTOM
56 REM
60 SYS LINE,159,0,1,200,2,1
64 REM
65 REM DRAW LINE FROM 158,199 TO TOP
66 REM
70 SYS LINE,158,199,1,0,1,2
80 NEXT I
84 REM
85 REM WAIT FOR KEY PRESS
86 REM
90 GETA$:IFA$="" THEN 90
94 REM NORMAL SCREEN,CLEAR COLOUR MEM,END
100 SYS NRM:SYS MEM:END
READY.
```


SINE GRAPH

This program draws a sine curve in two different ways. The first is to draw lines from the origin (0,100) to the point on the curve every 2 x co-ordinates. When this has been completed, the points are drawn with a resolution of every 1/2 x co-ordinate.

RUNNING THE PROGRAM

Just type RUN and watch the display appear. The display takes only about two minutes to appear compared with somewhere around one hour with all routines in Basic.

PROGRAM STRUCTURE

- 10-30 Give SYS call numbers variable names.
- 40 Enter standard Hi-Res mode with screen colour black.
- 50 Loop for the line plot.
- 60 Plot the line.
- 70 Repeat lines until complete.
- 80 Loop for the point plot.
- 90 Plot the points.
- 100 Repeat until complete.
- 110 Wait for key press.
- 120 Return to normal screen and end.

```

1 REM STANDARD HIRES LINES TO POINTS
2 REM ON A SINE CURVE
3 REM
8 REM VARIABLES FOR EACH OF THE CALLS
9 REM
10 HIRES=49152:GRAPH=49182:CLEAR=49231
20 MEM=49308:PLOT=49347:NRM=49821
30 LINE=49850:CHAR=50487
34 REM
35 REM STANDARD MODE
36 REM
40 SYS HIRES,0,0
44 REM
45 REM LOOP FOR X VALUES
46 REM
50 FOR I=0 TO 319 STEP 2
54 REM
55 REM PLOT THE LINES FROM 0,100 TO POINT
56 REM
60 SYS LINE,0,100,I,100-90*SIN(I*π/60),1,1
70 NEXT
74 REM
75 REM PLOT IN THE CURVE
76 REM
80 FOR I=0 TO 319 STEP .5
90 SYS PLOT,I,100-90*SIN(I*π/60),1,1
100 NEXT
104 REM
105 REM WAIT FOR KEY PRESS
106 REM
110 GETA$:IFA$="" THEN 110
114 REM
115 REM NORMAL SCREEN,END
116 REM
120 SYS NRM:END
READY.

```

DEMO OF ROUTINES 5 AND 7 AND A CIRCLE PLOTTING ROUTINE

This program plots a circle on the screen by calculating only one eighth of the circle and plotting eight points for the eight combinations of DC, XD, YC, and YD. The program then just demonstrates the use of the routines 5 and 7 to switch between the screen before the program ends.

RUNNING THE PROGRAM

Type RUN and enter the circle parameters. The circle will then be drawn. Press a key to change to the normal screen. Press a key again and the program will change back to the Hi-Res screen. Finally pressing another key will return to the normal screen and end.

PROGRAM STRUCTURE

- 10-20 Give SYS call numbers variable names.
- 30-31 Input and test x and y centre co-ordinates and radius.
- 40 Enter Multi-Colour mode with screen colour light grey.
- 50 Start loop for circle.
- 60-70 Calculate offsets from centre.
- 80-150 Plot the eight points.
- 160 Repeat until circle complete.
- 170 Wait for key press.
- 180 Return to normal screen.
- 190 Wait for key press.
- 200 Return to graphics screen.
- 210 Wait for key press.
- 220 Return to normal screen and end.

```

1 REM DEMO OF GRAPH AND NORM COMMANDS
2 REM AND A CIRCLE PLOTTING ROUTINE
3 REM
8 REM VARIABLES FOR EACH OF THE CALLS
9 REM
10 HIRES=49152:GRAPH=49182:CLEAR=49231
20 MEM=49308:PLOT=49347:NRM=49821
30 INPUT "X,Y CENTRE AND RAD";XC,YC,RA
31 IF RA>YC OR RA>XC THENPRINT "T":GOTO30
34 REM
35 REM CHOOSE MULTI,X AND Y CENTRE,
36 REM AND RADIUS
37 REM
40 SYS HIRES,1,12
44 REM
45 REM LOOP FOR 1/8 OF FULL CIRCLE
46 REM
50 FOR I=PI/4 TO PI/2 STEP 1/RA
54 REM
55 REM CALCULATE X AND Y DIFFERENCES
56 REM
60 XD=RA*COS(I)
70 YD=RA*SIN(I)
74 REM
75 REM PLOT EIGHT POINTS FOR EACH CALC
76 REM
80 SYS PLOT,XC+XD,YC+YD,2,1
90 SYS PLOT,XC-XD,YC-YD,2,1
100 SYS PLOT,XC+XD,YC-YD,2,1
110 SYS PLOT,XC-XD,YC+YD,2,1
120 SYS PLOT,XC+YD,YC+XD,2,1
130 SYS PLOT,XC-YD,YC-XD,2,1
140 SYS PLOT,XC+YD,YC-XD,2,1
150 SYS PLOT,XC-YD,YC+XD,2,1
160 NEXT I
164 REM
165 REM WAIT FOR KEY PRESS
166 REM
170 GETA$:IFA$="" THEN 170
174 REM
175 REM RETURN TO NORMAL SCREEN
176 REM
180 SYS NRM
184 REM
185 REM WAIT FOR KEY PRESS
186 REM
190 GETA$:IFA$="" THEN 190
194 REM
195 REM RETURN TO GRAPHICS SCREEN
196 REM

```

```
200 SYS GRAPH
204 REM
205 REM WAIT FOR KEY PRESS
206 REM
210 GETA$:IFA$="" THEN 210
214 REM
215 REM NORMAL SCREEN AND END
216 REM
220 SYS NRM:END
READY.
```

DIGICLOCK

This program displays the time as input from the user in the form of a twenty four hour digital clock. The majority of the program is for setting up the Sprites required for the display.

RUNNING THE PROGRAM

After a pause for reading in the Sprites, the user is requested to input the time in the form of HHMMSS. As soon as the return key has been pressed, the time will start. The display will continue until the stop key has been pressed to stop the program.

PROGRAM STRUCTURE

100	Set up Sprites.
110 – 140	Input time and set in TI\$.
150 – 330	Set up all registers for the Sprites. Includes X and Y co-ordinates, Sprite colours, Enlargement, and pointers.
340 – 355	Display titles.
370 – 430	Display Sprites in correct positions.
440 – 460	Flash second double dot character and pause for change in TI\$.
31000 – 30110	Data for Sprites.
31000 – 31050	Read Sprites into memory.

```

10 REM DIGICLOCK USING SPRITES FOR THE
20 REM NUMBERS.
30 REM
100 GOSUB 31000:REM GET SPRITES
110 PRINT":PLEASE ENTER THE TIME IN THE FORMAT"
120 PRINT"      HHMMSS";
130 INPUTTA$
140 IF LEN(TA$)=6 THEN TI$=TA$
150 V=53248
160 FORI=0TO7
170 POKEV+I*2;(I*40+20)AND255
180 POKEV+I*2+1,130
220 POKEV+39+I,7
230 IFI=2OR I=5THENPOKEV+39+I,0
240 POKE2040+I,244
250 NEXT
260 POKEV+33,2
270 POKEV+32,0
280 POKEV+21,255
290 POKEV+23,255
300 POKEV+29,255
310 POKEV+16,0
320 POKE2042,255:POKE2045,255
330 POKEV+16,192
340 PRINT"===== HOURS          ";
345 PRINT"MIN.           SECS."
350 PRINT"      _____";
355 PRINT"      _____"
370 FORI=0TO7
380 IFI=2OR I=5THEN430
390 A=I+1:IFA>6THENA=A-1
400 IFA>3THENA=A-1
410 B=VAL(MID$(TI$,A,1))
420 POKE2040+I,B+245
430 NEXT
440 POKEV+44,0:CLR:T$=TI$:V=53248
450 IFTI$=T$THEN450
460 POKEV+44,2:GOTO370
30000 DATA246
30001 DATA0,0,0,0,0,0,0
30002 DATA0,0,0,0,0,0,28
30003 DATA0,0,36,0,0,68,0
30004 DATA0,132,0,1,4,0,1
30005 DATA100,0,1,164,0,0,36
30006 DATA0,0,36,0,0,36,0
30007 DATA0,36,0,0,36,0,0
30008 DATA36,0,0,36,0,1,231
30009 DATA128,1,0,128,1,255,128
30010 DATA247
30011 DATA0,0,0,0,0,0,0
30012 DATA0,0,0,0,0,0,126

```

30013 DATA0,0,129,0,1,60,128
30014 DATA2,66,64,2,66,64,2
30015 DATA66,64,3,194,64,0,1
30016 DATA128,0,9,0,0,18,0
30017 DATA0,36,0,0,72,0,0
30018 DATA144,0,1,32,0,2,127
30019 DATA192,2,0,64,3,255,192
30020 DATA248
30021 DATA0,0,0,0,0,0,0
30022 DATA0,0,0,0,0,0,124
30023 DATA0,0,130,0,1,57,0
30024 DATA2,68,128,2,68,128,3
30025 DATA196,128,0,4,128,0,9
30026 DATA0,0,18,0,0,9,0
30027 DATA0,4,128,3,196,128,2
30028 DATA68,128,2,68,128,1,57
30029 DATA0,0,130,0,0,124,0
30030 DATA249
30031 DATA0,0,0,0,0,0,0
30032 DATA0,0,0,0,0,0,15
30033 DATA0,0,17,0,0,33,0
30034 DATA0,73,0,0,153,0,1
30035 DATA41,0,2,73,0,4,137
30036 DATA0,4,249,224,4,0,32
30037 DATA7,249,204,0,9,0,0
30038 DATA9,0,0,9,0,0,9
30039 DATA0,0,9,0,0,15,0
30040 DATA250
30041 DATA0,0,0,0,0,0,0
30042 DATA0,0,0,0,0,3,255
30043 DATA192,2,0,64,2,127,192
30044 DATA2,64,0,2,64,0,2
30045 DATA64,0,2,126,0,2,1
30046 DATA0,3,252,128,0,2,64
30047 DATA0,2,64,0,2,64,3
30048 DATA194,64,2,66,64,1,60
30049 DATA128,0,129,0,0,126,0
30050 DATA251
30051 DATA0,0,0,0,0,0,0
30052 DATA0,0,0,0,0,0,127
30053 DATA0,0,128,128,1,62,64
30054 DATA2,65,192,2,64,0,2
30055 DATA64,0,2,64,0,2,126
30056 DATA0,2,1,0,2,60,128
30057 DATA2,66,64,2,66,64,2
30058 DATA66,64,2,66,64,1,60
30059 DATA128,0,129,0,0,126,0
30060 DATA252
30061 DATA0,0,0,0,0,0,0
30062 DATA0,0,0,0,0,3,255
30063 DATA192,2,0,64,3,254,64


```

30064 DATA0,2,64,0,2,64,0
30065 DATA4,128,0,4,128,0,9
30066 DATA0,0,9,0,0,18,0
30067 DATA0,18,0,0,36,0,0
30068 DATA36,0,0,72,0,0,72
30069 DATA0,0,144,0,0,240,0
30070 DATA253
30071 DATA0,0,0,0,0,0,0
30072 DATA0,0,0,0,0,0,124
30073 DATA0,0,130,0,0,1,57,0
30074 DATA2,68,128,2,68,128,2
30075 DATA68,128,2,68,128,1,57
30076 DATA0,0,130,0,0,1,57,0
30077 DATA2,68,128,2,68,128,2
30078 DATA68,128,2,68,128,1,57
30079 DATA0,0,130,0,0,0,124,0
30080 DATA254
30081 DATA0,0,0,0,0,0,0
30082 DATA0,0,0,0,0,0,126
30083 DATA0,0,129,0,0,1,60,128
30084 DATA2,66,64,2,66,64,2
30085 DATA66,64,2,66,64,1,60
30086 DATA64,0,128,64,0,126,64
30087 DATA0,2,64,0,2,64,0
30088 DATA2,64,3,130,64,2,124
30089 DATA128,1,1,0,0,254,0
30090 DATA245
30091 DATA0,0,0,0,0,0,0
30092 DATA0,0,0,0,0,0,255
30093 DATA0,1,0,128,2,126,64
30094 DATA4,129,32,4,130,32,4
30095 DATA132,32,4,137,32,4,147
30096 DATA32,4,165,32,4,201,32
30097 DATA4,145,32,4,33,32,4
30098 DATA65,32,4,129,32,2,126
30099 DATA64,1,0,128,0,255,0
30100 DATA255
30101 DATA0,0,0,0,0,0,0
30102 DATA0,0,0,0,0,0,60
30103 DATA0,0,126,0,0,126,0
30104 DATA0,126,0,0,126,0,0
30105 DATA60,0,0,0,0,0,0
30106 DATA0,0,0,0,0,0,0
30107 DATA0,0,0,0,60,0,0
30108 DATA126,0,0,126,0,0,126
30109 DATA0,0,126,0,0,60,0
30110 DATA-1
31000 READA: IFA=-1 THEN 31040
31010 FORI=0 TO 62: READX

```

```
31020 POKE64*A+I,X: NEXT
31030 GOTO31000
31040 FORI=0TO62: POKE244*64+I,0: NEXT
31050 RETURN
READY.
```

DISASSEMBLER

This program disassembles an area in memory and displays it on the screen. The start address is input by the user and after every phase of disassembly a new start address is requested. To continue the disassembly from the point that the start is requested just press return.

RUNNING THE PROGRAM

After typing RUN, the user will be asked for the start address of the code to be disassembled. Enter this in decimal and press return. After a full page has been displayed, a new start address will be asked for. At this point, enter a new address (or just return if to continue from that point).

PROGRAM STRUCTURE

A brief explanation of the program follows.

10 Read the op codes.

20 Input start address.

40 – 690 Main loop of disassembler. M stores the line on the screen.

710 – 790 Display decimal number in hex.

800 – 1110 Data for the operation codes.

```

10 DIMT$(255):FORI=0TO255:READT$(I):NEXT
20 INPUT"DECIMAL START ADDRESS";A
30 PRINT" ";
40 FORM=1TO24
50 I=PEEK(A)
60 K=(I)AND15
70 IFI=0THENL=1:GOTO260
80 IFK=8THENL=1:GOTO260
90 IFK=10THENL=1:GOTO260
100 IFK<>0THENL40
110 IFI=32THENL80
120 IF(I)AND16THENL40
130 IFI<128THENL=1:GOTO260
140 IFK=0THEN220
150 IFK=9THEN190
160 IFK<10THEN250
170 J=PEEK(I)
180 L=3:GOTO260
190 K=(I)AND16
200 IFK=0THEN250
210 GOTO180
220 K=(I)AND240
230 IFK>111THEN250
240 IFK=32THEN180
250 L=2
260 N=A:GOSUB710
270 PRINT" ";
280 IFT$(I)="?"THENL=1
290 FORX=A+L-1
300 N=PEEK(X):GOSUB710:PRINT" ";
310 NEXT
320 PRINT:PRINT"#####";
330 IFT$(I)<>"?"THEN350
340 PRINT",BYT  $";N=I:L=1:GOSUB710:GOTO620
350 PRINTT$(I);
360 IFL=1THEN620
370 PRINT" ";
380 X=(I)AND15:Y=(I)AND240
390 IFL=3THEN530
400 ONX+1GOTO410,470,460,,500,500,500,,460
410 IF(Y)AND16THEN430
420 GOTO460
430 N=PEEK(A+1):IFN>127THENN=- (256-N)
440 PRINT"$";N=A+N+2
450 GOSUB710:GOTO620
460 PRINT"#$";N=PEEK(A+1):GOSUB710:GOTO620
470 PRINT"($";N=PEEK(A+1):GOSUB710
480 IF(Y)AND16THENPRINT"),Y";GOTO620
490 PRINT",X";GOTO620
500 PRINT"$";N=PEEK(A+1):GOSUB710
510 IF(Y)AND16THENPRINT",X";

```

```

520 GOT0620
530 ONX+1GOT0560,,,,,,540,,590,550,550
540 PRINT"$";N=PEEK(A+2):GOSUB710
545 N=PEEK(A+1):GOSUB710:PRINT",Y";:GOT0620
550 IFI=200THEN540
560 PRINT"$";N=PEEK(A+2):GOSUB710
565 N=PEEK(A+1):GOSUB710
570 IF(Y)AND16THENPRINT",X";
580 GOT0620
590 IJI=198THEN560
600 IFI<102THEN560
610 PRINT("$");N=PEEK(A+2):GOSUB710
615 N=PEEK(A+1):GOSUB710:PRINT")";
620 PRINT
625 PRINT"#####";
630 FORJ=1TOL
640 X=PEEK(A+J-1)
650 IFX<32THENX=X+32
660 IFX>127THENX=X-128:GOT0650
670 PRINTCHR$(X);:NEXT
680 A=A+L
690 PRINT: NEXT
700 GOT020
710 A$="0123456789ABCDEF"
720 H$=""
730 K=N-INT(N/16)*16
740 N=INT(N/16)
750 H$=MID$(A$,K+1,1)+H$
760 IFN>0THEN730
770 IFLEN(H$)=1THENPRINT"0";
780 PRINTH$;
790 RETURN
800 DATABRK,ORA,?,?,?,ORA,ASL,?
810 DATAPHP,ORA,ASL,?,?,ORA,ASL,?
820 DATABPL,ORA,?,?,?,ORA,ASL,?
830 DATACLC,ORA,?,?,?,ORA,ASL,?
840 DATAJSR,AND,?,?,BIT,AND,ROL,?
850 DATAPLP,AND,ROL,?,BIT,AND,ROL,?
860 DATABMI,AND,?,?,?,AND,ROL,?
870 DATASEC,AND,?,?,?,AND,ROL,?
880 DATARTI,EOR,?,?,?,EOR,LSR,?
890 DATAPHA,EOR,LSR,?,JMP,EOR,LSR,?
900 DATABVC,EOR,?,?,?,EOR,LSR,?
910 DATACLI,EOR,?,?,?,EOR,LSR,?
920 DATARTS,ADC,?,?,?,ADC,ROR,?
930 DATAPLA,ADC,ROR,?,JMP,ADC,ROR,?
940 DATABVS,ADC,?,?,?,ADC,ROR,?
950 DATASEI,ADC,?,?,?,ADC,ROR,?
960 DATA?,STA,?,?,STY,STA,STX,?
970 DATAIEY,?,TXA,?,STY,STA,STX,?

```

980 DATABCC, STA, ?, ?, STY, STA, STX, ?
990 DATATYA, STA, TXS, ?, ?, STA, ?, ?
1000 DATALDY, LDA, LIX, ?, LDY, LIA, LIX, ?
1010 DATATAY, LDA, TAX, ?, LDY, LDA, LDX, ?
1020 DATABCS, LDA, ?, ?, LDY, LDA, LDX, ?
1030 DATACLV, LDA, TSX, ?, LDY, LDA, LIX, ?
1040 DATACPY, CMP, ?, ?, CPY, CMP, DEC, ?
1050 DATAINY, CMP, DEX, ?, CPY, CMP, DEC, ?
1060 DATABNE, CMP, ?, ?, ?, CMP, DEC, ?
1070 DATACLI, CMP, ?, ?, ?, CMP, DEC, ?
1080 DATACPX, SBC, ?, ?, CPX, SBC, INC, ?
1090 DATAINX, SBC, NOP, ?, CPX, SBC, INC, ?
1100 DATABEQ, SBC, ?, ?, ?, SBC, INC, ?
1110 DATASED, SBC, ?, ?, ?, SBC, INC, ?
READY.

HANGMAN

Hangman is the computerised version of the children's game, although playing against the computer enlivens the proceedings well enough for adults to find the game challenging. The player guessing the word has ten moves in which to work his miracle (the computer always plays the questioner), no second chances are allowed as first the ground appears and so on until, in the usual format of hangman, the player finds himself either the winner or the corpse. In this particular program the listings contain some 200 of the more commonly misspelt words in the English language.

When typing in the words the user must ensure accuracy in his own copying. The possibility of the player being unable to play the game as intended, is negated by the sheer numerical volume of the words.

By changing the words in the data statements in lines 9500 to 9998 – always ensuring that there are 200 words contained therein – it is possible to use this game as an aid to learning a foreign language.

RUNNING THE PROGRAM

Load, type 'RUN' and follow the instructions.

PROGRAM STRUCTURE

For an interesting reference, the following listing of lines has been categorised for the user's convenience.

9 - 240	Instructions.
280 - 420	Choose a word from the data statements.
430 - 450	Input a letter.
460 - 490	Not a letter.
510 - 520	Is a letter, check if in word.
530 - 580	Guessed the word.
590 - 660	Letter in word.
670 - 720	Print letter in the correct position on the screen.
760 - 800	Letter is not in the word.
810 - 840	Display the man's feet.
850 - 880	Failed to guess the word before the man was hung.

890 - 920	Display the base of the rig.
930 - 990	Display the upright.
1000 - 1030	Display the horizontal.
1040 - 1070	Display the diagonal support.
1080 - 1110	Display the rope.
1120 - 1190	Display the man's head.
1200 - 1270	Display the man's body.
1280 - 1310	Display the man's hands.
1320 - 1380	Display the man's legs.
1570 - 2210	Data containing about 200 words.


