

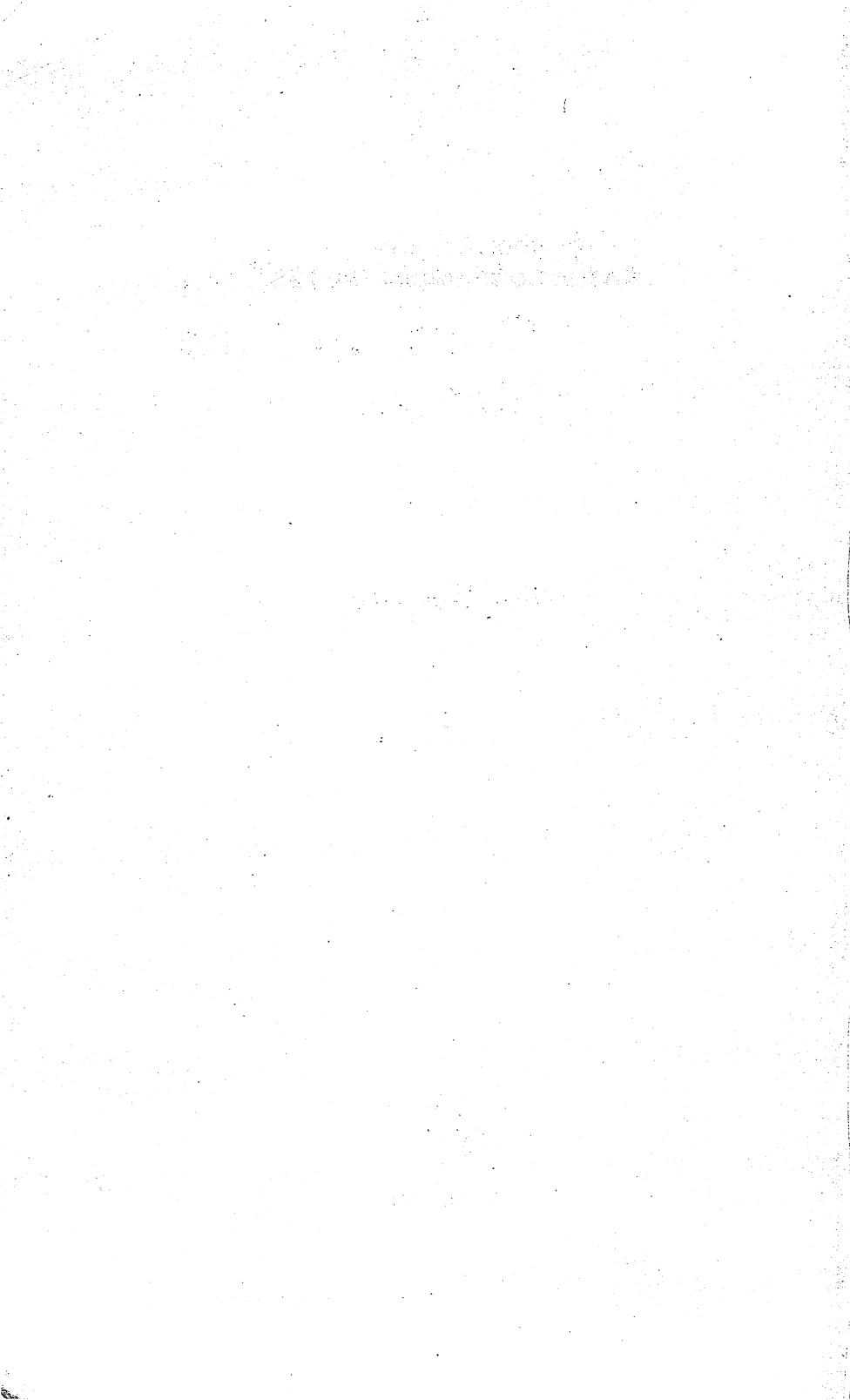
FURTHER ADVENTURES

on the
Commodore 64/128



Peter Gerrard

**Further Adventures
on the Commodore 64/128**



Further Adventures on the Commodore 64/128

Peter Gerrard



Duckworth

First published in 1986 by
Gerald Duckworth & Co. Ltd.
The Old Piano Factory
43 Gloucester Crescent, London NW1

© 1986 Peter Gerrard

All rights reserved. No part of this publication
may be reproduced 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 1859 8