```

9 POKE5328,2:POKE53281,12
10 PRINT"(C)1981#####HANGMAN"
20 PRINT"####"
30 PRINT"### DO YOU NEED INSTRUCTIONS?"
40 PRINT" (REPLY Y FOR 'YES' OR N FOR 'NO') "
50 GET X$:IF X#=""THEN 50
60 IF X#="N" THEN 260
70 PRINT"(C)1981 ALWAYS REPLY Y FOR YES,OR N FOR NO."
80 PRINT"I WILL PRINT:##"
90 PRINT" UNKNOWN WORD=----"
100 PRINT"EACH - STANDS FOR A MISSING LETTER."
110 PRINT
120 PRINT"WHEN I PRINT 'PRESS ANY LETTER',"
130 PRINT"FOLLOW THE INSTRUCTION !#"
140 PRINT"IF A LETTER IS IN THE UNKNOWN WORD"
150 PRINT"MORE THAN ONCE YOU WILL NEED TO PRESS"
160 PRINT"THE LETTER MORE THAN ONCE.##"
170 PRINT"E.G. 'APPLE' HAS 2 P'S,PRESS P ONCE AND"
180 PRINT"I REPLY: UNKNOWN WORD=-P----##"
190 PRINT"PRESS P A SECOND TIME AND I REPLY"
200 PRINT" UNKNOWN WORD=-PP--"
210 PRINT
220 PRINT"YOU HAVE TO COMPLETE THE WORD BEFORE"
230 PRINT"I HANG THE MAN."
240 PRINT"PRESS ANY KEY WHEN READY#"
250 GET X$: IF X#="" THEN 250

260 PRINT"(C)1981 ;TAB(16):"##HANGMAN"
270 PRINTTAB(16):"###";
280 RA=INT(RND(1)*197+1)
290 IF A#="END" THEN 500
300 FOR I=1 TO RA
310 READ A$: IF A#="END" THEN 500
320 NEXT I
330 RESTORE
340 LA=LEN(A#)
350 Z#=CHR$(18)+A#
360 I#=""
370 FOR I=1 TO LA
380 I#=I#+"- "
390 NEXT I
400 PRINT"##"
410 PRINTTAB(7);" ##UNKNOWN WORD=";I#;"###";
420 N=0:L=0
430 PRINT"(C)1981 'PRESS ANY LETTER' ##";
440 GET L$: IF L#="" THEN 440
450 IF L#<>" " THEN 500
460 PRINT"(C)1981 ' IS NOT A LETTER "
470 GOSUB 1540
480 PRINT"(C)1981 SPC(35)"###";
490 GOTO 430

```

```

500 GOSUB 590
510 IF N<LA THEN 430
520 IF N=(LA+1) THEN T1=TI:GOTO 540
530 PRINT"X(0000) YOU WIN!!                ":T1=TI
540 IF(TI-T1)<200 THEN 540
550 PRINT"DO YOU WANT ANOTHER GO?"
560 GET B$:IF B$=""THEN 560
570 IF B$="Y" THEN 260
580 PRINT"THANK-YOU FOR PLAYING":STOP
590 LF=0
600 FOR I=1 TO LA
610 B$=MID$(A$,I,1)
620 IF B$<>L$ THEN 730
630 PRINT"X(0000) LETTER IN WORD      S"
640 N=N+1
650 GOSUB 1540
660 IA=19+I
670 PRINT"X";
680 PRINT TAB(IA);L$;"X";
690 II=I
700 GOSUB 1390
710 LF=LF+1
720 RETURN
730 NEXT I
740 IF LF=0 THEN GOSUB 760
750 RETURN
760 L=L+1
770 PRINT"X(0000) LETTER NOT IN WORDS"
780 GOSUB 1540
790 IFL=9THEN1320
791 ONL GOSUB 890,930,1000,1040,1080,1120,1200,1280
800 IF L<10 THEN RETURN
810 K=21
820 GOSUB 1500
830 PRINTTAB(19);"X..X.X.X.X.X.X";
840 PRINT"X"
850 PRINTTAB(20);Z$
860 PRINT"X.SORRY, YOU LOST      S";
870 N=LA+1
880 RETURN
890 K=23
900 GOSUB 1500
910 PRINT"X_____X";
920 RETURN
930 K=6
940 GOSUB 1500
950 FOR I=1 TO 18
960 PRINTTAB(12);"X.X"
970 NEXT I
980 PRINT"X";

```

```

990 RETURN
1000 K=6
1010 GOSUB 1500
1020 PRINTTAB(13);"██████████";
1030 RETURN
1040 K=11
1050 GOSUB 1500
1060 PRINTTAB(13);"███/███/███/███/███";
1070 RETURN
1080 K=7
1090 GOSUB 1500
1100 PRINTTAB(22);"I";
1110 RETURN
1120 K=8
1130 GOSUB 1500
1140 PRINTTAB(20);"███"
1150 PRINTTAB(19);"██/++██"
1160 PRINTTAB(19);"██Y I P"
1170 PRINTTAB(20);"██/██"
1180 PRINTTAB(21);"███";
1190 RETURN
1200 K=12
1210 GOSUB 1500
1220 PRINTTAB(19);"███"
1230 PRINTTAB(18);" I↑↑ I"
1240 PRINTTAB(18);" I↑ I"
1250 PRINTTAB(18);" I↑ I"
1260 PRINTTAB(20);"███";
1270 RETURN
1280 K=16
1290 GOSUB 1500
1300 PRINTTAB(19);"███#███";
1310 RETURN
1320 K=17
1330 GOSUB 1500
1340 FOR I=1 TO 4
1350 PRINTTAB(20);"███ I"
1360 NEXT I
1370 PRINT"█";
1380 RETURN
1390 X=II-1:Y=LA-II
1400 IF X=0 AND Y=0 THEN 1450
1410 IF X=0 THEN A$="+RIGHT$(A$,Y):RETURN
1420 IF Y=0 THEN A$=LEFT$(A$,X)+" ":RETURN
1430 A$=LEFT$(A$,X)+" "+RIGHT$(A$,Y)
1440 RETURN
1450 A$=""
1460 FOR I=1 TO LA
1470 A$=A$+" "
1480 NEXT I
1490 RETURN

```

```

1500 FOR I=2 TO K
1510 PRINT "M";
1520 NEXT I
1530 RETURN
1540 T=TI
1550 IF (TI-T)<100 THEN 1550
1560 RETURN
1570 DATA ABSENCE, ACCEPT, ACCIDENTALLY
1580 DATA ACCOMMODATE, ACHIEVED, ACKNOWLEDGE
1590 DATA ACQUAINTED, ADDRESSES, AERIAL
1600 DATA AGGRAVATE, AGREEABLE, AMATEUR, AMONG
1610 DATA ANTARCTIC, ANXIETY, APPARENT, APPEARANCE
1620 DATA APPROPRIATE, ARCTIC, ARGUMENT, ARRANGEMENTS
1630 DATA ASCENT, ATHLETIC, AWFUL, BACHELOR
1640 DATA BEGINNING, BELIEVED, BENEFITED, BREATHE
1650 DATA BRITAIN, BUSINESS, CAPTAIN, CEILING
1660 DATA CEMETERY, CERTAIN, CHOICE, CLOTHES
1670 DATA COLLEGE, COMING, COMMITTEE, COMPARATIVE
1680 DATA COMPETENT, COMPLETELY, CONSCIENTIOUS
1690 DATA CONSCIOUS, CONSISTENT, CONVENIENCE
1700 DATA COPIES, COURSE, COURTEOUS, COURTESY
1710 DATA CRITICISM, DECEIVE, DECISION, DEFINITE
1720 DATA DESIRABLE, DESPERATE, DISAPPEARED
1730 DATA DISAPPOINTED, DISASTROUS, DISCIPLINE
1740 DATA DISSATISFIED, EFFICIENCY, EIGHTH
1750 DATA ELIMINATED, EMBARRASSED, EMPHASIZE
1760 DATA ENTHUSIASM, EQUIPPED, ESPECIALLY
1770 DATA ESSENTIAL, EXAGGERATED, EXCELLENT
1780 DATA EXERCISE, EXHAUSTED, EXISTENCE
1790 DATA EXPENSE, EXPERIENCE, FAMILIAR
1800 DATA FEBRUARY, FINANCIAL, FOREIGN
1810 DATA FORMERLY, FORTY, FRIEND, GAUGE
1820 DATA GENIUS, GOVERNMENT, GRAMMAR
1830 DATA GRAMOPHONE, GRIEVANCE, GUARD
1840 DATA GUARDIAN, HANDKERCHIEF, HEIGHT
1850 DATA HEROES, HONORARY, HUMOROUS, HUNGRY
1860 DATA HURRIEDLY, HYPOCRISY, IMAGINATION
1870 DATA IMMEDIATELY, IMMIGRATE, INCIDENTALLY
1880 DATA INDEPENDENT, INDISPENSABLE
1890 DATA INFLUENTIAL, INTELLIGENCE
1900 DATA IRRESISTIBLE, KNOWLEDGE, LIGHTENING
1910 DATA LITERATURE, LIVELIHOOD, LOSE
1920 DATA LOSING, LYING, MAINTENANCE
1930 DATA MARRIAGE, MEANT, MEDICINE
1940 DATA MEDITERRANEAN, MINIATURE
1950 DATA MINUTES, MISCHIEVOUS, MURMUR
1960 DATA NECESSARY, NIECE, NOTICEABLE
1970 DATA OCCASIONAL, OCCURRED, OCCURRENCE
1980 DATA OMITTED, OPINION, OPPORTUNITY
1990 DATA ORIGINALLY, PARALLEL, PARLIAMENT

```

2000 DATA FASTIME, PERMANENT, PERMISSIBLE
2010 DATA PERSEVERANCE, PHYSICAL, PLANNING
2020 DATA PLEASANT, POSSESSES, PRECEDING
2030 DATA PREFERENCE, PREJUDICE, PRIVILEGE
2040 DATA PROCEDURE, PROCEEDS, PROFESSIONAL
2050 DATA PROFESSOR, PRONUNCIATION
2060 DATA PROPRIETARY, PSYCHOLOGY, QUIET
2070 DATA REALLY, RECEIVED, RECOGNIZED
2080 DATA RECOMMENDED, REFERRED, RELIEVED
2090 DATA REPETITION, RESTAURANT, RHYTHM
2100 DATA SCARCELY, SECRETARIES, SEIZE
2110 DATA SENTENCE, SEPARATE, SERGEANT
2120 DATA SEVERELY, SHINING, SIEGE, SIMILAR
2130 DATA SINCERELY, SPEECH, STRENGTH
2140 DATA SUCCESSFUL, SUPERSEDE
2150 DATA SUPPRESSION, SURPRISING, SYNONYM
2160 DATA TENDENCY, TRAGEDY, TRANSFERRED
2170 DATA TWELFTH, UNCONSCIOUS
2180 DATA UNNECESSARY, UNTIL, USUALLY
2190 DATA VALUABLE, VIEW, WEDNESDAY
2200 DATA WOOLLEN
2210 DATA END
2220 END

LANDMINE

In Landmine the player must traverse a 10 by 10 field of possible mine locations, from top left to bottom right. The program sets up an almost too easy first two or three attempts but the machine is programmed to place the mines randomly increasing the numbers of mines by 2 each time. Rapidly the player is faced with the near impossible task of negotiating 48, 50 and more mines.

The program does offer the player a little help, in that it issues a warning whenever the player is one and/or two moves away from being blown to smithereens.

In giving the reader (and eventual user), the bare outlines of this addictive game, a lot of scope opens up for the ambitious and inventive home programmer. In this quiet utility program, he can create a more attractive display and with the increased sound capabilities of the 64 develop an exciting range of explosive sounds and warnings.

RUNNING THE PROGRAM

After the usual feat of typing in the program and typing 'RUN', the screen will display the instructions, giving the most important instruction of how to move the player's character:

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

The player takes the number 5 position, and corresponding to the compass, may use the numbers to move in any direction. Hitting any key will renew the game.

PROGRAM STRUCTURE

For an easy view of the positioning of interesting lines in the listing, the following has been correlated for the user's convenience:

70	Array for storing grass and mines.
90 - 130	Sets up grass in all positions of the array.
150 - 190	Chooses random positions for mines.
200 - 210	Sets grass in top left, house in bottom right position.
220 - 410	Displays minefield.
420 - 450	Input move.
460 - 640	Increase or decrease x and y position according to move input.
660 - 760	Check for mines near man.
780 - 810	Displays minefield successfully crossed.
830 - 920	Displays mine stepped on.
930 - 950	Pause for keypress, increase number of mines.
1000 - 1080	Display routine.
1110 - 1230	Checks and gives warning of mines near to man.
1320 - 1530	Instructions.

```
10 REM *****
20 REM *GAMES PROGRAM - SEE IF YOU CAN
30 REM *SUCCESSFULLY CROSS THE MINEFIELD
40 REM *****
50 REM SET UP MINEFIELD IN ARRAYS
55 POKE53281,1
60 NM=10:REM NO MINES = 10 INITIALLY
70 DIM A$(15,16)
80 GOSUB 1310
90 FOR I=1 TO 15
100 FOR J=1 TO 15
110 A$(I,J)=", "
120 NEXT J
130 NEXT I
140 REM SET UP MINES AT RANDOM
150 FOR I=1 TO NM
160 X%=RND(1)*15+1
170 Y%=RND(1)*15+1
180 A$(X%,Y%)="M"
190 NEXT I
200 A$(1,1)=", "
210 A$(15,15)=", "
220 PRINT"XXXXXXXXXXXXXXXXXXXX RAND MINE"
230 PRINT"XXXXXXXXXXXXXXXXXXXX"
240 PRINT"NO OF MINES = ";NM
250 REM PRINT MINEFIELD ON SCREEN
260 CL=15:NO=15:Z$=", ":T$="H"
270 FOR RW=5 TO 19
280 GOSUB 960
290 NEXT RW
300 CL=14:NO=17:RW=4:Z$="*":T$="H"
310 GOSUB 960
320 RW=20
330 GOSUB 960
340 CL=14:NO=17:RW=4:Z$="*":T$="V"
350 GOSUB 960
360 CL=30
370 GOSUB 960
380 X%=1:Y%=1
390 GOTO 660
400 GOSUB 1240
410 PRINT"@ ";
420 GET GG$: IF GG$="" THEN 400
430 GG=VAL(GG$)
440 IF GG<1 OR GG>9 THEN 400
450 ON GGGOTO 460,480,500,520,540,550,570,590,610
460 XXX=X%-1:YY%=Y%+1:CR$=" "
470 GOTO 620
480 XXX=X%:YY%=Y%+1:CR$=" "
490 GOTO 620
500 XXX=X%+1:YY%=Y%+1:CR$=" "

```



```

510 GOTO 620
520 XX%=X%-1:YY%=Y%:CR$="III"
530 GOTO 620
540 GOTO 400
550 XX%=X%+1:YY%=Y%:CR$="III"
560 GOTO 620
570 XX%=X%-1:YY%=Y%-1:CR$="III"
580 GOTO 620
590 XX%=X%:YY%=Y%-1:CR$="I"
600 GOTO 620
610 XX%=X%+1:YY%=Y%-1:CR$="III"
620 IFXX%<1 OR XX%>15 THEN 400
630 IFYY%<1 OR YY%>15 THEN 400
640 X%=XX%:Y%=YY%
650 REM NOW CHECK FOR MINES IN VICINITY
660 IF A$(X%,Y%)="M" THEN 820
670 A$(X%,Y%)="●"
680 IF XX%=15 AND Y%=15 THEN 780
690 REM CHECK FOR 1 MOVE AWAY
700 MS%=1:GOSUB 1090
710 IF MNC>0 THEN 740
720 REM CHECK FOR 2 MOVES AWAY
730 MS%=2:GOSUB 1090
740 REM RESET CURSOR WITHIN MINEFIELD
750 GOSUB 1240
760 GOTO 400
770 REM MINEFIELD CROSSED SUCCESSFULLY
780 CL=1:NO=1:RW=23:T$="H"
790 Z$="MINEFIELD SUCCESSFULLY CROSSED"
800 GOSUB 960
810 GOTO 860
820 REM LANDMINE EXPLODES
830 CL=1:NO=1:RW=23:T$="H"
840 Z$="B A N G !!!!!!!"
845 Z$=Z$+"
850 GOSUB 960
860 A$(X%,Y%)="X"
870 FOR Y=1 TO 15
880 X%=1:Y%=Y:GOSUB 1240
890 FOR X=1 TO 15
900 PRINT A$(X,Y);
910 NEXT X
920 NEXT Y
930 GET GG$:IF GG$="" THEN 930
940 IFGG$<>"E"THENNM=NM+2:GOTO90
950 END
960 REM SUBROUTINE TO DRAW LINE ON SCREEN
970 REM CL=START COLUMN,RW=START ROW
980 REM NO=NO OF CHARS,T$= V(VERT),H(HORIZ)
990 REM Z$=TEXT TO PRINT
1000 PRINT"■"Z$

```

```

1010 FOR I=1 TO CL-1:PRINT"■";NEXT I
1020 FOR I=1 TO RW-1:PRINT"■";NEXT I
1030 IF T$="H" THEN I$=""
1040 IF T$="V" THEN I$="■"
1050 FOR I=1 TO NO
1060 PRINTZ$;I$;
1070 NEXT I
1080 RETURN
1090 REM CHECK FOR MINES IN VICINITY
1100 REM MS%=1 CLOSE SEARCH,MS%=2 WIDE SEARCH
1110 MN=0
1120 FOR I=X%-MS% TO X%+MS%
1130 FOR J=Y%-MS% TO Y%+MS%
1140 IF I<1 OR J<1 THEN 1170
1150 IF I>15 OR J>15 THEN 1170
1160 IF A$(I,J)="M" THEN MN=MN+1
1170 NEXT J
1180 NEXT I
1190 Z$=""
1200 IF MN<=0 THEN 1210
1204 Z$=STR$(MN)+" MINES(S) "
1206 Z$=Z$+STR$(MS%)+ " MOVES AWAY"
1210 CL=1:NO=1:RW=23:T$="H"
1220 GOSUB 960
1230 RETURN
1240 REM CURSOR ADDRESS TO WITHIN MINEFIELD
1250 REM X%=X AXIS, Y%=Y AXIS
1260 PRINT"■";
1270 FOR I=1 TO X%+14:PRINT"■";NEXT I
1280 FOR I=1 TO Y%+4:PRINT"■";NEXT I
1290 PRINT"■";
1300 RETURN
1310 REM INTRODUCTION & INSTRUCTIONS FOLLOW
1320 PRINT"##### A N D M I N E"
1330 PRINT"#####"
1340 PRINT"THE OBJECT OF THIS GAME IS TO"
1350 PRINT"TRAVERSE A MINEFIELD. YOUR MINE"
1360 PRINT"DETECTOR GIVES SOME WARNING"
1370 PRINT"OF NEARBY MINES. BUT CAN'T"
1380 PRINT"GIVE THEIR EXACT LOCATION."
1390 PRINT"SO BE CAREFUL OR :-)"
1400 PRINT"  B A N G !!!  ■"
1410 PRINT"YOU START AT THE TOP LEFT CORNER"
1420 PRINT"AND FINISH (HOPEFULLY) AT THE"
1425 PRINT"BOTTOM RIGHT."
1430 PRINT"USE THE NUMERIC PAD TO MOVE,"
1440 PRINT"WITH YOUR POSITION BEING RELATED "
1450 PRINT"TO THE 5 KEY.■"
1460 PRINT"      7 8 9
1470 PRINT"      \ /

```

```
1480 PRINT"          4-6
1490 PRINT"          / \
1500 PRINT"          1 2 3
1510 PRINT"■PRESS ANY KEY WHEN READY.■"
1520 GET GG$:IF GG$="" THEN 1520
1530 RETURN
```

FRUIT MACHINE

This is a simple fruit machine game. It incorporates nudges, holds, and sound effects. Each symbol is represented by a string of three characters.

RUNNING THE PROGRAM

Just type RUN and follow the instructions.

PROGRAM STRUCTURE

A brief explanation of the program follows.

10-30	Set SID volume and goto instructions.
280	Dimension reel data array.
290-310	Data for reel 1.
320-340	Data for reel 2.
350-370	Data for reel 3.
380-420	Read data into reel array.
440	Set start position.
460	Screen colour blue.
470-810	Set up screen display.
940-2480	Spin the reels and display win if won.
2490-2530	Did not win.
2550-2650	A winner.
2670-2770	End of game.
2800-2850	Sound of reel stopping.
2880-2920	Sound of coins paying out.
2950-3020	Sound when a winner.
3040-3120	Display handle being pulled.
3150-3230	Sound for no win.
3250-3820	Instructions.
3830	End program.