British Library Cataloguing in Publication Data

Gerrard, Peter

Further adventures on the Commodore 64/128

**1. Computer games 2. Commodore 64/128
(Computer)**

I. Title

794.8'0928'54165

GV1469.2

ISBN 0-7156-1859-8

Typeset by The Electronic Village, Richmond
from text stored on a Commodore 64
Printed in Great Britain by
Redwood Burn Ltd., Trowbridge

Contents

Preface	7
1. Introduction	9
2. Graphics and Sound	13
3. Tape or Disk?	60
4. Useful Routines	72
5. Getting Started	99
6. Parsers	112
7. Text Compression	147
8. A Tape-based Adventure	158
9. A Disk-based Adventure	194
10. A Compiler	270
11. Role-playing Games	311
12. Conclusion	321
Index	325

Appendix

The following table shows the results of the experiments conducted on the 10th of May 1900. The results are given in the form of a table, the columns of which are headed by the names of the experiments, and the rows by the names of the experiments.

The results of the experiments conducted on the 10th of May 1900 are given in the following table. The columns of the table are headed by the names of the experiments, and the rows by the names of the experiments.

The results of the experiments conducted on the 10th of May 1900 are given in the following table. The columns of the table are headed by the names of the experiments, and the rows by the names of the experiments.

The results of the experiments conducted on the 10th of May 1900 are given in the following table. The columns of the table are headed by the names of the experiments, and the rows by the names of the experiments.

The results of the experiments conducted on the 10th of May 1900 are given in the following table. The columns of the table are headed by the names of the experiments, and the rows by the names of the experiments.

The results of the experiments conducted on the 10th of May 1900 are given in the following table. The columns of the table are headed by the names of the experiments, and the rows by the names of the experiments.

The results of the experiments conducted on the 10th of May 1900 are given in the following table. The columns of the table are headed by the names of the experiments, and the rows by the names of the experiments.

The results of the experiments conducted on the 10th of May 1900 are given in the following table. The columns of the table are headed by the names of the experiments, and the rows by the names of the experiments.

The results of the experiments conducted on the 10th of May 1900 are given in the following table. The columns of the table are headed by the names of the experiments, and the rows by the names of the experiments.

The results of the experiments conducted on the 10th of May 1900 are given in the following table. The columns of the table are headed by the names of the experiments, and the rows by the names of the experiments.

Preface

The first book in the 'Exploring Adventures' series was published in 1983. The series proved a great success and started many people off on the adventure-writing trail. Since then scores of people have written to me about the books, and a common thread running through most of the letters has been the question: 'When is the next one coming out?'

Well, here it is. Using the first book, *Exploring Adventures on the 64*, you could produce reasonable text-only adventures in Basic. By the time you've finished reading through this one, you should be in a position to produce text, graphics or role-playing games, using tape or disk as a storage medium. Generally you should be able to improve dramatically on those earlier games.

Although much of what I've got to say is still in Basic, the introduction of a number of machine-code routines speeds things up considerably, and the use of a compiler will enable you to write games with lightning-fast response times.

By taking a long look at text compression, room descriptions can become lengthier, and an improved parser (in either Basic or machine code: both have their advantages) now provides for Infocom-like complexity.

Many thanks to everyone who has written in with suggestions for this book, and thanks also to those people who took the time and trouble to write their own adventures using the techniques in *Exploring Adventures on the 64* and let me have a look at them, in particular Andrew Cray, Ray Davies, Jack Knight and John Ryan.

A dedication? To anyone who writes or plays adventure games, and M.U.D. in the eye to anyone who doesn't.

P.G.

1

Introduction

While many of you will probably have read the first book in this series of adventure-writing guides, a fair number will have neither seen nor heard of it. So if you old timers will bear with me for a while, let's take a look back over the last few years, and trace the progress of what has become one of the biggest selling types of computer game.

The history of adventure games

By now just about everyone must know something of the history of the first-ever 'real' adventure game. This, of course, was 'Colossal Cave Adventure' by Crowther and Woods, which ran in about 300K-worth of Fortran on a DEC PDP-10.

How anyone could write a text adventure game in Fortran is beyond me. I had enough trouble using Fortran at university writing scientific programs to try to predict the rate of decay of Wolf-Rayet stars. Even printing out a table of results was hard work.

Crowther and Woods, however, did manage it, setting a standard that has yet, in most people's opinion, to be bettered. It was a great improvement over the simple 'Wumpus' type of game that went before it, in which all you had to do was to explore a very small number of interconnecting caves, hoping to find an arrow and shoot the Wumpus before being attacked by bats or sailing down over the edge of a precipice.

In 'Colossal Cave' the player had to find and manipulate many different objects, explore over a hundred different locations, and solve many ingenious problems before being carried off into the sunset by hordes of cheering elves.

Little did Crowther and Woods know what they were letting the rest of the computer world in for. Before long Scott Adams had crammed

'Adventureland' (his first adventure) into 16K of TRS-80 computer, and then the adventure writing and playing world really took off.

'Zork', from Infocom, was the next great adventure game, running on three disks for a variety of computers - another superb game, with a variety and richness of prose that has seldom (if ever) been bettered. 'The Hobbit', from Melbourne House, was the first of a new breed of graphical adventures, with simple pictures showing each location, instead of vast reams of text describing them.

Which type of adventure you prefer is obviously up to you. To be honest, I don't particularly like graphical games, as I prefer to picture the scene for myself, but if that's the type of game you like, you play and write them!

'Zork' and the disk-based version of 'The Hobbit' are probably the two 'state-of-the-art' adventure games in the text and graphical fields respectively. 'Valhalla', from Legend, started a new trend towards animated adventures, and there followed a whole host of so-called arcade adventures, with the gifted software house Ultimate leading the field here.

To my mind these are not adventure games at all, and so they'll receive no further mention in this book. If you want to play them (and even I'll admit to dabbling with a joystick every now and again) that's fine, but you'll have to ask Ultimate how to write them: I'm not going to tell you!

Finally, we have the great game 'M.U.D.', short for 'Multi-User Dungeon', a vast adventure game which many different people can play at once. Since the micro on your desk is not likely to match the mainframe computer at Essex University used to play this game, there's not much point talking about how it was written, but if you ever get the chance to play it, grab it. You'll need a telephone (preferably someone else's), a modem, and access to the mainframe, but it's worth having a dabble. Just the shock of finding other people in your adventure world makes it stand out from other games. However, don't expect those other people to be nice to you.

Aims of this book

In the 'Exploring Adventures' series no knowledge of Basic or machine code was assumed, and we started right from scratch, with

lessons in Basic before going on to the nitty-gritty of writing adventure games.

In this book I assume that you are reasonably competent in Basic programming, but not necessarily in machine code. All the machine-code routines will be presented in their machine-code form, if you happen to own an assembler, or as a Basic loader, so you can just type them in and use them. You'll also be told how to adapt them to your own adventures.

The only exception to this rule will be the Basic compiler, which will be shown as a hexadecimal memory dump, because it's such a vast program. We'll worry more about that in Chapter 10.

Back in the 'Exploring Adventures' days, the programs that could be produced using the techniques described were not of earth-shattering proportions, although with a little thought and ingenuity you could succeed in producing a quite large and interesting game. The three things that were 'wrong' with this approach were:

- 1) You couldn't produce any graphics.
- 2) It could only handle a VERB NOUN input.
- 3) The response time was not as fast as one might like.

Moreover, it was pretty difficult to produce lengthy room descriptions and have a large vocabulary. All these problems will be solved in this book.

Since many Commodore 64 owners must by now own disk drives, it would be a shame to leave them out, and so we'll be concentrating on them as much as on tapes, although tape-only users will still be able to use most of what is included here.

Some (a very few) routines will be device-specific, so apologies in advance if there are any that you can't use on your particular set-up. The only one that may present major problems is the compiler program, since that is most definitely disk-only. Sorry about that, tape users, but I couldn't find a compiler that would work on tape - everything seems to require a disk drive, which really is not altogether surprising.

Still, you could always borrow a friend's disk drive!

Conclusion

Unlike the 'Exploring Adventures' series, which explained everything for the beginner (and was inevitably too slow for some people), in this book I'll assume that you want to get on with writing adventures, that you know a reasonable amount of Basic, and you want the quickest results in the shortest space of time. You won't be able to produce an adventure in a weekend, but, using the techniques in this book, it shouldn't take you too long.

So, without further ado, let's look at how we might use graphics and sound in an adventure game.

2

Graphics and Sound

Introduction

Adventure games come in many shapes and sizes, and cover a wide range of topics. You can buy text only games (e.g. 'Zork'), text-based games with additional graphics (Level Nine's work), games that are largely graphical and have only the most rudimentary of text descriptions ('The Hobbit'), loosely traditional adventures with animated graphics ('Valhalla'), icon-adventures ('The Fourth Protocol'), and the latest genre of 'arcade-adventures' ('Blackwych').