```

10 PRINT"J":CLR
20 NO=0:POKE54296,15
30 GOTO3240
40 REM :
50 REM 'FRUIT MACHINE' PROGRAM
60 REM ADAPTED BY BILL RUSSELL
70 REM DATE: 15TH FEB. 1983
80 REM *** VARIABLES ***
90 REM REEL DATA - DT
100 REM LOOPS FOR REELS - II,JJ
110 REM PLAYER'S MONEY - MY
120 REM POKE LOCATIONS - L1..3
130 REM START OF REELS - P1..3
140 REM REEL CO-ORDS - X,Y
150 REM REEL COLOURS - CL
160 REM HOLD REELS - HD1..3
170 REM RANDOM PLAYING - RN
180 REM RANDOM SPIN - RS
190 REM LOOP - I
200 REM PLAYER'S CHOICE - PL$
210 REM RANDOM NUDGE - RG
220 REM MONEY WON - M
230 REM NUMBER OF TRIES - NO
240 REM LIMIT TRIES - LM,LM$
250 REM START/RESTART PLAY - Q$
260 REM :
270 REM DATA FOR REELS
280 DIMDT(3,20)
290 REM REEL #1
300 DATA28,83,83,65,42,81,65,65,83,65
310 DATA83,81,42,28,83,42,65,81,83,42
320 REM REEL #2
330 DATA65,65,81,28,83,65,42,65,81,83
340 DATA81,65,42,81,83,42,28,65,83,83
350 REM REEL #3
360 DATA42,65,28,65,42,81,65,83,81,65
370 DATA83,65,65,28,81,42,83,81,42,65
380 REM READ IN DATA
390 FOR II=1 TO 3
400 FOR JJ=1 TO 20
410 READ DT(II,JJ)
420 NEXT JJ:NEXT II
430 REM SET START POSITION
440 P(1)=1:P(2)=7:P(3)=15
450 REM SET SCREEN
460 PRINT"J":POKE53280,6:POKE 53281,6
470 PRINTTAB(11)"  FRUIT MACHINE"
480 PRINT"  WIN LINES"
490 PRINT"  - \1.00"
500 PRINT"  X - \0.50"

```

```

510 PRINT "*** - \0.20"
520 PRINT"000 - \0.20"
530 PRINT"000 - \0.20"
540 PRINT"000 - \0.20"
550 PRINT"*X - \0.10"
560 PRINT"00X - \0.10"
570 PRINT"00X - \0.10"
580 PRINT"00X - \0.10"
590 REM PRINT FRUIT MACHINE
600 PRINT"000000"
610 PRINTTAB(14)"000"
620 PRINTTAB(14)"000"
630 PRINTTAB(14)"000 00 00 00 00 00"
640 PRINTTAB(14)"000 00 00 00 00 00"
650 PRINTTAB(14)"000 00 00 00 00 00"
660 PRINTTAB(14)"000 00 00 00 00 00"
670 PRINTTAB(14)"000 00 00 00 00 00"
680 PRINTTAB(14)"000 00 00 00 00 00"
690 PRINTTAB(14)"000 00 00 00 00 00"
700 PRINTTAB(14)"000"
710 PRINTTAB(14)"000"
720 PRINT"000000000000000000"
730 PRINTTAB(14)"000"
740 PRINTTAB(14)"000"
750 PRINT"0000000000"
760 PRINTTAB(36)"000000000000000000000000000000000000"
770 PRINT"000TAB(14)"000 00
780 PRINTTAB(14)"000"
790 PRINT"000MONEY 0000"
800 REM SET UP PLAY ROUTINE
810 PRINT"000":GOTO870
820 REM SET CURSOR POSITION
830 PRINT"000000000000000000000000000000000000"
840 PRINTTAB(14)" "
850 PRINT"000"
860 RETURN
870 MY=50
880 HD(1)=0:HD(2)=0:HD(3)=0
890 GOSUB820
900 PRINTTAB(15)"YOU START WITH 50P"
910 FORZ=1TO1500:NEXTZ:GOSUB940
920 GOSUB1000
930 GOTO1350
940 REM RESET MONEY ROUTINE
950 PRINT"000000000000000000000000000000000000"
960 PRINT" "
970 PRINT"000MONEY "MY
980 PRINT"000TRIES "NO
990 FORZ=1TO500:NEXTZ:RETURN
1000 REM SET UP REELS ROUTINE

```

```

1010 L(1)=1521:L(2)=1527:L(3)=1533
1020 FORX=1TO3
1030 IF HD(X)=1 THEN1090
1040 Y=P(X)
1050 FORJJ=0TO2
1060 POKEL(X)+JJ,(DT(X,Y))+128
1070 POKEL(X)+JJ+54272,1
1080 NEXT JJ
1090 REM SET ABOVE LINE
1100 IFHD(X)=1THEN1170
1110 IFP(X)=1THEN1290
1120 Y=(P(X))-1
1130 FORJJ=0TO2
1140 POKE(L(X)+JJ)-80,(DT(X,Y))+128
1150 POKEL(X)+JJ+54192,1
1160 NEXT JJ
1170 REM SET NEXT ABOVE LINE
1180 IF HD(X)=1THEN1260
1190 IF P(X)=2 THEN1310
1200 IF P(X)=1 THEN1330
1210 Y=(P(X))-2
1220 FORJJ=0TO2
1230 POKE(L(X)+JJ)-160,(DT(X,Y))+128
1240 POKEL(X)+JJ+54072,1
1250 NEXT JJ
1260 GOSUB2780
1270 NEXTX
1280 RETURN
1290 REM Y-1 IS ZERO
1300 Y=20:GOTO1130
1310 REM Y-2 IS ZERO
1320 Y=20:GOTO1220
1330 REM Y-2 IS NEG
1340 Y=19:GOTO1220
1350 REM START PLAYING
1360 GOSUB820
1370 PRINTTAB(15)"PRESS 'S' TO STOP"
1380 FORZ=1TO1500:NEXTZ
1390 RN=INT(RND(0)*9)+1:NO=NO+1
1400 MY=MY-5:M=0:IFMY<0THEN2670
1410 IF NO=LM THEN2660
1420 IF MY=-1 THEN2660
1430 GOSUB940
1440 IF RN=9THEN1800
1450 IF RN=8THEN2070
1460 GOSUB820
1470 PRINTTAB(15)"PRESS 'P' TO PLAY"
1480 GET PL$:IF PL$=""THEN1480
1490 IF PL$="S" THEN2660
1500 IF PL$<>"P" THEN1480
1510 REM STRAIGHT PLAY ROUTINE

```

```

1520 GOSUB3030
1530 HD(1)=0:HD(2)=0:HD(3)=0
1540 RS=INT(RND(0)*4)+4
1550 FORI=1TORS
1560 IF HD(1)=1 THEN1590
1570 P(1)=P(1)-1
1580 IFP(1)=0THEN P(1)=20
1590 IF HD(2)=1 THEN1620
1600 P(2)=P(2)-1
1610 IFP(2)=0THEN P(2)=20
1620 IF HD(3)=1 THEN1650
1630 P(3)=P(3)-1
1640 IFP(3)=0THEN P(3)=20
1650 GOSUB1000
1660 NEXTI
1670 IF HD(2)=1 THEN1700
1680 P(2)=P(2)-1
1690 IFP(2)=0THEN P(2)=20
1700 IF HD(3)=1 THEN1770
1710 P(3)=P(3)-1
1720 IFP(3)=0THEN P(3)=20
1730 P(3)=P(3)-1
1740 IF HD(3)=1 THEN1770
1750 IFP(3)=0THEN P(3)=20
1760 GOSUB1000
1770 FORZ=1T0500:NEXTZ
1780 GOSUB2220
1790 GOT01350
1800 RG=0:REM NUDGE ROUTINE
1810 RG=INT(RND(0)*5)+1
1820 HD(1)=1:HD(2)=1:HD(3)=1
1830 GOSUB820
1840 PRINTTAB(15)"YOU HAVE "RG" NUDGES"
1850 FORZ=1T01500:NEXTZ
1860 GOSUB820
1870 PRINTTAB(15)"USE NUDGE (Y/N)"
1880 GET PL$:IFPL$=""THEN1880
1890 IF PL$="N" THEN1510
1900 IF PL$<>"Y" THEN1880
1910 I=1
1920 GOSUB820
1930 PRINTTAB(15)"NUDGE REEL "I" (Y/N)"
1940 GET PL$:IFPL$=""THEN1940
1950 IF PL$="Y" THEN1990
1960 IF PL$<>"N" THEN1940
1970 I=I+1:IF I=4THEN1910
1980 GOT01920
1990 HD(I)=0
2000 P(I)=P(I)-1
2010 IFP(I)=0THEN P(I)=20
2020 GOSUB1000

```



```

2030 GOSUB2220
2040 IF M>0 THEN RG=1
2050 RG=RG-1:IFRG=0THEN1350
2060 GOTO1830
2070 REM HOLD ROUTINE
2080 GOSUB820
2090 HD(1)=0:HD(2)=0:HD(3)=0
2100 PRINTTAB(15)"YOU HAVE A HOLD"
2110 FORZ=1TO1500:NEXTZ
2120 FORI=1TO3
2130 GOSUB820
2140 PRINTTAB(15)"HOLD REEL "I" (Y/N)"
2150 GET PL$:IF PL$="" THEN2150
2160 IF PL$="Y" THEN HD(I)=1:GOTO2180
2170 IF PL$<>"N" THEN2150
2180 NEXT I
2190 GOTO1540
2200 GOSUB2220
2210 GOTO1350
2220 REM CHECK WIN LINE ROUTINE
2230 HD(1)=0:HD(2)=0:HD(3)=0
2240 REM CHECK \S
2250 IF DT(1,P(1))<>28 THEN2290
2260 IF DT(2,P(2))<>28 THEN2490
2270 IF DT(3,P(3))<>28 THEN M=50:GOTO2540
2280 M=100:GOTO2540
2290 REM CHECK STARS
2300 IF DT(1,P(1))<>42 THEN2340
2310 IF DT(2,P(2))<>42 THEN2490
2320 IF DT(3,P(3))<>42 THEN M=10:GOTO2540
2330 M=20:GOTO2540
2340 REM CHECK SPADES
2350 IF DT(1,P(1))<>65 THEN2390
2360 IF DT(2,P(2))<>65 THEN2490
2370 IF DT(3,P(3))<>65 THEN M=10:GOTO2540
2380 M=20:GOTO2540
2390 REM CHECK BLOBS
2400 IF DT(1,P(1))<>81 THEN2440
2410 IF DT(2,P(2))<>81 THEN2490
2420 IF DT(3,P(3))<>81 THEN M=10:GOTO2540
2430 M=20:GOTO2540
2440 REM CHECK HEARTS
2450 IF DT(1,P(1))<>83 THEN2490
2460 IF DT(2,P(2))<>83 THEN2490
2470 IF DT(3,P(3))<>83 THEN M=10:GOTO2540
2480 M=20:GOTO2540
2490 REM NO WIN
2500 GOSUB820
2510 PRINTTAB(15)"NO WIN !"
2520 GOSUB3130
2530 RETURN

```

```

2540 REM MONEY WIN
2550 GOSUB820
2560 PRINTTAB(15)"YOU WON "M" P !"
2570 GOSUB2930
2580 PRINT"█0000000000000000"
2590 FORJ=1TO(M/10)
2600 PRINTTAB(19+J)"▣";:GOSUB2860
2610 NEXTJ:PRINT"▣"
2620 FORZ=1TO1500:NEXTZ
2630 PRINT"█0000000000000000"TAB(18)"
2640 MY=MY+M:GOSUB940
2650 PRINT"█":RETURN
2660 REM END ROUTINE
2670 PRINT"J"
2680 POKE53280,14
2690 PRINTTAB(15)"▣ GAME OVER ▣"
2700 PRINT"▣ YOU FINISHED WITH \"(MY/100)+.05
2710 PRINT"▣ FROM A START OF \0.50"
2720 PRINT"▣ TAKING A TOTAL OF "NO-1" TURNS."
2730 PRINT"▣ ▣ WILL YOU PLAY AGAIN (Y OR N) ▣"
2740 GET Q$:IFQ$=""THEN2740
2750 IF Q$="Y"THEN10
2760 IF Q$="N"THEN3830
2770 GOTO2740
2780 REM SOUND EFFECT #1
2790 REM BELL WHEN REEL STOPS
2800 POKE54276,65:POKE54277,8:POKE54278,100
2810 POKE54275,50:POKE54274,200
2820 POKE54273,50:POKE54272,100
2830 FORZ=1TO50:NEXTZ
2840 POKE54276,0
2850 RETURN
2860 REM SOUND EFFECT #2
2870 REM COINS DROPPING
2880 POKE54276,129:POKE54277,8:POKE54278,100
2890 POKE54273,1:POKE54272,255
2900 FORZ=1TO50:NEXTZ
2910 POKE54276,0
2920 RETURN
2930 REM SOUND EFFECT #3
2940 REM WINNING CHIME
2950 POKE53280,1
2960 FORK=10TO25
2970 POKE54276,33:POKE54277,8:POKE54278,8
2980 POKE54273,K:POKE54272,250-K
2990 FORZ=1TO30:NEXTZ
3000 POKE54276,0
3010 NEXTK:POKE53280,6
3020 RETURN
3030 REM PULL HANDLE
3040 PRINT"█00000000"

```

```

3050 PRINTTAB(36)" ":PRINTTAB(36)"●"
3060 FORZ=1T050:NEXTZ
3070 PRINT"J"TAB(36)" ":PRINTTAB(36)"●"
3080 FORZ=1T050:NEXTZ
3090 PRINT"J"TAB(36)"●":PRINTTAB(36)"J"
3100 FORZ=1T050:NEXTZ
3110 PRINT"JJ"TAB(36)"●":PRINTTAB(36)"JJ"
3120 RETURN
3130 REM SOUND EFFECT #4
3140 REM NO WIN CHIME
3150 FORK=1T05
3160 POKE54276,17:POKE54277,8:POKE54278,200
3170 POKE54273,75:POKE54272,25
3180 FORZ=1T050:NEXTZ
3190 POKE54273,50:POKE54272,200
3200 FORZ=1T050:NEXTZ
3210 POKE54276,0
3220 NEXTK
3230 RETURN
3240 REM DIRECTIONS OF PLAY
3250 PRINT"000"TAB(8)"  "
3260 PRINTTAB(8)"  "
3270 PRINTTAB(8)"  "
3280 PRINTTAB(8)"  "
3290 PRINTTAB(8)"  "
3300 PRINTTAB(8)"  "
3310 PRINTTAB(8)"  "
3320 PRINTTAB(8)"  "
3330 PRINTTAB(7)"  "
3340 PRINTTAB(7)"  "
3350 PRINTTAB(7)"  "
3360 PRINT"00000"
3370 PRINTTAB(30)"●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●"
3380 PRINT"00000000"TAB(12)"FRUIT MACHINE"
3390 FORZ=1T01500:NEXTZ
3400 PRINT"J"TAB(11)"BY BILL RUSSELL"
3410 FORZ=1T01500:NEXTZ
3420 PRINT"J"TAB(10)"DIRECTIONS (Y/N)?"
3430 GETQ$:IFQ$=""THEN3430
3440 IFQ$="N"THEN3760
3450 IFQ$="Y"THEN3430
3460 PRINT"J"TAB(15)"DIRECTIONS"
3470 PRINT"  YOU START WITH 50P & MAY PLAY ON "
3480 PRINT"EXCEPT IF YOU RUN OUT OF CASH."
3490 PRINT"5P IS DEDUCTED FOR EACH TRY, & THE WIN"
3500 PRINT"LINES ARE DISPLAYED."
3510 PRINT"TO STOP WHEN ASKED TO PLAY 'P', PRESS"
3520 PRINT"THE 'S' KEY."
3530 PRINT"NUDGES AND HOLDS ARE SELF EXPLANATORY."
3540 PRINT"NB: IF YOU DON'T NUDGE THE REELS SPIN."
3550 PRINT"  CHECK VOLUME IS TURNED UP!"

```

```

3560 PRINT"WHEN A REEL SPINS IT GIVES THIS SOUND"
3570 GOSUB3720
3580 GOSUB2780
3590 PRINT"WHEN A COIN DROPS IT GIVES THIS SOUND"
3600 GOSUB3720
3610 GOSUB2860
3620 PRINT"WHEN YOU WIN IT GIVES THIS SOUND"
3630 GOSUB3720
3640 GOSUB2930
3650 POKE53280,14
3660 PRINT"WHEN YOU LOSE IT GIVES THIS SOUND"
3670 GOSUB3720
3680 GOSUB3130
3690 PRINT" PRESS THE 'SPACE BAR' TO START PLAY"
3700 GOSUB3740
3710 GOTO3760
3720 REM PRESS SPACE ROUTINE
3730 PRINT" PRESS SPACE BAR TO CONTINUE PLEASE!"
3740 GETQ$:IFQ$<>" "THEN3740
3750 PRINT" ":RETURN
3760 REM LIMIT ON GAMES
3770 PRINT" DO YOU WANT TO LIMIT NUMBER OF TRIES"
3780 PRINT"TYPE 'N' OR AMOUNT."
3790 INPUT LM$
3800 IF LM$="N"THEN LM=9999:GOTO50
3810 LM=VAL(LM$)+1
3820 GOTO50
3830 POKE54296,0:END

```

CAR

Car is a racing game on a track the shape of which looks similar to a rifle, thus incorporating the required bends to make the game interesting. Car differs from the usual racing game because it requires as near as possible the actual driving skills one would learn in a 'real car'. The player must use his judgement when accelerating and changing through the gears. The driving novice is almost guaranteed to stall or over rev the engine causing a blow out.

RUNNING THE PROGRAM

Load, type 'RUN' and follow the instructions.

The user will be given a line of information on each bend as the vehicle approaches. Directly below this on the screen, the gear number, rev count and speed will appear and the player uses this information to guide him round the course.

PROGRAM STRUCTURE

The lines of interest to help the user are as follows:

80 - 230	Set up arrays for track, gear ratios etc.
240 - 280	Display keypress choice.
290 - 380	Set up initial position of car on track.
390 - 460	Input and check for valid key.
480 - 530	Check speed, revs etc.
535 - 550	Over-revved.
560 - 565	Stalled.
570 - 580	Braked to a standstill.
590	Blow up the engine.
600 - 640	Warning to check what gear you are in.
645	Crashed.
650 - 670	Skidded.
680 - 720	Lap completed.

```

10 REM *****
20 REM *GAME TO DRIVE A CAR AROUND *
25 REM *A RACE TRACK *
30 REM *USE THE NUMBER KEYS TO *
35 REM *CONTROL SPEED *
40 REM *BRAKING AND GEAR CHANGES. *
50 REM *TRY NOT TO TAKE THE BENDS *
55 REM *TOO FAST YOU *
60 REM *MAY SKID OFF THE TRACK *
65 REM *BEST TIME WINS. *
70 REM *****
75 POKE53280,2:POKE53281,12:CO=54272
80 DIM A(14),B(14),C(14),GS(5),BS(10),MS(14)
90 X=1115:G=1:R=0:S=0
100 PRINT"Q"
110 FOR Z=1TO14
115 READ A,B,C:A(1(Z))=A:B(1(Z))=B:C(1(Z))=C
120 FORD=1TOA:POKEX,42:POKEX+CO,1:X=X+B:NEXTD
130 X=X+C
140 NEXT Z
150 DATA 20,1,40,3,41,-40,4,1,40,2,41,-1
155 DATA 2,40,-1,6,-1,-40,3,-41,40
160 DATA 2,39,1,2,39,-40,4,-1,-40,5,-41,40
165 DATA 12,-1,-40,2,-41,2,6,1,1
170 FOR Z=1TO5:READ GS(Z):NEXT
180 DATA 2,4,1,9,1,5,1,2,1
190 FOR Z=0TO9:READ BS(Z):NEXT
200 DATA 0,6,15,20,25,30,40,59,80,98
210 FOR Z=1TO14:READ MS(Z):NEXT
220 DATA 70,70,70,85,70,70,35,40
225 DATA 60,60,70,80,40,200
230 I=10:GOSUB 870
240 PRINT"7 TO CHANGE UP
250 PRINT"4 TO CHANGE DOWN
260 PRINT"8 TO BRAKE
270 PRINT"9 TO ACCEL
280 PRINT"5 TO START
290 GET S$:IF S$="" THEN 290
300 X=1115:Y=X:S9=TI
310 A=3:GOSUB 840
320 FOR Z=1 TO 14
330 A2=A1(Z):B2=B1(Z):C2=C1(Z):MX=MS(Z)
340 I=15:GOSUB 870
345 IF Z<14 THEN PRINT"MAX SPEED AT BEND IS ";MX
350 FOR DL=1TO A2
360 I=16:GOSUB890
365 PRINT"GEAR=";G;"REVS=";INT(R);
366 PRINT"MPH=";INT(S);" "
370 FORZ2=1TO500:NEXT
380 POKE X,145:POKE Y,42:Y=X:X=X+B2
390 GET N

```

```

400 IF N=0 AND NN=0 THEN 480
410 IF N=0 AND NN=1 THEN NN=0:GOTO 470
420 IF N=7 THEN G=G+1:IF G>5 THEN G=4:GOTO 600
430 IF N=4 THEN G=G-1:IF G<1 THEN G=1:GOTO 600
440 IF N=8 OR N=5 OR N=2 THEN GOSUB 750
450 IF N=9 THEN GOSUB 810
460 NN=1:GOTO 390
470 GOSUB 840
480 IF S<2 THEN 570
490 IF R<2000 THEN 560
500 IF R=<12000 THEN 540
510 IF R>14000 THEN 590
520 OV=OV+1:IF OV>2 THEN 590
530 D=17:GOSUB 870
535 PRINT"⚠WARNING⚠ YOU HAVE OVERREVVED":S=S-S/4
540 NL=1:IF S<200 THEN NL=INT((200-S)/8)
550 GOSUB 730:D=18:GOSUB 870:GOTO 610
560 D=17:GOSUB 870:PRINT"⚠YOU HAVE STALLED⚠"
565 GOTO580
570 D=17:GOSUB 870
575 PRINT"⚠YOU HAVE BRAKED TO A STANDSTILL⚠"
580 B=0:G=1:S=0:A=3:GOSUB 840:GOTO540
590 D=17:GOSUB 870:PRINT"⚠YOU HAVE BLOWN UP":END
600 D=17:GOSUB 870
605 PRINT"⚠CHECK YOUR GEARS":GOTO 540
610 NEXT DL
620 X=X+C2
630 IF S<MX OR Z=14 THEN 670
640 IF S>(MX+MX/5) THEN D=17:GOSUB 870
645 PRINT"⚠YOU HAVE CRASHED⚠":END
650 D=17:GOSUB 870
655 PRINT"⚠YOU HAVE SKIDDED & LOST SPEED⚠"
660 S=S-S/3
670 NEXT Z
680 TS=(TI-S9)/60:TM=INT(TS/60)
690 TS=TS-(TM*60)
700 D=17:GOSUB 870:
710 PRINT"⚠YOUR LAP TIME WAS";TM;"MINS";TS;"SECS"
720 END
730 FOR L=0TONL:L1=SQR(NL):NEXT
740 RETURN
750 IF N=8 THEN VV=3
760 IF N=5 THEN VV=2
770 IF N=2 THEN VV=1
780 B=B+VV:A=0
790 IF B>9 THEN B=9
800 RETURN
810 A=A+1:IF A>9 THEN A=9
820 B=0
830 RETURN

```

```
840 S=S+(A*3.2*GS(G))-BS(B)-S/20
850 R=S*GS(G)*61
860 RETURN
870 PRINT"⌘"
880 IF D>16 THEN D=D-1:ML=1:RETURN
890 PRINT"⌘":FOR D1=1 TO D
900 PRINT
910 NEXT
920 IF ML=0 THEN 940
930 ML=0
935 PRINT"
940 RETURN
950 END
READY.
```

⌘"

MAZE

Maze is a highly amusing and intricate journey through randomly created mazes in three different levels of complexity. The player is offered a choice of maze level and asked whether he wishes to see the maze being created. If the player opts for a maze without seeing its creation then he must be prepared to wait for a moment, until, with a flourish, the concocted maze appears.

This game makes use of the four function keys on the right hand side of the 64. To manipulate the cursor through the maze, the keys' values are as follows:

f1 = up
f3 = left
f5 = right
f7 = down

The player is timed against a seconds clock in the top left hand corner of the screen. A fast time for one of the small mazes is 26 seconds.

RUNNING THE PROGRAM

After loading and typing 'RUN' the program asks the player to press 'G' and then asks the required size of maze, small, medium or large. Having chosen the level, the program then asks the player if he wishes to view the maze as it's being formed. The program is dependent on the answer at this point and will display a grid of squares before it proceeds visibly or invisibly to dismember the grid into the maze. Then it is a race against time.

PROGRAM STRUCTURE

The more interesting lines are as follows:

10 - 464	Introduction.
465	Wait for key G
560	Dimension maze arrays.
620 - 680	Determine size of maze.
700 - 720	Set up blank maze.
800 - 940	Print up blank maze.
960 - 1600	Create maze and start clock.
1620 - 2000	Get move and if legal do.
2020 - 2080	Exit has been found, end game.
2100 - 2300	Display maze.

```

10 REM * CLEAR SCREEN AND SET BACKGROUND AND BORDER COLOURS
100 PRINT"■□":POKE53280,6:POKE53281,7
120 POKE53272,21
140 PRINT"XXXXXXXXXXXX"
160 PRINT"*****
180 PRINT"      **  MAZE  **
200 PRINT"      *****
220 PRINT
240 PRINT
260 PRINT
280 FORN=1TO2500:NEXT:REM DELAY LOOP
300 PRINT"□"
320 PRINT:PRINT
340 PRINT"      MAZE - CAN YOU ESCAPE?■"
360 PRINT:PRINT:PRINT
380 PRINT"YOU HAVE TO ESCAPE FROM THE MAZE IN THE
HE
400 PRINT"SHORTEST TIME POSSIBLE
420 PRINT
440 PRINT"THE CONTROLS ARE THE KEYS F1,F7,F5 &
F3 FOR
460 PRINT"UP,DOWN,RIGHT AND LEFT RESPECTIVELY.
461 PRINT:PRINT:PRINT
462 PRINTSPC(15)"GOOD LUCK!■"
463 PRINT:PRINT
464 PRINTSPC(12)"KEY G TO CONTINUE■"
465 GETG$:IFG$=""THEN465:REM WAIT TILL G IS PRESSED TO CONT.
466 IFG$<>"G"THEN465
467 PRINT"□"
560 DIMW(38,22),V(38,22),D(8):REM * ARRAYS FOR MAZE
580 X=RND(-TI)
600 H=10:V=11
620 PRINT"DO YOU WANT A SMALL,MEDIUM, OR BIG MAZE (S/M/B)":INPUTA$
621 PRINT:PRINT:PRINT
640 X=ASC(A$):IFX=83GOTO 700
660 H=15:V=11:IFX=77GOTO 700
680 H=19:V=11
700 X=200:FORI=1TOH:W(I,0)=X:NEXTI
720 FORI=1TOV:W(0,I)=X:W(H+1,I)=X:NEXTI
740 PRINT"DO YOU WANT TO WATCH ME CONSTRUCT IT (Y/N)":INPUTA$
760 F=0:IFASC(A$)<>89THENF=1
780 Z=INT(RND(1)*H+1)
800 PRINT"□":PRINT" ";:FORI=1TOH:REM PRINTS MAZE
E
820 :A$="__":IFI=ZTHENA$="  "
840 :PRINTA$:NEXTI

```

```

860 PRINT
880 :FORJ=1TOV:PRINT" I";
900 : FORI=1TOH:PRINT" I";NEXTI
920 :PRINT:PRINT" I";
940 FORI=1TOH:PRINT" _I";NEXTI:PRINT:NEXTJ
960 W(Z,1)=1:C=2:R=Z:S=1:GOTO 1060
980 IFR<>HTHENR=R+1:GOTO 1040
1000 R=1:IFR<>VTHENS=S+1:GOTO 1040
1020 S=1:GOTO 1040
1040 IFW(R,S)=0GOTO 980
1060 FORJ=1TO4:D(J)=0:NEXTJ
1080 D(1)=W(R-1,S)=0
1100 D(2)=W(R,S-1)=0
1120 D(3)=W(R+1,S)=0
1140 D(4)=W(R,S+1)=0AND(C>V*H/2ORS<V)
1160 D=0:FORJ=1TO4:D=D+D(J):NEXTJ
1180 IFD=0GOTO 980
1200 X=RND(1)*D
1220 FORD=1TO4:X=X-D(D):IFX<0THENNEXTD
1240 ONDGO 1280, 1320, 1460, 1360
1260 STOP
1280 R=R-1:V(R,S)=2
1300 GOTO1340
1320 S=S-1:V(R,S)=1
1340 GOSUB2160:GOTO 1500
1360 IFS<VGO 1400
1380 X=200:FORJ=1TOH:W(J,V+1)=X:NEXTJ
1400 V(R,S)=V(R,S)OR1
1420 GOSUB2160:S=S+1:IFR<>VANDC<H*V+1THENC=C+1:G
OTO 1020
1440 GOTO 1500
1460 V(R,S)=V(R,S)OR2
1480 GOSUB2160:R=R+1
1500 W(R,S)=C:C=C+1
1520 IFC<H*V+2GOTO 1060
1540 IFF<>1GOTO1580
1560 PRINT"###";FORR=1TOV:PRINT"###";FORR=1TOH:
GOSUB2220:NEXTR:PRINT:PRINT:NEXTS
1580 R=Z:S=1:GETA#:GETA#:GETA#
1600 TI$="000000":REM SET CLOCK
1620 R1=R:S1=S:A$=" ":GOSUB2100
1640 PRINT"###";TI$
1680 D=PEEK(197):IFD=64 THEN 1640:REM GET KEY P
RESS AND DIRECTION
1700 IFD=3GOTO 1840
1720 IFD=5GOTO 1880
1740 IFD=6GOTO 1940
1760 IFD<>4GOTO 1640
1780 IFS=1GOTO 1640
1800 IFABS(V(R,S-1)-1)=1GOTO 1640
1820 S=S-1:GOTO 1980

```

```

1840 IFABS(V(R,S)-1)=1GOTO 1640
1860 S=S+1:GOTO 1980
1880 IFR=1GOTO 1640
1900 IFV(R-1,S)<2GOTO 1640
1920 R=R-1:GOTO 1980
1940 IFV(R,S)<2GOTO 1640
1960 R=R+1
1980 A$=" ":GOSUB2100
2000 IFS<=VGOTO 1620
2020 PRINT"#####CONGRATULATIONS!":REM EXIT
ROUTINE
2040 FORJ=1TO1000:NEXT
2060 GETZ$:IFZ$<>" "GOTO2060
2080 PRINT"#####":END
2100 PRINT"$"
2120 FORJ=1TOS1*2-1:PRINT:NEXT
2140 FORJ=1TOR1:PRINT"||";NEXTJ:PRINT"||";A$:RE
TURN
2160 IFF=1THENRETURN
2180 PRINT"$":FORJ=1TOS*2-1:PRINT:NEXTJ
2200 FORJ=1TOR:PRINT"||";NEXTJ:PRINT"||";
2220 A$=" ":X=V(R,S):IFX<2THENA$="|"
2240 PRINTA$;"|||";A$=" ":IFX=2THENA$="__"
2260 IFX=0THENA$="_|"
2280 IFX=1THENA$=" |"
2300 PRINTA$;"|":RETURN

```

READY.

KEYBOARD SYNTHESIZER

This program allows the user to play tunes on the 64 by using the 64's keyboard for the notes. Full use is made of the envelopes and filters and very interesting effects can be heard by playing the same note through more than one voice.

As long as the key is pressed down, the note will play, and when the key is released, the note will stop. It would have been nice to be able to play different notes for each voice but because of the keyboard scanning routine in the 64, only one key can be detected at any one time. Therefore, there is the option, by using keys 1-7, to choose any of the seven combinations of the three voices with the same frequency.

RUNNING THE PROGRAM

Turn up the volume of your 64 and type RUN. A page of instructions will display on the screen and will stay there while the user plays the notes. Pressing RETURN will allow the user to make changes to each or any of the three voices including setting a voice to go through the programmable filter (which can also be changed). This program was written on an early version of the 64 and since that time, Commodore have made changes to the SID chip. We have not had enough time to investigate the changes made, but have noticed that the operation of the filters is different.

The program has been written so that there is not a lot of lost speed. Therefore, the use of the keys 1-7 must be explained:

1. Voice 1 only,
2. Voice 2 only,
3. Voice 1 and 2 only,
4. Voice 3 only,
5. Voice 1 and 3 only,
6. Voice 2 and 3 only,
7. All three voices.

The key 'O' has also been given an operation: This key will turn off all three voices. The use of the frequency table rather than testing for each key prevent lots of IF ... THEN statements which are very slow.

PROGRAM STRUCTURE

- 10 Dimension frequency table array.
- 20-50 Set up frequency table.
- 60-130 Frequency table data.
- 140 S is the start of the SID chip.
- 150 V:initial voice; LK>Last key (no key); LS>Last shift (no shift).
- 160-250 Read in and set up envelopes, waveforms, etc.
- 260-280 Data for above.
- 290 Read initial filter settings.
- 300-330 Set filters.
- 340 Filter data.
- 350 Display screen.
- 360 Start playing.
- 1000 Get key press.
- 1010 Check to see if it is the same note as before.
- 1020-1045 Decide on note or other key.
- 1050-1130 Is note, play it.
- 2000-2050 Turn off notes in all three voices.
- 3000-3040 Decide which voices to play on.
- 4000-4580 Make changes to the voices (after pressing RETURN).
- 5000-5150 Display instruction page.

```

5 REM * ALLOWS USER TO SET AND USE SYNTH.
10 DIM N(64):REM READS IN KEY PRESS ARRAY
20 FOR I=0 TO 64
30 READ A
40 N(I)=A
50 NEXT I
60 DATA ,-1,,,,,,
70 DATA 65539,9854,4389,65540,2195,4927
80 DATA 11060,,65541,11718,5530,65542,2765,5859
90 DATA 13153,2463,65543,14764,6577,,3288,7382
100 DATA 16572,2930,,17557,8286,65536,4143,8779
110 DATA ,3691,,,,,,,,,4389,,,
120 DATA ,,,,,,65537,,,,65538,
130 DATA ,8779,,
140 S=54272:REM * SET VARIABLE TO START OF SID CHIP
150 V=1:LK=64:LS=0
160 FOR I=0 TO 2:REM * READS WAVEFORM, ATTACK, SUSTAIN, DECAY
170 READ W(I),A(I),D(I),SU(I),R(I),PW(I),W#(I),SY(I),RM(I),FI(I)
180 POKES+I*7+4,W(I)*16+RM(I)*4+SY(I)*2
185 REM * SETS CONTROL REGISTER FOR EACH VOICE
190 POKE S+I*7+5,A(I)*16+D(I)
200 POKE S+I*7+6,SU(I)*16+R(I)
210 P1=INT(PW(I)/256)
220 P2=PW(I)-P1*256
230 POKE S+I*7+2,P2
240 POKE S+I*7+3,P1
250 NEXT I
260 DATA 4,0,9,0,0,2048,"PULSE",0,0,0
270 DATA 1,4,2,10,5,0,"TRIANGLE",0,0,0
280 DATA 2,6,5,2,2,0,"SAWTOOTH",0,0,1
290 READ FF,FR,FB,FB#:REM READ FILTER SETTINGS
300 POKES+23,FR*16+FI(0)+FI(1)*2+FI(2)*4
310 POKES+21,FFAND7
320 POKES+22,INT((FF-(FFAND7))/8

```

```

)
330 POKES+24,FB*16+15
340 DATA 1024,12,4,"HI"
350 GOSUB 5000
360 GOTO 3010
999 REM K=KEY PRESS AND PS=SHIFT
REGISTER
1000 K=PEEK(197):PS=PEEK(653)
1010 IF K=LK AND PS=LS THEN 1000

1020 F=N(K):LK=K:LS=PS
1030 IF F=0 THEN 2000
1040 IF F>65535 THEN 3000
1045 IF F=-1 THEN 4000
1050 IF PS=1 THEN F=INT(F*2↑(1/1
2))
1060 IF PS=2 THEN F=INT(F/2↑(1/1
2))
1070 F1=INT(F/256)
1080 F2=F-F1*256
1085 FOR I=0 TO 2
1086 IF VO(I)=0 THEN 1125
1090 POKE S+I*7+4,0
1100 POKE S+I*7+4,W(I)*16+RM(I)*
4+SY(I)*2+1
1110 POKE S+I*7,F2
1120 POKE S+I*7+1,F1
1125 NEXT I
1130 GOTO 1000
2000 FOR I=0 TO 2
2010 POKE S+I*7,0
2020 POKE S+I*7+1,0
2030 POKE S+I*7+4,W(I)*16
2040 NEXT I
2050 GOTO 1000
3000 V=F-65536
3010 FOR I=0 TO 2
3020 VO(I)=(VAND2↑I)/2↑I
3030 NEXT I
3040 GOTO 1000
4000 POKE 53280,14:POKE 53281,6
4005 PRINT"□";
4010 FOR I=0 TO 2
4020 PRINT"☐";TAB(10*I+10);"☐=VO
ICE";I+1
4030 PRINT"☐HAVEFORM☐";TAB(10*I
+10);W(I)
4040 PRINT"☐ATTACK ☐";TAB(10*I
+10);A(I)

```



```

4050 PRINT"  DECRY  ";TAB(10*I
+10);D(I)
4060 PRINT"  SUSTAIN  ";TAB(10*I
+10);SU(I)
4070 PRINT"  RELEASE  ";TAB(10*I
+10);R(I)
4080 PRINT"  P. WIDTH";TAB(10*I
+10);PW(I)
4081 PRINT"  SYNC  ";TAB(10*I
+10);SY(I)
4082 PRINT"  RING MOD";TAB(10*I
+10);RM(I)
4083 PRINT"  FILTER  ";TAB(10*I
+10);FI(I)
4090 NEXT I
4091 PRINT"  FILTER FREQ.  ";FF
4092 PRINT"  FILTER RES.  ";FR
4093 PRINT"  FILTER BAND  ";FB$

4100 POKE 198,0:REM CLEAR KEY BO
ARD BUFFER
4110 PRINT"  DO YOU WISH TO CHA
NGE THE VALUES ? ";
4120 GET A$:IF A$<>"N" AND A$<>"
Y" THEN 4120
4130 IF A$="N" THEN GOSUB 5000:G
OTO 1000
4140 PRINT"YES"
4150 PRINT"  WHICH VOICE DO YOU W
ISH TO CHANGE ? ";
4160 GET A$:IF A$="" THEN 4160
4170 IF ASC(A$)<49 OR ASC(A$)>51
THEN 4160:REM CHECK WITHIN BOUNDS

4180 PRINTA$:VV=VAL(A$)
4190 PRINT "  WAVEFORM  ? ";
4200 GETA$:IFA$<>"S"ANDA$<>"N"AN
DA$<>"P"ANDA$<>"T"THEN4200
4210 IF A$="N" THEN PRINT"NOISE"
:W$(VV-1)="NOISE":W(VV-1)=8
4220 IF A$="P" THEN PRINT"PULSE"
:W$(VV-1)="PULSE":W(VV-1)=4
4230 IF A$="S" THEN PRINT"SAWTOO
TH":W$(VV-1)="SAWTOOTH":W(VV-1)=2
4240 IF A$="T" THEN PRINT"TRIANG
LE":W$(VV-1)="TRIANGLE":W(VV-1)=1
4250 INPUT"  ATTACK  "
":A(VV-1)
4260 IF A(VV-1)<0 OR A(VV-1)>15
THEN PRINT"  "):GOTO 4250
4270 INPUT"  DECRY  "

```

```

";D(VV-1)
  4280 IF D(VV-1)<0 OR D(VV-1)>15
THEN PRINT"J";:GOTO 4270
  4290 INPUT"§SUSTAIN          ||
";SU(VV-1)
  4300 IF SU(VV-1)<0 OR SU(VV-1)>1
5 THEN PRINT"J";:GOTO 4290
  4310 INPUT"§RELEASE          ||
";R(VV-1)
  4320 IF R(VV-1)<0 OR R(VV-1)>15
THEN PRINT"J";:GOTO 4310
  4330 INPUT"§PULSE WIDTH
";PW(VV-1)
  4340 IF PW(VV-1)<0 OR PW(VV-1)>4
095 THEN PRINT"J";:GOTO 4330
  4341 INPUT"§SYNC            ||
";SY(VV-1)
  4342 IF SY(VV-1)<0 OR SY(VV-1)>1
THEN PRINT"J";:GOTO 4341
  4344 INPUT"§RING MOD        ||
";RM(VV-1)
  4345 IF RM(VV-1)<0 OR RM(VV-1)>1
THEN PRINT"J";:GOTO 4344
  4346 INPUT"§FILTER          ||
";FI(VV-1)
  4347 IF FI(VV-1)<0 OR FI(VV-1)>1
THEN PRINT"J";:GOTO 4346
  4350 POKE S+(VV-1)*7+4,W(VV-1)*1
6+RM(VV-1)*4+SY(VV-1)*2
  4360 POKE S+(VV-1)*7+5,A(VV-1)*1
6+D(VV-1)
  4370 POKE S+(VV-1)*7+6,SU(VV-1)*
16+R(VV-1)
  4380 P1=INT(PW(VV-1)/256)
  4390 P2=PW(VV-1)-P1*256
  4400 POKE S+(VV-1)*7+2,P2
  4410 POKE S+(VV-1)*7+3,P1
  4420 T=TI
  4430 IF TI-T<50 THEN 4430
  4450 INPUT"FILTER FREQUENCY
";FF
  4460 IF FF<0 OR FF>2047 THEN PRI
NT"J";:GOTO 4450
  4470 INPUT"FILTER RES
";FR
  4480 IF FR<0 OR FR>15 THEN PRINT
"J";:GOTO 4470
  4490 INPUT"FILTER BAND(HI,LO,OR
BAND)";FB$
  4500 IF FB$="HI" THEN FB=4:GOTO4
540

```

```

4510 IF FB#="LO" THEN FB=1:GOTO4
540
4520 IF FB#="BAND" THEN FB=2:GOT
04540
4530 PRINT"7";:GOTO 4490
4540 POKES+24,FB*16+15
4550 POKES+23,FR*16+FI(0)+FI(1)*
2+FI(2)*4
4560 POKES+21,FFAND7
4570 POKES+22,INT((FF-(FFAND7))/
8)
4580 GOTO 4000
5000 POKE 53280,4:POKE 53281,2:R
EM BACKGROUND AND BORDER COLOURS
5005 PRINT"***** 64
KEYBOARD *****"
5010 PRINT" USE THE KEYS Q W E
R T Y U I"
5020 PRINT"X S D
F G H J K"
5030 PRINT"Z X C
V B N M ,"
5040 PRINT"FOR THE THREE AVAI
LABLE OCTAVES."
5050 PRINT"USE THE SHIFT KEY
FOR A SHARP,"
5060 PRINT"CBM KEY
FOR A FLAT."
5070 PRINT"USING KEYS 1 2 3
4 5 6 7 YOU MAY"
5080 PRINT"CHOOSE ANY COMBINATIO
N OF THE THREE"
5090 PRINT"VOICES. THEY ARE SET
UP BY USING BINARY"
5100 PRINT"ARITHMETIC. THEREFORE
, VOICE 1 IS TURNED";
5110 PRINT"ON USING KEY 1, VOICE
1 AND 3 ARE TURNED";
5120 PRINT"ON USING KEY 5,ETC"
5130 PRINT"USE THE RETURN KEY
TO CHANGE THE VALUES"
5140 PRINT"IN THE VOICES"
5150 RETURN

```

READY.

RESERVED WORD CHANGER

This program shows how the user can change the reserved words (command names) of the 64. Because of the size of the 64's memory capacity, there is memory behind the ROMs. This can be used for things like Hi-Res screens (see section on Hi-Res graphics) or for other data storage. The problem is that with the Basic ROM switched in, when PEEKing a location, the value in the ROM will be returned but when POKEing to that location, the value is stored in the RAM behind. This RAM could be used for making changes to the Basic ROM by copying it first into the RAM behind and making changes to it.

We have produced a little routine that will copy the ROM, and then allow the user to make changes to the Basic commands. Each command name is stored in a look-up table and is terminated by having the last character plus 128.

RUNNING THE PROGRAM

The user may change a command name by just entering the Basic command and when it has been found, changing it to another command of the same length, i.e. if you preferred WRITE to PRINT, enter PRINT and when it has been found, type WRITE. On exiting the program, listing it will show every occurrence of PRINT changed to WRITE.

PROGRAM STRUCTURE

20	Copy ROM into RAM.
80	Input command to be changed.
100	Convert command entered to be in the same formats as in the table.
120	Search for command.
130	If not found.
140	Input new command name.
160	Check for length.
180	As 100.
200-220	Change command name.
230	Ask for new command.
240-270	If no, go to ending routine.

290-500 Scan for first command entered.
1000-1070 Ending routine. Save new Basic or end. Switch Basic
ROM back in by POKE1,55.

```

10 REM LOOP TO COPY ROM INTO RAM, BEHIND ROM
20 FORA=40960TO49151
30 POKEA,PEEK(A)
40 NEXT
70 REM PUT RESERVED WORD INTO R$
80 INPUT"RESERVED WORD";R$
90 REM SET TERMINATOR MARKER ON LAST BYTE OF
STRING
100 R$=LEFT$(R$,LEN(R$)-1)+CHR$(ASC(RIGHT$(R$,
1)))+128)
110 REM ROUTINE TO SEARCH ROM FOR RESEVED WOR
D
120 GOSUB290
130 IFF=0THENPRINT"NOT FOUND":GOTO23
0
140 INPUT"YOUR WORD (SAME LENGTH)";N$
150 REM CHECK LENGTH OF WORDS ARE THE SAME
160 IFLEN(N$)<>LEN(R$)THEN140
170 REM ADD TERMINATOR
180 N$=LEFT$(N$,LEN(N$)-1)+CHR$(ASC(RIGHT$(N$,
1)))+128)
190 REM LOOP TO POKE IN NEW WORD
200 FORJ=1TOLEN(N$)
210 POKEAD+J-1,ASC(MID$(N$,J,1))
220 NEXT
230 PRINT"ANOTHER WORD (Y/N)"
240 GETA$:IFA$<>"Y"ANDA$<>"N"THEN240
250 IFA$="N"THEN1000
260 REM ANOTHER WORD
270 GOTO80
280 REM START ADDRES OF ROM
290 AD=40960
300 REM GET FIRST CHARACTER
310 C=ASC(MID$(R$,1,1))
320 REM IF FIRST CHAR. CHECK OTHERS
330 IFPEEK(AD)=CTHEN420
340 REM LOOK AT NEXT ROM POSITION
350 AD=AD+1
360 REM CHECK FOR END OF WORD TABLE IN ROM
370 IFAD=>42000THENF=0:RETURN
390 REM STARTS NEXT CHECK
400 GOTO310
410 REM SET POINTER TO POSITION OF SECOND CHA
R.
420 AD=AD+1
430 REM LOOP CHECK REST OF CHAR. MATCH
440 FORJ=2TOLEN(R$)
450 REM CHECK EACH CHAR.
460 IFPEEK(AD+J-2)<>ASC(MID$(R$,J,1))THEN310
470 NEXT
480 REM SET POINTER TO FIRST START OF WORD AN

```

```

D: SET FOUND FLAG
490 AD=AD-1:F=-1
500 RETURN
1000 PRINT"***** DO YOU WISH TO SAVE"
1010 PRINT"***** VERSION OF BASIC (Y/N)"
1020 P=PEEK(197):IFP<>25ANDP<>39THEN1020
1030 IFP=39THENPRINT"Y:POKE1,54:END:REM TAKE
OUT BASIC ROM
1040 POKE665,PEEK(45):POKE666,PEEK(46)
1050 POKE45,0:POKE46,192:POKE43,0:POKE44,160
1060 POKE1,54:SAVE"@@:BASICS",8:REM 8 FOR DISK A
ND 1 FOR TAPE
1065 REM FOR TAPE * SAVE"BASIC"
1070 POKE43,1:POKE44,8:POKE45,PEEK(665):POKE46
,PEEK(666)

```

+/- LISTER

This is a short program that uses the '+' and '-' keys to list lines of a program on the screen. This program should be put at the end of a program and when run, it scans through the Basic program and lists just one line at a time either forward or backward through the program.

RUNNING THE PROGRAM

When RUN, this routine will display the first line in the program. Pressing the '+' key, it will scan through the program and display the next line. Pressing '-', the previous line will be listed. Press the STOP key to end the list.

PROGRAM STRUCTURE

- 20 Start address of the Basic program.
- 30 Get the first line number.
- 40 Display in background the commands GOTO 60 and LIST line.
- 50 Set keyboard buffer to operate on these commands and end.
- 60-80 Input from the keyboard.
- 90-100 Find start of next line.
- 110-130 Find start of previous line.