This book deals solely with the first three types of games. It ignores animation, icons and arcade games, since they are not really adventure games in the traditional sense of the word, and most of them owe very little to the real world of adventure-quest solving. In this chapter we'll look at adding graphics and sound to more traditional adventure games.

Existing graphics

A number of highly successful games have made use of no more than the existing Commodore 64 graphics character set. For every location you have a simple graphical picture of the area (e.g. 'Mystery of Munroe Manor'), which may or may not tell you something about the room that you're exploring.

These sort of graphical pictures are easily built up and appear virtually instantaneously on the screen, but they look clumsy and take up a fair amount of memory. All that is required to draw them is a great deal of patience, and a thorough knowledge of the characters that the 64 can produce in all its variety of colours and reverse/normal fields of display.

Consequently I'm not going to say too much about them. If you want to use this sort of display, it's easy enough to implement. Whenever you print up a room description you should also have a collection of simple PRINT statements filled with whatever graphical characters are necessary to build up a picture of that room or location. You won't be able to achieve anything very sophisticated, but reasonable pictures can be produced (as anyone who has seen any of the old *Cursor* magazine programs can testify), and you will be justified in calling your game a 'graphical adventure'.

However, using the Commodore character set to produce sophisticated graphical adventures is like trying to paint the Sistine chapel by numbers. In order to produce anything of any lasting quality we have to resort to other methods, of which the simplest and easiest is altering the existing character set.

New graphics

Our first example will only alter the alphabetic characters on the 64, and can be used to give a kind of 'gothic' image to a game, or a 'futuristic' image, or whatever, depending on the type of character set you use. The following example produces a futuristic set of letters, and looks quite interesting.

```
1 POKE53280.9:POKE53281.7:PRINT"[CLR.BLK]CHARACTER
  SET!";PRINTCHR$(8)
2 PRINT"[CD]JUST HANG ON WHILE WE READ SOME DATA I
  N."
3 POKE52.48:POKE56.48:CLR
4 POKE56334,PEEK(56334)AND254
6 POKE1,PEEK(1)AND251
8 FORI=0TO511:POKEI+12288,PEEK(I+53248):NEXT
10 PRINT"[CLR]"
12 POKE1,PEEK(1)OR4
14 POKE56334,PEEK(56334)OR1
16 POKE53272,(PEEK(53272)AND240)+12
18 FOR I=12296 TO 12527:READC:POKEI,C:NEXT
20 REM ***DATA FOR ALPHABET***
22 DATA254,198,254,198,230,230,0,252,206,206,2
  52,206,206,252,0
24 DATA254,198,192,192,192,198,254,0,252,206,198,1
  98,198,206,252,0
26 DATA254,198,192,248,192,198,254,0,254,198,192,2
  48,224,224,224,0
28 DATA254,198,192,206,198,206,254,0,198,198,198,2
  54,230,230,230,0
30 DATA254,48,48,56,56,56,254,0,62,24,24,24,28,220
```

```

.252.0
32 DATA198,204,248,240,248,220,206,0,192,192,192,2
24,224,224,254,0
34 DATA198,238,254,214,230,230,230,0,230,246,254,2
54,206,230,230,0
36 DATA248,216,206,198,198,198,254,0,255,230,230,2
54,192,224,224,0
38 DATA254,230,198,198,222,254,14,0,254,198,198,25
4,248,220,206,0
40 DATA254,198,192,254,6,198,254,0,254,178,48,48,5
6,56,56,0
42 DATA198,198,198,230,230,230,254,0,198,198,230,2
30,230,124,56,0
44 DATA198,198,198,214,254,238,198,0,198,198,124,5
6,124,198,198,0
46 DATA198,198,238,124,48,48,48,0,255,198,12,24,48
,230,254,0
48 DATA0,127,127,127,0,247,247,247
50 DATA36,129,24,24,60,126,126,60
52 DATA24,60,126,255,126,110,110,106

```

READY.

The important lines are those from line 2 to line 18, which tell the 64 to get its character data from our new data (lines 22-52) rather than its existing set stored in ROM.

In this example room is set aside at the top of memory to store the character data, although you may wish to change the exact location of the top of memory in order to give yourself a larger or smaller amount of room to play with. To do this, you'll need to alter the values POKEd into memory locations 52 and 56 in line 3. Location 52 controls the amount of room we've got in which to store strings, and location 56 alters the top of Basic memory itself. In the example given the top of Basic is moved to hexadecimal location \$4800, or decimal 18432. You may wish to move this higher up in order to give yourself more room for the adventure game itself.