```

10 REM * +/- LIST FOR THE 64
20 ADDRESS = 2048
30 LINE = PEEK(ADDRESS+3)+PEEK(ADDRESS+4)*256
40 PRINT":GOTO60":PRINT"LIST";LINE;
50 POKE 631,19:POKE 632,17:POKE 633,159:POKE 6
34,13:POKE 635,19:POKE 636,13:POKE 198,6:END
60 IF PEEK(197) = 40 THEN 90
70 IF PEEK(197) = 43 THEN 110
80 GOTO 60
90 IF PEEK(ADDRESS+5)<>0 THEN ADDRESS = ADDRESS
S+1:GOTO90
100 ADDRESS = ADDRESS +5:GOTO30
110 ADDRESS = ADDRESS-1
120 IF PEEK(ADDRESS)=0 AND PEEK(ADDRESS-4)<>0
AND PEEK(ADDRESS-3)<>0 THEN 30
130 GOTO 110

```

READY.

LARGE CHARACTERS

This program gives a set of large characters using some of the 64's graphics characters. The characters are 7 x 5 normal characters large and have been given a 3-dimensional effect.

RUNNING THE PROGRAM

When RUN, the program will ask the user to enter a letter from A to O. When one of these keys has been pressed, the character for that key will be displayed.

PROGRAM STRUCTURE

1 -4	Display	intro.
10 -90	Character	L.
100 -180	Character	I.
190 -270	Character	M.
280 -360	Character	A.
370 -450	Character	B.
460 -540	Character	C.
550 -630	Character	D.
640 -720	Character	E.
730 -810	Character	F.
820 -900	Character	G.
910 -990	Character	H.
1000 -1080	Character	J.
1090 -1165	Character	K.
1200 -1280	Character	N.
1290 -1370	Character	O.
12000-12020	Get a character and test it.	
12040	Print the character.	
12050	Repeat.	


```

450 RETURN
460 PRINTTAB(N)"  3  /  "
470 PRINTTAB(N)"  3  3  "
480 PRINTTAB(N)"  3  3  "
490 PRINTTAB(N)"  3  3  "
500 PRINTTAB(N)"  3  3  "
510 PRINTTAB(N)"  3  3  "
520 PRINTTAB(N)"  3  3  "
530 PRINTTAB(N)"  3  3  "
540 RETURN
550 PRINTTAB(N)"  3  3  /  "
560 PRINTTAB(N)"  3  3  3  "
570 PRINTTAB(N)"  3  3  3  3  "
580 PRINTTAB(N)"  3  3  3  3  "
590 PRINTTAB(N)"  3  3  3  3  "
600 PRINTTAB(N)"  3  3  3  3  "
610 PRINTTAB(N)"  3  3  3  3  "
620 PRINTTAB(N)"  3  3  3  3  "
630 RETURN
640 PRINTTAB(N)"  3  3  /  "
650 PRINTTAB(N)"  3  3  3  "
660 PRINTTAB(N)"  3  3  3  "
670 PRINTTAB(N)"  3  3  /  "
680 PRINTTAB(N)"  3  3  3  "
690 PRINTTAB(N)"  3  3  /  "
700 PRINTTAB(N)"  3  3  3  "
710 PRINTTAB(N)"  3  3  3  "
720 RETURN
730 PRINTTAB(N)"  3  3  /  "
740 PRINTTAB(N)"  3  3  3  "
750 PRINTTAB(N)"  3  3  3  "
760 PRINTTAB(N)"  3  3  /  "
770 PRINTTAB(N)"  3  3  3  "
780 PRINTTAB(N)"  3  3  3  "
790 PRINTTAB(N)"  3  3  3  "
800 PRINTTAB(N)"  3  3  3  "
810 RETURN
820 PRINTTAB(N)"  3  3  /  "
830 PRINTTAB(N)"  3  3  3  "
840 PRINTTAB(N)"  3  3  3  3  "
850 PRINTTAB(N)"  3  3  3  "
860 PRINTTAB(N)"  3  3  /  "
870 PRINTTAB(N)"  3  3  3  "
880 PRINTTAB(N)"  3  3  3  "
890 PRINTTAB(N)"  3  3  3  "
900 RETURN
910 PRINTTAB(N)"  3  3  /  3  /  "
920 PRINTTAB(N)"  3  3  3  3  3  "
930 PRINTTAB(N)"  3  3  3  3  3  "
940 PRINTTAB(N)"  3  3  3  3  3  "
950 PRINTTAB(N)"  3  3  3  3  "

```

```
960 PRINTTAB(N)"  | | |  | | |  | | |"
970 PRINTTAB(N)"  | | |  | | |  | | |"
980 PRINTTAB(N)"  | | |  | | |  | | |"
990 RETURN
1000 PRINTTAB(N)"  | | |  | | |"
1010 PRINTTAB(N)"  | | |  | | |"
1020 PRINTTAB(N)"  | | |  | | |"
1030 PRINTTAB(N)"  | | |  | | |"
1040 PRINTTAB(N)"  | | |  | | |"
1050 PRINTTAB(N)"  | | |  | | |"
1060 PRINTTAB(N)"  | | |  | | |"
1070 PRINTTAB(N)"  | | |  | | |"
1080 RETURN
1090 PRINTTAB(N)"  | | |  | | |"
1100 PRINTTAB(N)"  | | |  | | |"
1110 PRINTTAB(N)"  | | |  | | |"
1120 PRINTTAB(N)"  | | |  | | |"
1130 PRINTTAB(N)"  | | |  | | |"
1140 PRINTTAB(N)"  | | |  | | |"
1150 PRINTTAB(N)"  | | |  | | |"
1160 PRINTTAB(N)"  | | |  | | |"
1165 RETURN
1200 PRINTTAB(N)"  | | |  | | |"
1210 PRINTTAB(N)"  | | |  | | |"
1220 PRINTTAB(N)"  | | |  | | |"
1230 PRINTTAB(N)"  | | |  | | |"
1240 PRINTTAB(N)"  | | |  | | |"
1250 PRINTTAB(N)"  | | |  | | |"
1260 PRINTTAB(N)"  | | |  | | |"
1270 PRINTTAB(N)"  | | |  | | |"
1280 RETURN
1290 PRINTTAB(N)"  | | |  | | |"
1300 PRINTTAB(N)"  | | |  | | |"
1310 PRINTTAB(N)"  | | |  | | |"
1320 PRINTTAB(N)"  | | |  | | |"
1330 PRINTTAB(N)"  | | |  | | |"
1340 PRINTTAB(N)"  | | |  | | |"
1350 PRINTTAB(N)"  | | |  | | |"
1360 PRINTTAB(N)"  | | |  | | |"
1370 RETURN
12000 GETA$: IFA$="" THEN 12000
12010 A=ASC(A$)
12020 IFA<65 OR A>79 THEN 12000
12030 PRINT "XXXXXXXXXX"
12040 DIM A-6400SUB280,370,460,550,640,730,820,91
0,100,1000,1090,10,190,1200,1290
12050 GOTO 12000
```

READY.

USER DEFINED CHARACTERS

This program uses the principles behind Hi-Resolution Graphics to plot points onto an area of user defined characters. The size of the area is limited to 128 characters.

RUNNING THE PROGRAM

When run, there will be a pause as the full 64 character set is copied into RAM. After this, the user will be asked to specify the size of the area to plot onto. If the area is too large to fit into 128 characters, the question will be asked again. After that the area will be displayed at the top left corner of the screen and co-ordinates can be plotted by entering the x and y position and whether plot or delete.

PROGRAM STRUCTURE

- 50 – 110 Copy characters from ROM to RAM
- 130 Set up the user defined characters.
- 150 Input dimensions of area. The reversed H is CTRL +H which disables the switch between upper and lower case.
- 160 – 210 Check size is within limits and display area.
- 250 – 440 Input co-ordinates and plot or unplot.
- 460 – 470 Calculate byte in memory and the bit to be changed.
- 500 – 550 Display border around area.
- 570 – 590 Blank the input line.
- 610 – 630 Display top and bottom lines of the border.

```

10 REM PROGRAM FOR PLOTTING POINTS ON
20 REM USER DEFINED CHARACTERS.
30 REM
40 REM COPY CHARACTER ROM INTO RAM
50 POKE56334,PEEK(56334)AND254
60 POKE1,PEEK(1)AND251
70 FORI=0TO1023
80 POKE12288+I,PEEK(53248+I):NEXT
90 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
100 POKE53272,(PEEK(53272)AND240)OR12
110 POKE56,48:POKE52,48
120 REM SET REVERSED CHARACTERS TO 255
130 FORI=1024TO2037:POKE12288+I,255:NEXT
140 REM INPUT SIZE OF AREA
150 INPUT"ENTER DIMENSIONS (X,Y)";X,Y
160 IFX=0ORY=0ORX*Y>128THEN150
170 BASE=13312:PRINT" ":GOSUB500
180 FORI=0TOY-1:FORJ=0TOX-1
190 POKE1065+J+I*40,128+J+I*X
200 POKE55337+J+I*40,1
210 NEXT:NEXT
220 CD#=LEFT$(" ",Y+2)
230 PRINT"#####MAX X="X*8-1
240 PRINT"#####MAX Y="Y*8-1
250 PRINTCD#
260 REM INPUT CO-ORDINATE FOR PLOT
270 INPUT"ENTER COORDINATE X=-1 TO END";X1,Y1
280 PRINTCD#:GOSUB570
290 IFX1>X*8-1ORY1>Y*8-1THEN250
300 IFX1=-1THEN480
310 IFX1<0ORY1<0THEN250
320 REM INPUT WHETHER PLOT OR DELETE
330 PRINTCD#:INPUT"PLOT OR DELETE";A#
340 PRINTCD#:GOSUB570
350 IFA#<"P"ANDR#<"D"THEN330
360 IFA#="D"THEN410
370 GOSUB460
380 REM PLOT POINT
390 POKEBYTE,PEEK(BYTE)AND(255-2*BIT)
400 GOTO250
410 GOSUB460
420 REM UNPLOT POINT
430 POKEBYTE,PEEK(BYTE)OR2*BIT
440 GOTO250
450 REM CALCULATE MEM LOC TO CHANGE
460 BYTE=BASE+INT(Y1/8)*X*8+INT(X1/8)*8+(Y1AND7)
470 BIT=7-(X1AND7):RETURN
480 END
490 REM DISPLAY BORDER AROUND AREA
500 PRINT" ":GOSUB610

```

```
510 FORI=1TOY
520 PRINT"█";TAB(X+1);"█"
530 NEXT
540 GOSUB610
550 RETURN
560 REM BLANK INPUT LINE
570 FORI=1TO60
580 PRINT" ";NEXT
590 PRINT:RETURN
600 REM TOP AND BOTTOM LINE OF BORDER
610 FORI=1TOX+2
620 PRINT"█";NEXT
630 PRINT:RETURN
REPEAT.
```


MACHINE CODE TO DATA

This program enables machine code programs to be converted to decimal data statements and added to the program. The machine code is read by PEEKing the memory and printing it to the screen. The computer is then made to put returns over the data and add them to the program. Each line of data created is seven values long.

RUNNING THE PROGRAM

When the user has typed run, the program will ask for the entry of start line number, line number increment, start address of machine code, and end address of machine code. Once this has been done, the data will be added to the program until the end address is reached. The data can then be used in the form of a basic loader by deleting the lines above 63000 and writing a short routine to read the data values and poke them into the correct memory.

PROGRAM STRUCTURE

63000-63120 Input the required parameters.
63160-63150 Put the parameters into safe memory locations.
63200-63220 Check for end and increase line number.
63230-63280 Set up screen for addition of data.
63290 Tell the computer that there are 3 characters in the keyboard buffer and put them in.
63300-63340 Update parameters and store them away.
63350 End of program and point where data is entered.

```

63000 PRINT"DATACREATE DECIMAL DATA ";
63010 PRINT"STMTS FROM MCHCODE"
63020 PRINT"DATA *** CBM MODEL 2-64-III ***"
63030 INPUT"DATASTART LINE # IIII";S$
63040 IFS$=" "THEN63030
63050 INPUT"DATASTEP IIII";T$
63060 IFT$=" "THEN63050
63070 PRINT"DATASTART ADDRESS DECIMAL IIII";
63080 INPUTB$:IFB$=" "THEN63070
63090 PRINT"DATAEND ADDRESS DECIMAL IIII";
63100 INPUTE$:IFE$=" "THEN63090
63110 S=VAL(S$):T=VAL(T$):B=VAL(B$)
63120 E=VAL(E$):F=B:L=F+6:PRINT"DATA"
63130 POKE831,INT(E/256)
63140 POKE832,E-INT(E/256)*256
63150 POKE828,T:GOTO63230
63160 T=PEEK(828)
63170 S=PEEK(826)*256+PEEK(827)
63180 L=PEEK(829)*256+PEEK(830)
63190 E=PEEK(831)*256+PEEK(832)
63200 IF L>=E THEN63350
63210 F=L+1:L=L+7
63220 PRINT"DATA"
63230 PRINTS;
63240 PRINT"DATA";
63250 FORP=FTOL
63260 PRINTMID$(STR$(PEEK(P)),2);",,":NEXTP
63270 PRINT"II "
63280 PRINT"GOTO63160'TTT";
63290 POKE 198,2:POKE631,13:POKE 632,13
63300 S=S+T
63310 POKE 826,INT(S/256)
63320 POKE 827,S-INT(S/256)*256
63330 POKE 829,INT(L/256)
63340 POKE 830,L-INT(L/256)*256:END
63350 END
READY.

```

PERSONAL INFORMATION RETRIEVAL

As the name suggests, this program allows the user to store personal information in the form of addresses or diary entries in the program. The entries are entered using the Update. Each entry can be up to 10 lines long and each line a maximum of 27 characters (for ease of reading).

RUNNING THE PROGRAM

Addresses are stored under address number and there is a maximum of 365 addresses. The date entries are entered as ddmmyy where dd is between 1 and 31, mmm jan, feb, mar, etc. And yy is the last two digits of the year e.g. 84 for 1984. When the date has been entered, it will be validated to ensure a true date and the full date will be displayed prior to data input.

If an entry already exists, the existing entry will be displayed. To change any part, press return to get to the required line and just make the change. A " character will appear at the beginning of the line to signify that a change has been made. When all changes have been made, pressing the ↑ key and return will add the new or changed entry to the existing data. Pressing the ? key and return will print the entry to printer (but not enter it into the data).

There are two data retrieval routines also in the program. The first is the list all routine which will list all entries to the screen. As this would run rather too fast without pauses, the display will halt until the shift key is pressed. Therefore, holding the shift key down will scroll the display and releasing the shift key will halt the display. To return to the menu before all entries have been listed, hold down the shift key and press the back arrow key.

The other data retrieval method is the find routine. When this routine is requested, the user must enter the search string and every entry containing this string at the beginning of any line in the entry will be displayed.

To save the data means saving the program as the data has been added to the end of the program. This can be done to either disk or tape. On the tape save, there are two passes, the first is the save and then the user is requested to rewind the tape and then the program is verified. Make a note of the message after verifying as the basic program cannot detect whether the save was successful or not. In the disk save, the error channel is read and if there is an error it will be displayed and the user will be asked whether he would like to try and save again.

The last routine is the end routine. This will cause a cold start of the program but before that it makes sure that the program has been saved if the user so wishes.

In the update, list all, and find routines, there is a slight pause because of the searching for the correct data entry. This is done with a Basic version of a restore to line number.

PROGRAM STRUCTURE

10	Calls the parameter set up routine.
15-65	Displays the menu page.
70-105	Get command and pass to routine.
110-150	End program.
155-295	Update routine.
155-185	Input address number or date and set up for input.
190-235	Input entry.
240-280	Add data to program.
285-295	Print entry.
300-315	Remove entry.
320-355	List all entries.
360-415	Save to disk.
420-450	Save to tape.
455-530	Find entry.
455-460	Get search string.
465-495	Find entry and set pointers to start of entry.
500-530	Display entry and set pointers to end of entry.
535-545	Pause for F1 key and return to menu.
550-595	Get address no. or date and check for which one.
600-650	Set up string for either date or address no.
655	Set up data line number in x.
660-700	Restore to line number in x.
705-720	Set up date parameters.
725	Read disk error channel.
730-810	Set up program parameters.
1000-end	Storage area for data.


```

520 POKE64, INT(BX/256):POKE63, BX-INT(BX/256)*256
525 POKE66, INT(AX/256)
530 POKE65, AX-INT(AX/256)*256:GOTO465
535 PRINT"    000      000PRESS '-1' TO CONTINUE"
540 GETAN$:IFAN$<" "THEN540
545 RETURN
550 INPUT"ENTER DATE/ADDRESS NO.":C$
555 IFC$=""THEN550
560 DD=VAL(C$):LL=LEN(C$):IFLL>3THEN575
565 IFDD>3650RDD<1THEN550
570 Y=-1:M=1:RETURN
575 IFDD<10RDD>31THEN550
580 Y=VAL(RIGHT$(C$,2)):IFY>0THENY=Y-Y0+1900
585 X=3+(DD<10):MM$=MID$(C$,X,3)
590 FORM=1T012:IFMM$=S$(M)THENRETURN
595 NEXT:GOTO550
600 IFY<-1THEN610
605 AD$="      000 ADDRESS /0."+STR$(DD):RETURN
610 TT$="TH.":S=DD:X=Y+Y0:S=S+X*365
615 IFM=>3THENS=S-INT(M*F1+F2):X=X+1
620 S=S+INT(M*31+(X-1)/F):W=S-INT(S/7)*7
625 WD$=W$(W):MD$=M$(M)
630 IFDD=10RDD=21ORDD=31THENTT$="ST."
635 IFDD=20RDD=22THENTT$="ND."
640 IFDD=30RDD=23THENTT$="RD."
645 AD$="      "+WD$+STR$(DD)+TT$+" "+MD$+STR$(Y+Y0)
650 RETURN
655 X=INT(Y*T+T+M*1000+DD*10):RETURN
660 GOSUB675:C=C+4:POKE66,INT(C/256)
665 POKE65,CAND255:IFC<>4THENREADC$
670 RETURN
675 RESTORE:READXX$:AD=PEEK(65)+PEEK(66)*256
680 AA=PEEK(AD+3)+PEEK(AD+4)*256
685 IFAA>XTHENC=0:RETURN
690 IFAA<>XTHEN700
695 C=AD+1:RETURN
700 AD=PEEK(AD+1)+PEEK(AD+2)*256-1:GOTO680
705 X=X-T:Y=INT((X-1000)/T):M=INT((X-Y*T)/1000)
710 DD=INT(((X-Y*T)-M*1000)/10)
715 IFY=-1THENDD=DD+(M-1)*HU
720 RETURN
725 OPEN1,8,15:INPUT#1,EN,EN$,ET,ES:CLOSE1:RETURN
730 X=0:HU=100:T=12000:F1=.4:F2=2.3:F=4
735 Y0=1984:DIMM$(12),S$(12):POKE53280,2
740 POKE53281,2:POKE53272,23
745 W$(0)="TUESDAY":W$(1)="WEDNESDAY"
750 W$(2)="THURSDAY":W$(3)="FRIDAY"
755 W$(4)="SATURDAY":W$(5)="SUNDAY"
760 W$(6)="MONDAY"
765 M$(1)="JANUARY":M$(2)="FEBRUARY"
770 M$(3)="MARCH":M$(4)="APRIL":M$(5)="MAY"

```

```
775 M$(6)=" JUNE":M$(7)=" JULY":M$(8)=" AUGUST"  
780 M$(9)=" SEPTEMBER":M$(10)=" OCTOBER"  
785 M$(11)=" NOVEMBER":M$(12)=" DECEMBER"  
790 S$(1)="JAN":S$(2)="FEB":S$(3)="MAR"  
795 S$(4)="APR":S$(5)="MAY":S$(6)="JUN"  
800 S$(7)="JUL":S$(8)="AUG":S$(9)="SEP"  
805 S$(10)="OCT":S$(11)="NOV":S$(12)="DEC"  
810 GOTO15  
1000 DATA*  
1010 DATA"/ICK HAMPSHIRE PUBLICATIONS  
1011 DATA"167/169 RT. PORTLAND VT.  
1012 DATA"LT/RT/ 01  
1013 DATA"  
1014 DATA"TEL 01-636 6354  
63999 DATA*  
READY.
```


WILL O' THE WISP

WILL O' THE WISP

DESCRIPTION

Will o' the Wisp is a great adventure game for devious and nimble minded computophiles. Trudging through a vast forest in the guise of a fairly simple minded woodsman, and destined to wed the fair Brunhilde on the morn, the player seeks a final day of adventurous freedom before the dread shackles of marriage are fixed firmly about his ankles. The poor fool recklessly throws his life on the mere whims of the program until, eventually, more by good luck than management, he finds himself outside the Great Cave. Tempted in by a floating wisp of mist the player ventures fearlessly on, then to his surprise a rumbling sounds behind him, the doors to the cave swing shut, and the only way to go is onwards.

Dread horrors await the fearless woodland yokel, and this is all you are going to get, dear Reader. Onward and may the best yokel win!

RUNNING THE PROGRAM

Load and type 'RUN', O fearless yokel!

```

1 POKE53280,6:POKE53281,7:PRINT"■"
2 GOTO6
4 PRINT"WELL I'LL HELP YOU ALRIGHT... I'LL ";;R
ETURN
6 CLR:GOSUB8:GOSUB54:GOSUB86::END
8 Y=10:F=20:G=30:H=40:R=50:J=60:K=70:L=80
10 REM
12 PRINT"XXXXXXXXXXXXXXXXXXXXHILL 'O THE WISP"
14 PRINT"XXXXXXXXXXXXXXXXXXXXBY MARK CAPELLA"
16 PRINT"XXXXXXXX YOU ARE A POOR COUNTRY BOY WH
0 IS
18 PRINT"DESTINED TO MARRY THE BEAUTIFUL AND KI
ND
20 PRINT"XXXXXXXXBRUNHILDE ON TOMORROW MORN. BUT BEING

22 PRINT"YOUNG AND STUPID, YOU HAVE DECIDED TO
24 PRINT"SPEND YOUR LAST DAY OF FREEDOM
26 PRINT"EXPLORING THE ENDLESS FOREST THAT
28 PRINT"SURROUNDS YOUR SHACK. YOU COULD STAY
30 PRINT"HOME AND WATCH TV BUT THEN YOU'D NEVER

32 PRINT"GET LOST IN THE FOREST AND FIND HIGH
34 PRINT"ADVENTURE AND LEARN TO BECOME A MAN
36 PRINT"AND WHAT THE MEANING OF LIFE IS.
38 PRINT"X NOW IN THE TRADITION OF ALL THESE
40 PRINT"TYPES OF GAMES, I'M SURE YOU'LL GO ALO
NG
42 PRINT"XXXXXXXXWITH THE OBVIOUS AND GET YOURSELF LOS
T.
44 PRINT"SO NOW THAT I'M BUSY, WHY DON'T YOU DO

46 PRINT"JUST THAT?? HAVE FUN ANYHOW...
48 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX PRESS RETUR
N WHEN YOU ARE READYXXXXXXXXXX";
50 INPUT AN$
52 PRINT"XXXXXXXXXXXXXXXXXXXXHILL 'O THE WISPXXXX":R
ETURN
54 DIM FL%(10),ITM%(10),ITM$(10),EX$(10)
56 FORI=1TO9:READITM%(I),ITM$(I),EX$(I):NEXT
58 DATA 30,BOTTLE,AN EMPTY BOTTLE LIES DISCARDE
D NEARBY
60 DATA 67,WATER,A SMALL POOL OF WATER COLLECTS
HERE
62 DATA 0,BROOM,A SMELLY WITCHES BROOM FLOATS N
EARBY
64 DATA 59,BALL,A MAGICAL CRYSTAL BALL GLISTENS
HERE
66 DATA 51,CARPET,A FLYING CARPET IS ROLLED UP
HERE,0,,RING
68 DATA 75,GUANO,A PILE OF BAT GUANO SLOWLY ROT
S HERE

```

```

70 DATA 31,PEEL,AN OLD BANANA PEEL IS LYING IN
THE DIRT
72 DATA 10,CAN,AN EMPTY BEER CAN RUSTS HERE
74 PSMZ=1
76 ITM$(6)="A GOLD WEDDING RING IS IN YOUR POCK
ET FOR YOUR SWEET BRUN"
78 ITM$(6)=ITM$(6)+"HILDA FROM THE GREAT MAGICI
AN RALPH"
80 TRUE=-1
82 FALSE=0
84 RETURN
86 :
88 GOSUB 104
90 IF PS=66 THEN GOSUB 410
92 IF EOG THEN RETURN
94 GOSUB 132
96 GOSUB 152
98 GOSUB 200
100 IF EOG THEN RETURN
102 GOTO 88
104 IF INT(RND(1)*100)+1<11 THEN PRINT:PRINT"A
SMALL WISP IS FLOATING HERE..."
106 MVE=MVE+1:IF MVE=1 THEN PRINT:GOTO 112
108 IF MVE=4 THEN MVE=0
110 RETURN
112 IFPS<11THENONPSGOTO606,622,634,642,652,662,
676,690,704,716
114 IFPS<21THENONPS-YGOTO734,752,766,778,792,80
4,806,808,810,812
116 IFPS<31THENONPS-FGOTO814,816,828,840,850,85
8,864,876,884,896
118 IFPS<41THENONPS-GGOTO902,914,920,928,932,94
2,950,956,966,972
120 IFPS<51THENONPS-HGOTO978,982,984,986,988,99
0,992,994,1004,1012
122 IFPS<61THENONPS-RGOTO1018,1026,1034,1040,10
48,1054,1060,1066,1072,1078
124 IFPS<71THENONPS-JGOTO1082,1090,1096,1102,11
08,1116,1124,1132,1140,1148
126 IFPS<81THENONPS-KGOTO1154,1160,1168,1174,11
80,1186,1192,1196,1202,1210
128 IFPS<91THENONPS-LGOTO1218,1224,1228,1234
130 RETURN
132 PRINT:FORI=1TO9:IFIT%(I)=PSTHENPRINT""EX$(I
)
134 NEXT:PRINT"OBVIOUS EXITS:";:IFN>0THENPRINT"
N";
136 IFS>0THENPRINT" S";
138 IFE>0THENPRINT" E";
140 IFW>0THENPRINT" W";

```

```

142 IFNE>0THENPRINT" NE";
144 IFNW>0THENPRINT" NW";
146 IFSE>0THENPRINT" SE";
148 IFSW>0THENPRINT" SW";
150 RETURN
152 REM
154 PRINT:INPUT"请输入要说的":COM$
156 IFCOM$=""THENPRINT"请输入":GOTO 154
158 IFCOM$="AGAIN"ORCOM$="REPEAT">ANDVERB$<>"
THENRETURN
160 IFCOM$="AGAIN"ORCOM$="REPEAT">ANDVERB$=""THE
N1242
162 CP=0:GOSUB186
164 VERB$=BY$
166 GOSUB 186
168 NOUN$=BY$:IFVE$="NORTH"THENVE$="N"
170 IFVE$="SOUTH"THENVE$="S"
172 IFVE$="EAST"THENVE$="E"
174 IFVE$="WEST"THENVE$="W"
176 IFVE$="NORTHEAST"THENVE$="NE"
178 IFVE$="NORTHWEST"THENVE$="NW"
180 IFVE$="SOUTHEAST"THENVE$="SE"
182 IFVE$="SOUTHWEST"THENVE$="SW"
184 RETURN
186 REM
188 IF CP>LEN(CO$)THEN BY$="":RETURN
190 FOR SP=CP+1TOLEN(CO$):IF MID$(CO$,SP,1)<>"
"THEN NEXT
192 BY$=MID$(CO$,CP+1,SP-CP-1):CP=SP:IFBY$="THE
"ORBY$="IT"ORBY$="A"THEN 188
194 IF BY$="AN"ORBY$="THAT"ORBY$=""THEN 188
196 IF BY$="JUMP"ORBY$="GO"ORBY$="TRAVEL"ORBY$=
"WALK"ORBY$="MOVE"THEN 188
198 RETURN
200 REM
202 IF(VE$="N" OR VE$="E" OR VE$="S" OR VE$="W"
) AND IT$(8)=PS THEN 1262
204 IF(VE$="NE" OR VE$="NW" OR VE$="SE" OR VE$=
"SW") AND IT$(8)=PS THEN 1262
206 IF VE$="N" THEN 250
208 IF VE$="E" THEN 254
210 IF VE$="S" THEN 258
212 IF VE$="W" THEN 262
214 IF VE$="NE" THEN 266
216 IF VE$="NW" THEN 270
218 IF VE$="SE" THEN 274
220 IF VE$="SW" THEN 278
222 IF VE$="GET"ORVE$="TAKE"ORVE$="GRAB"ORVE$="
CARRY"THEN328
224 IF VE$="DROP"ORVE$="THROW"ORVE$="TOSS"ORVE$
="LEAVE"THEN 348

```

```

226 IF VE#="I" OR LEFT$(VE#,4)="INVE"THEN368
228 IF VE#="BREAK" OR VE#="SMASH" OR VE#="DESTR
OY" THEN 382
230 IF VE#="D"ORVE#="DOWN"THEN400
232 IF VE#="IN" OR VE#="ENTER" THEN 532
234 IF VE#="OUT" OR VE#="EXIT" THEN 544
236 IF VE#="LOOK" OR VE#="SEE" OR VE#="VIEW" TH
EN MVE=0:RETURN
238 IFVE#="SCORE"ORVE#="TOTAL"THENPRINT"YOU ARE
STILL STUCK IN THE MAZES ";:GOTO1264
240 IF VE#="HELP" THEN 566
242 IFVE#="END"ORVE#="QUIT"ORVE#="STOP"ORVE#="D
ONE"THEN 586
244 IFVE#="KILL"ORVE#="ATTACK"THENPRINT"YOU HAV
E TO TELL ME HOW TO";:GOTO1268
246 PRINT"WHAT???:":RETURN
248 REM
250 IF N THEN PS=N:GOTO 286
252 GOTO 280
254 IF E THEN PS=E:GOTO 286
256 GOTO 280
258 IF S THEN PS=S:GOTO 286
260 GOTO 280
262 IF W THEN PS=W:GOTO 286
264 GOTO 280
266 IF NE THEN PS=NE:GOTO 286
268 GOTO 280
270 IF NW THEN PS=NW:GOTO 286
272 GOTO 280
274 IF SE THEN PS=SE:GOTO 286
276 GOTO 280
278 IF SW THEN PS=SW:GOTO 286
280 PRINT"YOU CANNOT GO THAT WAY."
282 IF PS=36THENPRINT"> BANG! < AS YOU HIT THE
WALL, THE ROCKSWAKE";:GOTO1274
284 RETURN
286 BW#="A BLUE WILL O' THE WISP BECKONS FROM "
"
287 GW#="A GREEN WILL O' THE WISP BECKONS FROM
"
288 MVE=0
290 IF PS=23 AND NOT FLG%(1) THEN FLG%(1)=TR:PR
INT:GOTO1244
292 IF PS=84 AND NOT FLG%(3) THEN PRINT:PRINT"Y
OU'VE FOUND PRUDENCE !!!"
294 IF PS=16THENPRINTBW#"THEEAST"
296 IF PS=18THENPRINTBW#"THEEAST"
298 IF PS=19THENPRINTBW#"THESOUTH"
300 IF PS=20THENPRINTBW#"THESOUTHWEST"
302 IF PS=21THENPRINTBW#"THESOUTHEAST"
304 IF PS=22THENPRINTBW#" INSIDE THE CAVE"

```

```

306 IF PS=23THENPRINTW$" THE CAVE'S ENTRANCE"
308 IF(FLG%(2)◊1)ORFLG%(3)◊0THEN326
310 IF PS=30THENPRINTGW$" THE NORTHEAST"
312 IF PS=37THENPRINTGW$" THE NORTH"
314 IF PS=42THENPRINTGW$" THE EAST"
316 IF PS=43THENPRINTGW$" THE NORTHEAST"
318 IF PS=44THENPRINTGW$" THE SOUTHEAST"
320 IF PS=45THENPRINTGW$" THE SOUTHWEST"
322 IF PS=46THENPRINTGW$" THE SOUTHWEST"
324 IF PS=47THENPRINTGW$" THE SOUTH"
326 RETURN
328 REM
330 FOR I=1TO9
332 IF IT%(I)=NO$THEN336
334 NEXT:PRINT"THAT IS BEYOND YOUR POWER.":RETU
RN
336 IF IT%(I)=-1THENPRINT"YOU ARE ALREADY CARRY
ING IT!":RETURN
338 IFIT%(I)=0 THENPRINT"YOU CHEATER!! YOU HAVE
TO WORK FOR THAT":RETURN
340 IFIT%(I)◊P$THENPRINT"I DO NOT SEE ANY "NO$
" HERE.":RETURN
342 IFI=2ANDIT%(1)◊-1THENPRINT"WHAT WILL YOU C
ARRY THE WATER IN?? ":GOTO1250
344 PRINT"OK! WHY NOT?":IT%(I)=-1
346 RETURN
348 REM
350 FOR I=1 TO 9
352 IF IT%(I)=NO$ THEN 356
354 NEXT:PRINT"THAT IS BEYOND YOUR POWER.":RETU
RN
356 IF IT%(I)◊-1THENPRINT"YOU ARE NOT CARRYING
IT!":RETURN
358 IF I=1 AND IT%(2)=-1THENIT%(2)=PS:REM IF DR
OP BOTTLE, ALSO WATER
360 PRINT"OK! WHY NOT?":IT%(I)=PS
362 IF PS=84 AND IT%(2)=84 AND NOT FLG%(3) THEN
1252
364 IF PS=84 AND IT%(2)=84 AND NOT FLG%(3)THEN
FLG%(3)=TR:IT%(3)=PS:IT%(2)=-2
366 RETURN
368 REM
370 PRINT
372 PRINT:PRINT"YOUR INVENTORY IS AS FOLLOWS":P
RINT:TMP=0
374 FOR I=1 TO 9
376 IF IT%(I)=-1 THEN PRINT""IT%(I):TMP=TMP+1
378 NEXT:IF TMP=FALSE THEN PRINT" (NOTHING)"
380 RETURN

```

```

382 REM
384 IF NO$="MIRROR" THEN 388
386 PRINT"HOW CAN I BREAK THAT FOR YOU?":RETURN

388 REM
390 IF PS<>35THENPRINT"I DON'T SEE ANY MIRROR H
ERE. PERHAPS IF YOU KEEP ";
392 IF PS<>35THENPRINT"LOOKING YOU'LL FIND ONE.
":RETURN
394 PRINT"THERE WAS SOMETHING BEHIND THAT MIRRO
R. A HUNGRY TROLL WAS BEHIND";
396 PRINT" THAT MIRROR! HE LEFT OUT AND MADE Y
OU HIS SUPPER. TOO BAD ";
398 PRINT"THOUGH, YOU JUST CAN'T TRUST A HUNGR
Y TROLL.":EOG=TR:RETURN
400 REM
402 IF PS<>61THENPRINT"THERE IS NO WAY TO TRAVE
L THROUGH THE SOLID ROCK";
404 IF PS<>61THENPRINT" STUPID!!!":RETURN
406 PRINT"BOY ARE YOU DUMB. I TOLD YOU NOT TO J
UMPFAND NOW YOU'VE KILLED";
408 PRINT" YOURSELF.":EOG=TR:RETURN
410 REM
412 IF FLG%(2) THEN 488
414 PRINT:PRINT"RALPH LOOKS UP AT YOU SLOWLY...
THE ACCUMULATED WISDOM";
416 PRINT" OF THE AGES POURING FROM HIS SWEATY
BROW. WELL?? HE CACKLES. ";
418 PRINT"WHAT IS IT THAT YOU WANT SO BADLY AS
TO INTERRUPT ME FROM MY ";
420 PRINT"STUDIES?????";
422 INPUTANS$:IFANS$="FREEDOM"ORANS$="HELP"ORAN
S$="OUT"ORANS$="HOME"THEN436
424 PRINT:INPUT"WHAT??? HE SAYS... DO YOU WANT
TO GET OUT OF HERE????";ANS$
426 IF LEFT$(ANS$,1)="Y" THEN 436
428 PRINT:PRINT"WELL HE MUTTERS, SINCE I CAN'T
SEEM TO UNDERSTAND YOU...";
430 PRINT" I'LL JUST KILL YOU!! AND WITH THAT
HE WAVES HIS HANDS IN A"
432 PRINT"MAGICAL MOTION AND YOU DISAPPEAR IN A
CLOUD OF BRIGHT ORANGE ";
434 PRINT"DUST.":EOG=TR:RETURN
436 IF ITM%(4)=-1OR ITM%(5)=-1THENPRINT:PRINT"I
'D LIKE TO HELP YOU, BUT THAT IS MY"
438 IF ITM%(4)=-1OR ITM%(5)=-1THENPRINT"TREASUR
E THAT YOU HAVE THERE, AND I
440 IF ITM%(4)=-1OR ITM%(5)=-1THENPRINT"DISLIKE
A CLUMSY THIEF. SO SAY GOODBYE
442 IF ITM%(4)=-1OR ITM%(5)=-1THENPRINT"QUICK C

```



```

RUSE YOU'RE A GONER !!! AND WITH
444 IF ITM%(4)=-1OR ITM%(5)=-1THENPRINT"THAT HE
KILLS YOU !!!":EOG=TR:RETURN
446 PRINT:PRINT"I'LL HELP YOU OUT... BUT FIRST
I NEED A FAVOR. THERE IS A CERTAIN ";
448 PRINT"WITCH UP NORTH THAT I ONCE ANGERED AND
SHE HAS ENCHANTED MY ";
450 PRINT"MAGIC STAFF AWAY FROM ME":PRINT
452 PRINT"SHE TURNED IT INTO A BROOMSTICK AND I
WANT IT BACK! SO IF YOU";
454 PRINT"CAN STEAL IT FROM HER, I'LL SHOW
YOU HOW TO GET OUT OF HERE."
456 INPUT"DO YOU WANT TO TRY??NO":ANS#
458 ILEFT$(ANS#,1)O"Y"THEN PRINT:PRINT"WELL I
NEVER!!! I TRY TO HELP ";
460 ILEFT$(ANS#,1)O"Y"THENPRINT"YOU AND YOU
CAN'T HELP ME!!! ";GOSUB 4
462 ILEFT$(ANS#,1)O"Y"THENPRINT"HELP YOU INTO
A GRAVE !!AND WITH THAT HE ";
464 ILEFT$(ANS#,1)O"Y"THENPRINT"PLUNGES A SWO
RD INTO YOUR CHEST. YOU DIE."
466 ILEFT$(ANS#,1)O"Y"THEN EOG=TR:RETURN
468 PRINT:PRINT"GOOD!!! NOW AWAY YOU GO !!! BUT
FIRST.. THE WAY INTO THE WITCHES ";
470 PRINT"LAND IS NOT SIMPLE. ALL PATHS ARE
CHARMED SO AS TO ONLY LEAD YOU ";
472 PRINT"AWAY FROM THE LAND."
474 PRINT:PRINT"BUT IF YOU SIMPLY REVERSE THE T
AUNTINGS OF THE GREEN WILL TO THE ";
476 PRINT"WISP, YOU'LL GET THERE EASILY ENOUGH
H. OK... NOW FOR THE BIG ";
478 PRINT"FINISH !!!"
480 PRINT:PRINT"THE MAGICIAN WAVES HIS HANDS, A
ND YOU BLACK-OUT IN A FRENZY OF ";
482 PRINT"SENSATIONS. YOU AWAKE TO FIND..."
484 PS=69:FLG%(2)=TR:MV=0:GOSUB 104
486 RETURN
488 REM
490 PRINT:PRINT"WELL MUTTERS RALPH, I SEE YOU'R
E BACK."
492 IFITM%(4)=-1ORITM%(5)=-1THENPRINT"AND TRYIN
G TO STEAL MY TREASURE!!! BOY
494 IFITM%(4)=-1ORITM%(5)=-1THENPRINT"NOW YOU'V
E GONE AND DONE IT... I'M MAD!!
496 IFITM%(4)=-1ORITM%(5)=-1THENPRINT"AND A MAD
WIZARD IS DANGEROUS!!! >POOF<
498 IFITM%(4)=-1ORITM%(5)=-1THENPRINT"YOU ARE D
EAD !!!":EOG=TR:RETURN
500 IFITM%(3)O-1THENPRINT"AND EMPTY HANDED !!!
WELL I JUST CAN'T

```

```

502 IFITM%(3) <> -1 THEN PRINT "ABIDE ANY FREELoader
S. BEGONE GNAT!!!
504 IFITM%(3) <> -1 THEN PRINT "AND WITH THAT HE CLa
PS HIS HANDS. YOU
506 IFITM%(3) <> -1 THEN PRINT "TURN INTO A GNAT, AN
D GET EATEN BY THE
508 IFITM%(3) <> -1 THEN PRINT "NEAREST SPIDER.":EOG
=TR:RETURN
510 ITX(6)=-1:ITX(3)=0:PRINT "AND WITH THE BROOM
!! GOODY-GOODY HE
512 PRINT "YELLS !!:FOR FINDING MY STAFF I HEREB
Y
514 PRINT "GRANT YOU YOUR FREEDOM AND A PRESENT
!!
516 PRINT "CHECK YOUR BELONGINGS LATER HE CACKLE
S"
518 PRINT:PRINT ">POOF!< YOU ARE STANDING UNDER
A LARGE
520 PRINT "BEAUTIFUL SPREADING TREE.":GOSUB368
522 PRINT "YOUR BELOVED BRUNHILDE IS RUNNING
524 PRINT "TOWARDS YOU THROUGH THE FIELD IN SLOW

526 PRINT "MOTION... LIFE IS BEAUTIFUL ONCE AGAI
N
528 PRINT "AND ALL IS WELL. I HOPE YOU'VE ENJOYE
D YOUR LITTLE EXCURSION."
530 EOG=TR:RETURN
532 REM
534 IF PS=22 AND NO$="CAVE" THEN VE$="SW":GOTO
206
536 IF PS=7 AND NO$="HOUSE" THEN VE$="E":GOTO 2
06
538 IF PS=48 AND NO$="CASTLE" THEN VE$="S":GOTO
206
540 IF PS=64 THEN VE$="N":GOTO 206
542 PRINT "ENTER WHAT ???":RETURN
544 REM
546 IFPS=23ANDNO$="CAVE"THENPRINT "THE BARS ARE
QUITE SOLID AND YOU ARE BUT
548 IFPS=23ANDNO$="CAVE"THENPRINT "A POOR COUNTR
Y BOY. YOU CANNOT GO OUT":RETURN
550 IFPS=1ANDNO$="HOUSE"THENVE$="W":GOTO 262
552 IFPS=49THENVE$="N":GOTO 206
554 IF PS=56THENVE$="W":GOTO 206
556 IF PS=57 THEN VE$="E":GOTO 206
558 IF PS=65 THEN VE$="S":GOTO206
560 IF PS=50 THEN VE$="E":GOTO 206
562 IF PS=51 THEN VE$="W":GOTO 206
564 PRINT "EXIT WHAT ???":RETURN
566 REM
568 PRINT:INPUT "DO YOU REALLY WANT A LITTLE HEL

```

```

P  ♀":ANS#
570 IF LEFT$(ANS$,1)◇>"Y" THEN PRINT"WELL, SINCE NOT, BACK YOU GO.":RETURN!
572 PRINT:PRINT"‘GET’ WILL LET YOU PICK UP AN OBJECT ‘DROP’ PUTS IT DOWN
574 PRINT"‘INVENTORY’ PRINTS YOUR SUPPLIES ‘N’ GOES NORTH, ‘SW’ GOES";
576 PRINT" SOUTHWEST ETC ‘END’ LEAVES THE GAME"

578 PRINT:PRINT"IF YOU ENCOUNTER A WITCH AND WISH TO KILL HER (WHY NOT)";
580 PRINT" YOU HAVE TO FIGURE OUT HOW, I WON’T TELL YOU, BUT IT WILL
582 PRINT"INVOLVE SOME OBJECTS AND ACTIONS THAT

584 PRINT"YOU WILL HAVE AVAILABLE. THAT’S ALL."
:PRINT:RETURN
586 REM
588 PRINT:INPUT"DO YOU REALLY WISH TO QUIT NOW"
:ANS#
590 IF LEFT$(ANS$,1)◇>"Y" THENPRINT"WELL IF YOU ‘VE CHANGED YOUR MIND, THEN
592 IF LEFT$(ANS$,1)◇>"Y" THENPRINT"BACK YOU’LL GO!":RETURN
594 PRINT:PRINT"OK, BUT YOU DIDN’T DO SO WELL."