This program only alters the existing alphabetic characters, and in order to go a little bit further than that and make up some really exotic displays you may care to alter the entire graphical, numeric and alphabetic character set of the 64. The following program leads the way.

```

130 POKE 53280,9:POKE 53281,7
140 PRINT"[CLR.BLK]          * CHARACTER DESIGNER *
":PRINT"[3CD]PLEASE HANG ON!"
150 POKE 828,0

```

```

160 RUN 170
170 CS=12288
175 POKE 56334,PEEK(56334)AND254:POKE 1,PEEK(1)AND
251
180 FOR I=CS TO CS+2047
185 PRINT"[HOME]";I
190 POKE I,PEEK(53248+I-CS)
200 NEXT I
205 POKE 1,PEEK(1)OR4:POKE 56334,PEEK(56334)OR1
210 PRINT"[CLR]5 RUN 280"
220 PRINT"RUN"
230 POKE 198,3
240 POKE 631,19
250 POKE 632,13
260 POKE 633,13
270 END
280 S=1024:CL=40
290 CS=12288
300 CR=0:LN=30000+PEEK(828)
310 P=24:BG=1:BR=1
320 POKE 53280,9:POKE 53281,7
330 DEFFNA(XX)=S+R*2*CL+2*C:REM SCREEN POKE LOCATI
ON
340 DEFFNB(XX)=8*R+C:REM SCREEN POKE VALUE FOR CHA
R
350 GOTO 1000
360 PRINT"[CLR,YEL]":GOSUB 1200
370 PRINT"[HOME]";:FOR I=0 TO 7
380 PRINT". . . . .":PRINT
390 NEXT I:F=0
400 PRINT"[HOME]":R=0:C=0
410 Z=FNA(0)
420 IF F=0 THEN 460
430 IF Z=ZL THEN 450
440 POKE ZL,IL:ZL=Z:IL=PEEK(ZL)
450 POKE Z+54272,1
460 POKE Z+54272,1
470 GET A$:IF A$="" THEN 470
480 POKE Z+54272,0
490 REM
500 REM CURSOR CONTROL OPTIONS
505 REM
510 IF A$="Q" THEN 1500
520 IF A$="[CR]" AND C=7 THEN C=0:GOTO 410
530 IF A$="[CR]" THEN C=C+1:GOTO 410
540 IF A$="[CL]" AND C=0 THEN C=7:GOTO 410
550 IF A$="[CL]" THEN C=C-1:GOTO 410
560 IF A$="[CD]" AND R=7 THEN R=0:GOTO 410
570 IF A$="[CD]" THEN R=R+1:GOTO 410
580 IF A$="[CU]" AND R=0 THEN R=7:GOTO 410
590 IF A$="[CU]" THEN R=R-1:GOTO 410

```

```

600 IF A$="[HOME]" THEN 400
610 IF F=1 THEN 800
695 REM
700 REM DEFINE NEW CHARACTER OPTIONS
705 REM
710 IF A$="+" THEN POKE Z,81:GOTO 410
712 IF A$="M" THEN 2400
714 IF A$="I" THEN 2500
716 IF A$="T" THEN 2600
720 IF A$="-" THEN POKE Z,46:GOTO 410
730 IF A$="=" THEN 1600
740 IF A$="[CLR]" THEN 370
750 IF A$="R" THEN 1000
760 IF A$="B" THEN 2000
770 GOTO 410
795 REM
800 REM REVIEW CHARACTER SET OPTIONS
805 REM
810 CR=FNB(0)
820 IF A$="N" THEN POKE 53272,21:GOTO 360
830 IF A$="E" THEN POKE 53272,21:F=0:GOTO 1800
840 GOTO 410
995 REM
1000 REM DISPLAY CHARACTER SET OPTIONS
1005 REM
1010 POKE 53272,(PEEK(53272)AND240)+12:R=4:C=0
1020 ZL=FNA(0):IL=32
1030 F=1:PRINT"[CLR]";
1040 PRINT"[BLK]@ A B C D E F G":PRINT
1050 PRINT"H I J K L M N O":PRINT
1060 PRINT"P Q R S T U V W":PRINT
1070 PRINT"X Y Z [ \ ] ^ _":PRINT
1080 PRINT" ! "CHR$(34)" # $ % & '":PRINT
1090 PRINT"( ) * + , - . /":PRINT
1100 PRINT"0 1 2 3 4 5 6 7":PRINT
1110 PRINT"8 9 : ; < = > ?":PRINT
1120 PRINT"[HOME]"SPC(22)"OPTIONS":PRINT
1130 PRINTSPC(22)"[RVS]N[OFF] NEW CHAR":PRINT
1140 PRINTSPC(22)"[RVS]E[OFF] EDIT CHAR":PRINT
1150 PRINTSPC(22)"[RVS]Q[OFF] QUIT"
1160 BC=PEEK(55296)
1170 GOTO 410
1195 REM
1200 REM EDIT OPTIONS
1205 REM
1210 PRINT"[HOME,BLK]"SPC(25)"[RVS]OPTIONS[OFF]":P
RINT
1220 PRINT
1230 PRINTSPC(P)"[RVS]+[OFF]ADD DOT"
1240 PRINTSPC(P)"[RVS]-[OFF] ERASE"
1250 PRINTSPC(P)"[RVS]=[OFF] UPDATE"