596 IF PSC23 THENPRINT"YOU DIDN’T EVEN FIND THE CAVES !!!":GOTO 604
598 IF NOT FLG%(2)THENPRINT"YOU DIDN’T MEET THE MAGICIAN !!!":GOTO 604
600 IF NOT FLG%(3)THENPRINT"YOU NEVER KILLED THE WITCH !!!":GOTO 604
602 PRINT"YOU DIDN’T GET THE BROOM TO THE MAGICIAN"
604 PRINT:PRINT"SIGH... WELL I SUPPOSE YOU TRIE D..." :EDG=TR:RETURN
606 REM
608 PRINT"YOU ARE IN A SMALL FARMHOUSE. IT ISN’T
610 PRINT"MUCH BUT YOU CALL IT HOME. IN FACT IT
612 PRINT"ISN’T ANYTHING TO BE PROUD OF... IT IS
614 PRINT"REALLY PRETTY SHABBY... ACTUALLY IT IS
616 PRINT"DISGUSTING BUT YOU CALL IT HOME. ONE DOOR EXITS TO THE WEST.
618 N=0:E=0:S=0:W=7:NE=0:NW=0:SE=0:SW=0
620 RETURN
622 REM
624 PRINT"YOU ARE IN THE FOREST STILL. IT LOOKS

```

```

626 PRINT"LIKE YOU MAY BE LOST BUT YOU NEVER KH
OW
628 PRINT"WITH THESE POOR FARM PEASANTS.
630 N=0:E=0:S=0:W=0:NE=4:NW=3:SE=0:SW=0
632 RETURN
634 REM
636 PRINT"YOU ARE IN A FOREST OF HUGE TREES. TH
E FOREST IS FULL OF NUTS ..
638 N=0:E=0:S=0:W=0:NE=0:NW=0:SE=2:SW=5
640 RETURN.
642 REM
644 PRINT"IF YOU ARE NOT LOST YET THEN YOU KNOW

646 PRINT"THAT YOU ARE IN THE FOREST. (DOES THA
T MAKE ANY SENSE?)
648 N=0:E=0:S=0:W=0:NE=0:NW=0:SE=15:SW=2
650 RETURN
652 REM
654 PRINT"YOU ARE IN A FOREST OF HUGE TREES. TH
E
656 PRINT"FOREST IS FULL OF NUTS ..
658 N=0:S=0:E=0:W=0:NE=3:NW=0:SE=0:SW=6
660 RETURN
662 REM
664 PRINT"THE FOREST IS ALL AROUND YOU. YOU ARE
IN";
666 PRINT"THE FOREST. IN THE FOREST YOU ARE. DO

668 PRINT"YOU NEED ANY MORE HELP? YOU ARE IN TH
E HUGE FOREST WITH THE REST";
670 PRINT" OF THE NUTS.
672 N=0:E=0:S=0:W=0:NE=5:NW=0:SE=7:SW=0
674 RETURN
676 REM
678 PRINT"YOU ARE OUTSIDE YOUR DILAPIDATED RUND
OWN";
680 PRINT"DIRTY LITTLE FARMHOUSE. YOU CAN BREAT
HE
682 PRINT"FRESH AIR AGAIN. A WELL WORN PATH LEA
DS
684 PRINT"TO THE NORTH AND INTO THE FOREST.
686 N=5:E=1:S=0:W=0:NE=0:NW=0:SE=0:SW=0
688 RETURN
690 REM
692 PRINT"YOU ARE IN THE FOREST. GREAT MUSHROOM
S
694 PRINT"GROW HERE. THEY LOOK GOOD TO EAT BUT
696 PRINT"COULD BE VERY WELL POISONOUS. I WOULD

698 PRINT"NOT TRY TO EAT THEM IF I WERE YOU.
700 N=6:E=0:S=0:W=0:NE=0:NW=0:SE=9:SW=0

```

```

702 RETURN
704 REM
706 PRINT"THE FOREST IS A LOVELY PLACE ISN'T IT
?
708 PRINT"IT HAS TREES AND SQUIRRELS AND NUTS A
ND
710 PRINT"BUSHES AND YOU WHO ARE MOST LIKELY LO
ST!
712 N=0:E=0:S=0:W=0:NE=0:NW=0:SE=10:SW=0
714 RETURN
716 REM
718 PRINT"BOY YOU JUST KEEP WALKING DEEPER AND
720 PRINT"DEEPER INTO THE FOREST. WHY DID YOU H
AVE";
722 PRINT"TO LEAVE THAT NICE LITTLE RUN DOWN
724 PRINT"SHANTY OF YOURS IN SEARCH OF ADVENTUR
E?
726 PRINT"IF YOU WANTED ADVENTURE THEN YOU ARE
IN
728 PRINT"THE WRONG FANTASY.
730 N=0:E=0:S=0:W=0:NE=11:NW=9:SE=0:SW=0
732 RETURN
734 REM
736 PRINT"YOU ARE IN THE FOREST A LONG WAY
738 PRINT"FROM HOME. MONSTERS COULD BE ANYWHERE
.
740 PRINT"THEY COULD BE... BEHIND YOU NOW!!! HA
-HA";
742 PRINT"I'LL BET I SCARED YOU DIDN'T I??? WEL
L
744 PRINT"TO TELL THE TRUTH YOU ARE JUST IN THE
746 PRINT"FOREST WITH THE SQUIRRELS.
748 N=0:E=0:S=0:W=0:NE=0:NW=0:SE=12:SW=10
750 RETURN
752 REM
754 PRINT"THE FOREST IS GETTING DARKER AND YOU
ARE";
756 PRINT"OUT WAY PAST MY BEDTIME. MAYBE YOU
758 PRINT"SHOULD TURN AROUND AND JUST GO HOME.
760 PRINT"THAT IS IF YOUR SHACK IS STILL STANDI
NG.
762 N=0:E=0:S=0:W=0:NE=13:NW=11:SE=0:SW=0
764 RETURN
766 REM
768 PRINT"BOY IT IS REALLY DARK NOW... MONSTERS
770 PRINT"COULD BE ANYWHERE... WHY NOT GO HOME
TO
772 PRINT"YOUR MOTHER?
774 N=14:E=0:S=0:W=9:NE=17:NW=0:SE=0:SW=0

```

```

776 RETURN
778 REM
780 PRINT"AHA! YOU ARE IN A CLEARING AND CAN SE
E
782 PRINT"A TRAIL YOU ARE FAMILIAR WITH LEADING
TO";
784 PRINT"THE SOUTH-WEST. OF COURSE THERE ARE
786 PRINT"OTHER TRAILS BUT THIS SEEMS THE SMART
ESTMOVE AT THIS TIME.
788 N=15:E=0:S=13:W=0:NE=0:NW=0:SE=0:SW=7
790 RETURN
792 PRINT"WELL FOR THE HOPELESSLY LOST... YOU A
RE
794 PRINT"IN THE LARGE DARK MEAN SCARY NASTY OL
D
796 PRINT"FOREST AND MOST LIKELY YOU'LL DIE HER
E
798 PRINT"AND GET EATEN BY THE ROTTEN SQUIRRELS

800 N=0:E=0:S=14:W=0:NE=0:NW=4:SE=0:SW=0
802 RETURN
804 GOSUB 1270:N=0:E=18:S=17:W=0:NE=0:NW=15:SE=
0:SW=0:RETURN
806 GOSUB 1270:N=16:E=0:S=0:W=0:NE=0:NW=0:SE=0:
SW=13:RETURN
808 GOSUB 1270:N=15:E=19:S=17:W=16:NE=15:NW=15:
SE=17:SW=17:RETURN
810 GOSUB 1270:N=15:E=15:S=20:W=18:NE=15:NW=15:
SE=15:SW=17:RETURN
812 GOSUB 1270:N=19:E=15:S=15:W=17:NE=15:NW=17:
SE=15:SW=21:RETURN
814 GOSUB 1270:N=17:E=15:S=0:W=17:NE=20:NW=17:S
E=22:SW=0:RETURN
816 PRINT"YOU'RE IN FRONT OF A LARGE CAVE ENTRA
NCE";
818 PRINT"WHAT APPEARS TO BE A HUGE POT OF GOLD
IS";
820 PRINT"SITTING IN THE CAVE. YOU'LL BE RICH!
YOU";
822 PRINT"GET YOUR MOTHER HER OPERATION! YOU
824 PRINT"CAN BUY SHOES FOR YOUR SISTER! BOY AR
E YOU LUCKY!!!
826 N=15:E=15:S=15:W=17:NE=15:NW=21:SE=15:SW=23
:RETURN
828 PRINT"YOU ARE IN THE CAVE. THE EXIT IS BARR
ED
830 PRINT"BY HUGE STEEL BARS AND TRY AS YOU LIK
E
832 PRINT"YOU CANNOT GET THEM OPEN. NOW YOU'LL
834 PRINT"MISS YOUR WEDDING AND BRUNHILDE IS GO
ING";

```

```

836 PRINT"CRY VERY LOUDLY WHEN SHE FINDS OUT.
838 N=0:E=0:S=24:W=0:NE=0:NW=0:SE=0:SW=0:RETURN

E
840 PRINT"YOU ARE IN THE CAVES PROPER NOW. THERE
842 PRINT"SEEMS TO BE MANY DIFFERENT TUNNELS AB
OUT";
844 PRINT"HERE AND ONE LARGE ONE RUNS TO THE EA
ST.";
846 PRINT"OFCOURSE YOU DON'T HAVE TO LISTEN TO
ME YOU COULD ALWAYS TRY WEST.
848 N=23:E=28:S=25:W=26:NE=0:NW=0:SE=0:SW=0:RET
URN
850 PRINT"WELL LETS SEE... THERE IS A TUNNEL HE
RE
852 PRINT"THAT RUNS EAST/WEST AND THERE IS A SM
ALL";
854 PRINT"FISSURE BARELY MANAGEABLE THAT GOES T
O THE NORTH-EAST.
856 N=0:E=28:S=0:W=26:NE=24:NW=0:SE=0:SW=0:RETU
RN
858 PRINT"THE TUNNEL BRANCHES OFF HERE INTO TWO

860 PRINT"PATHS OTHER THAN THE ONE YOU JUST CAM
E THROUGH, CONFUSING ISN'T IT???"
862 N=24:E=25:S=27:W=0:NE=0:NW=0:SE=0:SW=0:RETU
RN
864 PRINT"THE TUNNEL HERE TURNS SLIGHTLY OFF
866 PRINT"COURSE. OH BY THE WAY, IF YOU WERE WO
ND-";
868 PRINT"ERING ABOUT HOW YOU COULD SEE IF YOU
ARE";
870 PRINT"THIRTY OR FORTY FEET UNDERGROUND... T
HE
872 PRINT"AIR IS FILLED WITH MAGIC AND IT GLOWS
ENOUGH FOR YOU TO SEE BY.
874 N=0:E=26:S=30:W=0:NE=0:NW=0:SE=0:SW=0:RETUR
N
876 PRINT"YOU ARE AT A SMALL INTERSECTION OF AB
OUT";
878 PRINT"FOUR PASSAGES. WELL EXACTLY FOUR PASS
-
880 PRINT"AGES. I'M NOT GOING TO TELL YOU WHICH
WAY THEY GO !!
882 N=24:E=0:S=31:W=25:NE=0:NW=0:SE=0:SW=29:RET
URN
884 PRINT"YOU ARE AT THE INTERSECTION OF TWO SM
ALL";
886 PRINT"PASSAGES. ONE LEADS NORTH-WEST AND TH
E
888 PRINT"OTHER LEAVES TO THE SOUTH. ON THE WAL

```

```

L
890 PRINT"IS SCRAWLED... 'DEATH TO THOSE WHO TA
KE
892 PRINT"THE SOUTH PASSAGE".
894 N=0:E=0:S=38:W=0:NE=0:NW=26:SE=0:SW=0:RETUR
N
896 PRINT"NOT 'MANY' BUT QUITE A FEW PASSAGES
898 PRINT"LEAVE THIS ROOM. CAN YOU FIND THEM??
900 N=27:E=0:S=0:W=0:NE=0:NW=0:SE=36:SW=37:RETU
RN
902 PRINT"THE ROOM HERE IS EXTREMELY DARK. I CA
N
904 PRINT"NOT TELL YOU ANYTHING EXCEPT THAT A
906 PRINT"LARGE ROCK HERE IS REPELLING THE MAGI
C
908 PRINT"THAT GIVES YOU LIGHT. YOU HAVE ALSO B
EEN";
910 PRINT"BUMPING INTO WALLS NOW FOR ABOUT FIFT
EENMINUTES.
912 N=0:E=0:S=33:W=28:NE=0:NW=0:SE=32:SW=0:RETU
RN
914 PRINT"AAA! YOU GOT YOURSELF OUT OF THAT LAS
T
916 PRINT"ROOM! GOOD! NOW YOU CAN FOLLOW THE NE
XT TUNNEL AS IT LEADS YOU ";
918 PRINT"SOUTHWARD.":N=0:E=0:S=34:W=0:NE=0:NW=
0:SE=0:SW=0:RETURN
920 PRINT"THE ROOM HERE IS FILLED WITH DUST. IT
922 PRINT"CLOGS YOUR EYES AND MAKES IT DIFFICUL
T
924 PRINT"TO SEE. FOLLOW ME TO THE NORTH OR TO
THE SOUTH-WEST AND WE'LL GET OUT.
926 N=31:E=0:S=0:W=0:NE=0:NW=0:SE=0:SW=35:RETUR
N
928 PRINT"A QUICK BEND IN THE PATH AND YOU NOW
HAVE TO TRAVEL A DIFFERENT WAY.
930 N=32:E=0:S=0:W=39:NE=0:NW=0:SE=0:SW=0:RETUR
N
932 PRINT"THESE TUNNELS SEEM ENDLESS DON'T THEY
?
934 PRINT"WELL YOU MIGHT BE SURPRISED AT WHAT Y
OU
936 PRINT"CAN FIND IN HERE. MAYBE IF YOU BREAK
THE";
938 PRINT"BIG MIRROR ON THE WEST WALL YOU'D FIN
D SOMETHING OF INTEREST BEHIND IT.
940 N=0:E=0:S=0:W=0:NE=33:NW=0:SE=0:SW=38:RETUR
N
942 PRINT"YOU HAVE ENTERED A SMALL CHAMBER OF
944 PRINT"SLEEPING ROCKS. THEY MUST BE SLEEPING

```



```

836 PRINT"CRY VERY LOUDLY WHEN SHE FINDS OUT.
838 N=0:E=0:S=24:W=0:NE=0:NW=0:SE=0:SW=0:RETURN

E
840 PRINT"YOU ARE IN THE CAVES PROPER NOW. THERE
842 PRINT"SEEMS TO BE MANY DIFFERENT TUNNELS AB
OUT";
844 PRINT"HERE AND ONE LARGE ONE RUNS TO THE EA
ST.";
846 PRINT"OFCOURSE YOU DON'T HAVE TO LISTEN TO
ME YOU COULD ALWAYS TRY WEST.
848 N=23:E=28:S=25:W=26:NE=0:NW=0:SE=0:SW=0:RET
URN
850 PRINT"WELL LETS SEE... THERE IS A TUNNEL HE
RE
852 PRINT"THAT RUNS EAST/WEST AND THERE IS A SM
ALL";
854 PRINT"FISSURE BARELY MANAGEABLE THAT GOES T
O THE NORTH-EAST.
856 N=0:E=28:S=0:W=26:NE=24:NW=0:SE=0:SW=0:RETU
RN
858 PRINT"THE TUNNEL BRANCHES OFF HERE INTO TWO

860 PRINT"PATHS OTHER THAN THE ONE YOU JUST CAM
E THROUGH. CONFUSING ISN'T IT???"
862 N=24:E=25:S=27:W=0:NE=0:NW=0:SE=0:SW=0:RETU
RN
864 PRINT"THE TUNNEL HERE TURNS SLIGHTLY OFF
866 PRINT"COURSE. OH BY THE WAY, IF YOU WERE WO
ND-";
868 PRINT"ERING ABOUT HOW YOU COULD SEE IF YOU
ARE";
870 PRINT"THIRTY OR FORTY FEET UNDERGROUND... T
HE
872 PRINT"AIR IS FILLED WITH MAGIC AND IT GLOWS
ENOUGH FOR YOU TO SEE BY.
874 N=0:E=26:S=30:W=0:NE=0:NW=0:SE=0:SW=0:RETUR
N
876 PRINT"YOU ARE AT A SMALL INTERSECTION OF AB
OUT";
878 PRINT"FOUR PASSAGES. WELL EXACTLY FOUR PASS
-
880 PRINT"AGES. I'M NOT GOING TO TELL YOU WHICH
WAY THEY GO !!
882 N=24:E=0:S=31:W=25:NE=0:NW=0:SE=0:SW=29:RET
URN
884 PRINT"YOU ARE AT THE INTERSECTION OF TWO SM
ALL";
886 PRINT"PASSAGES. ONE LEADS NORTH-WEST AND TH
E
888 PRINT"OTHER LEAVES TO THE SOUTH. ON THE WAL

```

```

L
890 PRINT"IS SCRAWLED... 'DEATH TO THOSE WHO TA
KE
892 PRINT"THE SOUTH PASSAGE".
894 N=0:E=0:S=38:W=0:NE=0:NW=26:SE=0:SW=0:RETUR
N
896 PRINT"NOT 'MANY' BUT QUITE A FEW PASSAGES
898 PRINT"LEAVE THIS ROOM. CAN YOU FIND THEM??"
900 N=27:E=0:S=0:W=0:NE=0:NW=0:SE=36:SW=37:RETU
RN
902 PRINT"THE ROOM HERE IS EXTREMELY DARK. I CA
N
904 PRINT"NOT TELL YOU ANYTHING EXCEPT THAT A
906 PRINT"LARGE ROCK HERE IS REPELLING THE MAGI
C
908 PRINT"THAT GIVES YOU LIGHT. YOU HAVE ALSO B
EEN";
910 PRINT"BUMPING INTO WALLS NOW FOR ABOUT FIFT
EENMINUTES.
912 N=0:E=0:S=33:W=28:NE=0:NW=0:SE=32:SW=0:RETU
RN
914 PRINT"AAA! YOU GOT YOURSELF OUT OF THAT LAS
T
916 PRINT"ROOM! GOOD! NOW YOU CAN FOLLOW THE NE
XT TUNNEL AS IT LEADS YOU ";
918 PRINT"SOUTHWARD.":N=0:E=0:S=34:W=0:NE=0:NW=
0:SE=0:SW=0:RETURN
920 PRINT"THE ROOM HERE IS FILLED WITH DUST. IT
922 PRINT"CLOGS YOUR EYES AND MAKES IT DIFFICUL
T
924 PRINT"TO SEE. FOLLOW ME TO THE NORTH OR TO
THE SOUTH-WEST AND WE'LL GET OUT.
926 N=31:E=0:S=0:W=0:NE=0:NW=0:SE=0:SW=35:RETUR
N
928 PRINT"A QUICK BEND IN THE PATH AND YOU NOW
HAVE TO TRAVEL A DIFFERENT WAY.
930 N=32:E=0:S=0:W=39:NE=0:NW=0:SE=0:SW=0:RETUR
N
932 PRINT"THESE TUNNELS SEEM ENDLESS DON'T THEY
?
934 PRINT"WELL YOU MIGHT BE SURPRISED AT WHAT Y
OU
936 PRINT"CAN FIND IN HERE. MAYBE IF YOU BREAK
THE";
938 PRINT"BIG MIRROR ON THE WEST WALL YOU'D FIN
D SOMETHING OF INTEREST BEHIND IT.
940 N=0:E=0:S=0:W=0:NE=33:NW=0:SE=0:SW=38:RETUR
N
942 PRINT"YOU HAVE ENTERED A SMALL CHAMBER OF
944 PRINT"SLEEPING ROCKS. THEY MUST BE SLEEPING

```

```

946 PRINT"SINCE THEY ARE NOT MOVING. TIP-TOE
    QUIETLY OUT AND YOU WON'T WAKE";
948 PRINT" THEM. ":N=0:E=0:S=41:W=0:NE=29:NW=30:
SE=0:SW=0:RETURN
950 PRINT"YOU ARE IN A SMALL EVEN WALLED CAVE R
OOM";
952 PRINT"A FEELING OF DIZZINESS CONFUSES YOUR
    TRAVELS. "
954 N=36:E=36:S=42:W=0:NE=36:NW=30:SE=0:SW=0:RE
TURN
956 PRINT"WELL... THE MESSAGE IN THE LAST ROOM
958 PRINT"MUST HAVE BEEN MISTAKEN... SINCE YOU
ARE";
960 PRINT"STILL ALIVE. WELL ONE CAN'T BELIEVE A
NY
962 PRINT"GRAFITTI THEY READ ON WALLS CAN THEY?

964 N=0:E=0:S=0:W=0:NE=35:NW=0:SE=0:SW=41:RETUR
N
966 PRINT"THREE PASSAGES LEAVE HERE. ONE TO THE

968 PRINT"NORTH... ONE TO THE WEST... AND ONE T
O THE SOUTH. WHICH WAY WILL";
970 PRINT" YOU TAKE?":N=34:E=0:S=40:W=33:NE=0:N
W=0:SE=0:SW=0:RETURN
972 PRINT"FOUR PATHS LEAVE HERE. LOUD MAJESTIC
    SINGING SEEMS TO BE COMI";
974 PRINT"NG FROM NEARBY. COULD THIS BE A SIGN?
    COULD IT LEAD OUT OF HERE?";
976 PRINT" COULD YOU BE GOING HOME???"":N=35:E=3
9:S=0:W=41:NE=0:NW=0:SE=0:SW=48:RETURN
978 PRINT"THE PASSAGE TURNS A CORNER. THAT'S AL
L THIS ROOM RATES AS A DESCRIPTION.
980 N=36:E=0:S=0:W=0:NE=38:NW=0:SE=0:SW=0:RETUR
N
982 GOSUB 1258:N=37:E=41:S=41:W=43:NE=41:NW=0:S
E=41:SW=44:RETURN
984 GOSUB 1258:N=42:E=42:S=41:W=0:NE=0:NW=0:SE=
41:SW=44:RETURN
986 GOSUB 1258:N=43:E=43:S=0:W=0:NE=43:NW=45:SE
=0:SW=0:RETURN
988 GOSUB 1258:N=44:E=43:S=44:W=44:NE=46:NW=44:
SE=44:SW=0:RETURN
990 GOSUB 1258:N=45:E=43:S=43:W=44:NE=47:NW=44:
SE=43:SW=45:RETURN
992 GOSUB 1258:N=67:E=43:S=43:W=46:NE=43:NW=46:
SE=43:SW=46:RETURN
994 PRINT"YOU ARE AT THE ENTRANCE TO A LARGE CL
EAN";
996 PRINT"CASTLE. IT IS THE HOME OF THE ALL POW

```

```

ER="";
  998 PRINT"FUL ALL KNOWING ALL KIND ALL DRUNK
  1000 PRINT"MAGICIAN RALPH. HE ALONE HAS THE POW
ER TO GET YOU HOME TO SWEET BRUN";
  1002 PRINT"HILDE." :N=40:E=0:S=49:W=0:NE=0:NW=0:
SE=0:SW=0:RETURN
  1004 PRINT"YOU ARE AT THE NORTH END OF THE GREA
T HALL. FINE PICTURES AND ";
  1006 PRINT"OIL PAINTINGS ARE HUNG FROM THE WA
LLS. THIS MAGICIAN SEEMS TO";
  1008 PRINT" HAVE DONE WELL FOR HIMSELF." :N=48:E
=0:S=50:W=0:NE=0:NW=0:SE=0:SW=0
  1010 RETURN
  1012 PRINT"YOU ARE IN THE MIDDLE OF THE HALL. A
DOOR TO YOUR RIGHT LEADS ";
  1014 PRINT"INTO THE BLUE ROOM AND ONE TO YOUR
LEFT TO THE GREEN ROOM." :N=49
  1016 E=51:S=53:W=52:NE=0:NW=0:SE=0:SW=0:RETURN
  1018 PRINT"YOU ARE IN THE BLUE ROOM. A GREAT SE
NSE OF DEEPRESSION FLOWS LIKE";
  1020 PRINT" WAVES THROUGH YOU. YOU LONG FOR HOM
E AND YOUR SWEET BRUNHILDE";
  1022 PRINT". HOW YOU WISH YOU WERE THERE." :N=0:
E=0:S=0:W=50:NE=0:NW=0:SE=0
  1024 SW=0:RETURN
  1026 PRINT"YOU ARE IN THE GREEN ROOM. YOU ARE S
UID-ONLY VERY JEALOUS OF RAL";
  1028 PRINT"PH AND HIS FINE CASTLE. HOW COME HE
RATES ALL THIS STUFFAND YOU ";
  1030 PRINT"LIVE IN A HOLE IN THE GROUND?!?" :N=0
:E=50:S=0:W=0:NE=0:NW=0:SE=0
  1032 SW=0:RETURN
  1034 PRINT"YOU ARE NOW AT THE SOUTH END OF THE
GREAT HALL. HALLWAYS LEA";
  1036 PRINT"VE TO THE NORTH AND TO THE EAST AN T
O THE WEST." :N=50:E=54:S=0
  1038 W=61:NE=0:NW=0:SE=0:SW=0:RETURN
  1040 PRINT"YOU ARE IN THE MIDDLE OF THE EAST HA
LL. A RAT SCURRIES PAST YOU ";
  1042 PRINT"AND TAKES A NIP AT YOUR LEG. IN AN I
NSTANT HE IS GONE WITH A S";
  1044 PRINT"SNEER AND A TWITCH OF HIS TAIL." :N=0
:E=55:S=0:W=53:NE=0:NW=0:SE=0
  1046 SW=0:RETURN
  1048 PRINT"YOU ARE AT THE END OF THE EAST HALL.
THEONLY NEW PASSAGE IS A SM";
  1050 PRINT"ALL SERVANTS EXIT TO THE SOUTH-EA
ST." :N=0:E=0:S=0:W=54:NE=0:NW=0
  1052 SE=52:SW=0:RETURN
  1054 PRINT"YOU ARE IN THE SERVANT'S AREA. SINCE
RALPHS LAST SERVANT DIED";

```

1056 PRINT "NINE-HUNDRED YEARS AGO THE PLACE
 IS A MESS AND ALSO SMELLS P";
 1058 PRINT "RETTY BAD." :N=0:E=0:S=0:W=57:NE=55:N
 W=0:SE=0:SW=0:RETURN
 1060 PRINT "YOU ARE OUTSIDE THE SERVANTS QUARTER
 S. A DISTRESSING SMELL COME";
 1062 PRINT "S FROM THE EAST. ITS BEST TO GO BACK
 TO THE WEST FOR YOURNOSES SAKE.
 1064 N=0:E=56:S=0:W=58:NE=0:NW=0:SE=0:SW=0:RETU
 RN
 1066 PRINT "YOU ARE IN THE MIDDLE OF THE SOUTH H
 ALL. AN EERIE LIGHT COMES FROM";
 1068 PRINT "THE WEST AND GREAT MUSIC FILLS THE
 AIR." :N=0:E=57:S=0:W=59:NE=0
 1070 NW=0:SE=0:SW=0:RETURN
 1072 PRINT "YOU ARE AT THE END OF THE SOUTH HALL
 . TWO SMALL PASSAGES LEAVE ";
 1074 PRINT "HERE. ONE TO THE NORTH-WEST AND ON
 E TO THE SOUTH-WEST":N=0:E=58
 1076 S=0:W=0:NE=0:NW=60:SE=0:SW=62:RETURN
 1078 PRINT "YOU ARE AT THE END OF THE WEST HALL.
 A SMALL PASSAGE LEADS TO TH";
 1080 PRINT "E SOUTH-WEST." :N=0:E=61:S=0:W=0:NE=0
 :NW=0:SE=0:SW=59:RETURN
 1082 PRINT "YOU ARE IN THE MIDDLE OF THE WEST HA
 LL. A LARGE HOLE IN THE FLOO";
 1084 PRINT "R REVEALS A HUGEDROP TO THE GROUND H
 UNDREDS OF FEET
 1086 PRINT "BELOW. BETTER NOT GO DOWN..." :N=0:E=53
 :S=0:W=60:NE=0:NW=0:SE=0
 1088 SW=0:RETURN
 1090 PRINT "YOU ARE IN A WINDING PASSAGE. AS YOU
 R TRAVEL PROGRESSES THE EER";
 1092 PRINT "IE MUSIC GETS LOUDER AND YOU GET CO
 LDER BUT THIS COULDBE THE WAY";
 1094 PRINT "OUT!!!" :N=0:E=59:S=0:W=0:NE=0:NW=63
 :SE=0:SW=0:RETURN
 1096 PRINT "YOU ARE IN A CONNECTING HALLWAY WITH
 SOME LOUD ALBEIT EERIE MU";
 1098 PRINT "SIC EMANATING FROM THE SOUTH-WEST."
 :N=0:E=0:S=0:W=0:NE=62:NW=0
 1100 SE=0:SW=64:RETURN
 1102 PRINT "YOU ARE IN A SMALL ANTE-CHAMBER. A L
 ARGEWELL USED PASSAGE LEADS ";
 1104 PRINT "TO THE NORTH ANDA SMALLER PASSAGE TO
 THE WEST. EERIE MUSIC COM";
 1106 PRINT "ES FROM THE NORTH." :N=65:E=0:S=0:W=66
 :NE=0:NW=0:SE=63:SW=0:RETURN
 1108 PRINT "YOU OPEN THE DOOR AND ENTER INTO....
 . RALPHS BATHROOM. A RADIO ";

```

1110 PRINT"TUNED TO ONE OF THE MORE POPULAR ROCK
STATIONS IS BLAST-ING FORTH ";
1112 PRINT"PUNK ROCK MUSIC. THE EFFECT IT HAS ON
THE CASTLES WALLS IS EERIE ";
1114 PRINT"INDEED":N=0:E=0:S=64:W=0:NE=0:NW=0:S
0:SW=0:RETURN
1116 PRINT"YOU ARE IN THE LIBRARY WHERE THE GOO
MAGICIAN RALPH IS PORING";
1118 PRINT" OVER AN OLD AND ANCIENT TOME. THE
LORD YOBYALP SHOWS THROUGH TH";
1120 PRINT"HE COVER FROM THE LIGHT BEHIND. RALPH
DEEPLY ABSORBED IN THE BOOK.
1122 N=0:E=64:S=0:W=0:NE=0:NW=0:SE=0:SW=0:RETUR

1124 PRINT"YOU ARE AT THE EDGE OF THE EVIL WITC
S LAND. THE EVIL PRUDENCE ";
1126 PRINT"DOES NOT SEEM TO BE ANYWHERE AROUND.
AT IS GOOD FOR YOU SINCE SH";
1128 PRINT"HE COULD TURN YOU INTO AN UGLIER TOAD
HAN YOU ALREADY ARE." :N=68
1130 E=47:S=73:W=0:NE=0:NW=0:SE=0:SW=0:RETURN
1132 PRINT"WALKING NORTH YOU WILL FIND A TWISTI
OLD TRAIL. TO THE WEST ";
1134 PRINT"LIES CONFUSION. SOUTH IS THE WAY TO
O FOR A FAST EXIT FROM THIS";
1136 PRINT" LAND. GO THERE IF YOU VALUE YOUR
FE." :N=69:E=0:S=67:W=70:NE=0
1138 NW=0:SE=0:SW=0:RETURN
1140 PRINT"THERE IS A TINY LITTLE HIDDEN TWISTI
SECRET OLD OVERGROWN DIRTY";
1142 PRINT" SNEAKY TRAIL THAT LEADS WEST FROM H
E. SOUTH IS MUCH NICER BUT ";
1144 PRINT"NOT A LOT OF FUN. A ROCK HERE SAYS
IS WAY TO THE TLHTSOODST";
1146 PRINT" AND IT POINTS WEST":N=0:E=0:S=68:W=7
NE=0:NW=0:SE=0:SW=0:RETURN
1148 PRINT"YOU ARE IN A SMALL CLEARING. A PAPER
LIES NEARBY. IT READS ";
1150 PRINT"CONFUSION SAYS- EXPLORER WHO GETS
T... IS VERY LUCKY INDEED..."
1152 N=70:E=70:S=70:W=70:NE=70:NW=70:SE=70:SW=7
RETURN
1154 PRINT"YOU ARE IN THE MIDDLE OF A LONG BORI
NORTH-SOUTH PATHWAY.";
1156 PRINT" YOUR HEAD IS SLIGHTLY ACHING.
N=72:E=0:S=70:W=0:NE=0:NW=0
1158 SE=0:SW=0:RETURN
1160 PRINT"YOU ARE IN THE SNEAKY CLEARING. THER
ARE THREE EXITS FROM HERE.";
1162 PRINT" ONE THAT GOES TO THE SOUTH... ONE TO

```

THE SOUTH-WEST...AND A DIRTY";
 1164 PRINT" SNEAKY....LITTLE TRAIL TO THE NOR
 TH.":N=69:E=0:S=71:W=0:NE=0:NW=0
 1166 SE=0:SW=76:RETURN
 1168 PRINT"A SMALL CLEARING AT THE SOUTHERN EDG
 E OFTHE EVIL PRUDENCE'S LAND ";
 1170 PRINT"GIVES YOU ROOM TO REST AND RELAX. TW
 O PATHS LEAD AWAYFROM HERE.
 1172 N=70:E=0:S=0:W=0:NE=0:NW=77:SE=0:SW=0:RETU
 RN
 1174 PRINT"AN INTERSECTION OF THREE PATHS IS HE
 RE. YOUR HEAD ACHES SLIGHTLY ";
 1176 PRINT"AS YOU APPROACHTHE AREA. YOU CAN LEA
 VE WITH MY BLESSINGIF YOU FIND";
 1178 PRINT" THE WAY OUT.":N=75:E=0:S=78:W=0:NE=
 70:NW=0:SE=0:SW=0:RETURN
 1180 PRINT"THE PATH'S ORIENTATION BECOMES SUBTL
 Y ALTERED AS ROUND YON CORNER. ";
 1182 PRINT"WATCH OUT FOR THE BAT GUANO TO YOUR
 LEFT.":N=0:E=0:S=74:W=0:NE=0
 1184 NW=79:SE=0:SW=0:RETURN
 1186 PRINT"THE MIDPOINT FOR A LONG SLOPING CORR
 IDORIS WHERE YOU ARE. NOT MUCH";
 1188 PRINT" SPECIAL ABOUTTHIS PLACE EXCEPT THE
 SMELL OF BAT GUANO":N=0:E=0:S=0:W=0
 1190 NE=72:NW=0:SE=0:SW=79:RETURN
 1192 PRINT"YOU ARE AT THE MIDPOINT OF A LONG LO
 NG SLOPING CORRIDOR LEADING ";
 1194 PRINT"SLIGHTLY SOUTH.":N=0:E=0:S=0:W=0:NE=
 74:NW=81:SE=73:SW=0:RETURN
 1196 PRINT"THE PATH HERE IS LITTERED WITH THE F
 RESHREMAINS OF BATS AND TOADS. ";
 1198 PRINT" THEIR BODIES ARE STILL DECAYING IN
 THE DIRT.":N=0:E=0:S=74:W=82:NE=0
 1200 NW=0:SE=0:SW=0:RETURN
 1202 PRINT"YOU ARE AT THE INTERSECTION OF THREE
 TRAILS. ONE TO THE WEST ";
 1204 PRINT"AND ONE TO THE NORTH/EAST AND ONE
 TO THE SOUTH/EAST. WHEN YOUR ";
 1206 PRINT"COMPASS WORKS... IT SURE IS A HANDY
 THING TO HAVE.":N=0:E=0:S=0:W=83
 1208 NE=76:NW=0:SE=75:SW=0:RETURN
 1210 PRINT"YOU ARE OVERCOME BY THE FEELING OF
 NAUSEA. AS YOU TRY TO ALIGN ";
 1212 PRINT"YOURSELF ON YOUR COMPASS YOU ROCK AN
 D REEL BACK AND FORTH AND FIND";
 1214 PRINT" IT DIFFICULT TO WALK IN A STRAIGHT L
 INE. WHAT COULD BE CAUSING
 1216 PRINT"THIS TO HAPPEN???" :N=71:E=72:S=79:W=
 77:NE=69:NW=84:SE=76:SW=82:RETURN
 1218 PRINT"AT THE SOUTHWEST EDGE OF THE WITCHES

```

    LAND THE TUNNEL TURNS SLIGHTLY.");
1220 PRINT" EITHER THAT OR WALK INTO THE HUGE
ROCK THAT IS IN ITS WAY.":N=82:E=0
1222 S=0:W=0:NE=0:NW=0:SE=77:SW=0:RETURN
1224 PRINT"OOPS!!! A SHARP TURN FORCES YOU TO G
O EITHER SOUTH OR EAST DEPENDING ";
1226 PRINT"ON WHICH WAY YOU CAME FROM.":N=0:E=7
8:S=81:W=0:NE=0:NW=0:SE=0:SW=0:RETURN
1228 PRINT"A ROOM HERE HAS AN EASTERN EXIT AND
ALSO A NORTHERN ONE. FROM THE NORTH ";
1230 PRINT"COMES A FUNNY FEELING OF DIZZINESS.
":N=80:E=79:S=0:W=0:NE=0:NW=0:SE=0
1232 SW=0:RETURN
1234 PRINT"YOU ARE AT THE EVIL WITCH PRUDENCES
OLDEBLACKSMITHE SHOPPE. HERE WITH ";
1236 PRINT"HER BLASTEFURNACEE IS WHERE SHE FORG
ES HER HORSES SHOES FOR HER HUNTIN";
1238 PRINT"G RAIDES. THE HEAT FROM THE FURNACE
BURNES TO YOUR SOUL.":N=0:E=0:S=0
1240 W=0:NE=80:NW=0:SE=0:SW=0:RETURN
1242 PRINT"I CAN'T LET YOU DO IT AGAIN. YOU HAV
EN'T DONE ANYTHING YET!!!!":GOTO154
1244 PRINT"*** BANG *** !!! AS YOU ENTER HERE,
THE GOLD DISAPPEARS IN A ";
1246 PRINT"FLASH, AND A HUGE SET OF STEEL BARS
DROPS INTO PLACE LOCKING";
1248 PRINT" YOU INTO THE CAVES.":PRINT:GOTO 292

1250 PRINT"YOUR HANDS?? YOU HAVE TO FIND A CONT
AINER.":RETURN
1252 PRINT"AUURRRGGGHHHHH!!! SCREAMS THE WITCH
!!! IN HER HASTE, THE WITCH";
1254 PRINT" FALLS INTO THE BLAST FURNACE!! HELP
!! I'M SMELTING... ";
1256 PRINT"SMELTING... IS THE LAST SHE BREATHES
.":GOTO364
1258 PRINT"THE TUNNELS IN HERE ARE VERY CONFUSI
NG. YOU WILL NEVER GET OUT ALIVE."
1260 RETURN
1262 PRINT"YOU SLIPPED AND FELL ON THE BANANA P
EEL.WHY NOT TRY AGAIN?":RETURN
1264 PRINT"AREN'T YOU ??? WELL THAT'S ALL YOU N
EED TO KNOWABOUT HOW YOU ARE";
1266 PRINT" DOING.":RETURN
1268 PRINT" KILL THE "NO$:RETURN
1270 PRINT"THE FOREST IS VERY DARK. MOST TRAILS

1272 PRINT"SEEM ALIKE AND YOU CANNOT GET YOUR
BEARINGS IN HERE.":RETURN

```


1274 PRINT" UP AND NOTICE YOU IN THEIR REALM.
THEY STONED YOU TO DEATH."
1276 EOG=TRUE:RETURN

READY.

INDEX

STARTREK	10
An adventure program in which you are in charge of the Star Ship Enterprise.	
HIGH RESOLUTION GRAPHICS	44
An explanation of how to produce Hi-Res graphics, and Machine Code routines to use the methods.	
MULTI-COLOUR BARCHART	88
Displays a barchart with a 3D effect in multi-colour mode.	
3D GRAPH	91
Displays a graph with a 3D effect in standard Hi-Res mode	
MULTI-COLOUR PATTERN	93
Produces an interesting effect from the cross-over of lines in Multi-colour mode.	
SINE GRAPH	95
Plots a sine graph in standard Hi-Res mode.	
DEMO OF HI-RES ROUTINES 5 & 7	97
Demonstrates the use of the routines to switch between the Hi-Res screen and the Text screen.	
DIGICLOCK	100
A program demonstrating the time-keeping facility of the CBM 64. Also demonstrates Sprites.	
DISASSEMBLER	105
Allows you to disassemble any part of memory.	
HANGMAN	109
An intelligent version of the children's game, which would be useful as a teaching/learning aid.	
LANDMINE	116
Cross the minefield without getting blown up.	
FRUIT MACHINE	122
This game is a simple version of a fruit machine.	
CAR	131
Try to drive the car around a race track without crashing.	
MAZE	135
A maze game in which you are timed from the entrance to the exit.	

KEYBOARD SYNTHESIZER	139
Enables you to make full use of the CBM 64's Sound Interface Device.	
RESERVED WORD CHANGER	146
This program allows you to modify the Basic commands of the CBM 64.	
+/- LISTER	150
A useful program that lists just one line at a time either before or after the previous line.	
LARGE CHARACTERS	152
Provides examples of how large characters can be produced on the CBM 64.	
USER DEFINED CHARACTERS	156
Allows you to plot points on a user defined character area.	
MACHINE CODE TO DATA	159
A useful program which enables you to convert memory into data statements.	
PERSONAL INFORMATION RETRIEVAL	161
Can be used as a diary or address book or both.	
WILL O' THE WISP	168
An adventure game in which you must do certain things before you are wed.	

DUCKWORTH HOME COMPUTING

All books written by Peter Gerrard, former editor of *Commodore Computing International*, author of two top-selling adventure games for the Commodore 64, or by Kevin Bergin. Both are regular contributors to *Personal Computer News*, *Which Micro?* and *Software Review*.

THE COMPLETE 64 ROM DISASSEMBLY

by Peter Gerrard and Kevin Bergin

This book is for anyone who has ever wondered how the Commodore 64 really works. Intended for the serious programmer, it includes fundamental memory maps, memory architecture maps, the disassembly itself and (for reference) the complete 6510 machine code instruction set. £5.95

ADVANCED BASIC & MACHINE CODE FOR THE 64

by Peter Gerrard

For the more serious user of the Commodore 64, this book teaches you all about programming in Machine Code, with sections on double precision arithmetic and animation, along with a series of chapters on using the special features of the Commodore 64.

The all-important link to Basic is not forgotten, and the opening chapters form a guide to improving your Basic programming techniques, along with many program examples. £6.95

Other titles in the series include *Using the 64*, *12 Simple Electronic Projects for the VIC*, *Will You Still Love Me When I'm 64*, *Advanced Basic & Machine Code Programming on the VIC*, as well as *Pocket Handbooks for the VIC, 64, Dragon, Spectrum and BBC Model B*.

Write in for a descriptive leaflet (with details of cassettes).



DUCKWORTH

The Old Piano Factory, 43 Gloucester Crescent, London NW1 7DY
Tel: 01-485 3484

Duckworth Home Computing

CBM-64 PROGRAMS 1

Edited by Nick Hampshire

This book provides the reader with useful and interesting programs for the 64. Topics covered include — Hi Res Graphics — Music — Games — Utilities — Sprites and User Defined Characters — Functional Programs, etc. There is a detailed explanation of Hi Res Graphics and the necessary machine code routines to implement them, along with demonstration programs. An exciting version of Star Trek is also included as well as a full length adventure game. Among the functional programs is a Personal Information Retrieval package which enables you to create and manipulate up to 365 records.

Nick Hampshire is the publisher of *Commodore Computing International*.

ISBN 0-7156-1882-2



9 780715 618820

Duckworth
The Old Piano Factory
43 Gloucester Crescent, London NW1

ISBN 0 7156 1882 2

IN UK ONLY £6.95 NET