```



```

1260 PRINTSPC(P)"[RVS]R[OFF] REVIEW"
1270 PRINTSPC(P)"[RVS]Q[OFF] QUIT"
1272 PRINTSPC(P)"[RVS]M[OFF] MIRROR CHAR"
1274 PRINTSPC(P)"[RVS]I[OFF] INVERT CHAR
1276 PRINTSPC(P)"[RVS]T[OFF] TURN CHAR
1280 PRINTSPC(P)"[RVS]B[OFF] ADD DATA"
1290 PRINTSPC(P+2)"STATEMENT"
1300 RETURN
1495 REM
1500 REM QUIT
1505 REM
1510 REM
1520 POKE 53272,21
1530 POKE 53281,6:POKE 53280,14
1540 PRINT"[CLR,2CD]PLEASE BE CAREFUL: UDGS CONFUS
E THE 64!"
1550 END
1595 REM
1600 REM UPDATE
1605 REM
1610 PRINT"[HOME]":
1620 X=CS+8*CR
1630 FOR R=0 TO 7:SM=0
1640 FOR C=0 TO 7:D=7-C
1650 SM=SM-2^D*(PEEK(FNA(0))=81)
1660 NEXT C
1670 POKE X+R,SM
1680 PRINTSPC(17):SM:PRINT
1690 NEXT R:R=0:C=0
1700 GOTO 410
1795 REM
1800 REM EDIT CHAR
1805 REM
1810 PRINT"[CLR]"
1820 X=CS+8*CR
1830 FOR R=0 TO 7:Y=PEEK(X+R)
1840 FOR C=0 TO 7:Z=FNA(0)
1850 Q=46:Y=Y*2
1860 IF Y>255 THEN Q=81:Y=Y-256
1870 POKE Z,Q:POKE Z+54272,0
1880 NEXT C,R
1890 R=0:C=0
1900 GOSUB 1200
1910 GOTO 410
1995 REM
2000 REM ADD DATA STATEMENTS
2005 REM
2010 X=CS+8*CR
2020 PRINT"[CLR,8CD]"
2030 PRINTLN:"DATA":
2040 PRINTRIGHT$(STR$(X),LEN(STR$(X))-1):

```

```

2050 FOR I=X TO X+7
2060 PRINT",";
2070 PRINTRIGHT$(STR$(PEEK(I)),LEN(STR$(PEEK(I)))-
1);
2080 NEXT I
2090 PRINT:PRINT"RUN [HOME]"
2100 POKE 828,PEEK(828)+1
2110 POKE 198,9
2120 FOR I=0 TO 8
2130 POKE I+631,13
2140 NEXT I
2160 END
2200 FORI=0TO7:FORJ=0TO7
2202 A%(I,J)=0:IFPEEK(S+I*80+J*2)=81THENA%(I,J)=1
2204 NEXTJ,I
2206 RETURN
2300 FORI=0TO7:FORJ=0TO7
2302 IF A%(I,J)=1THENPOKES+I*80+2*J,81:GOTO2306
2304 POKES+I*80+2*J,46
2306 NEXTJ,I
2308 RETURN
2400 GOSUB2200
2402 FORI=0TO7:FORJ=7TO0STEP-1
2404 IF A%(I,J)=1THENPOKES+I*80+(7-J)*2,81:GOTO2408
2406 POKES+I*80+(7-J)*2,46
2408 NEXTJ,I:GOTO410
2500 GOSUB2200
2502 FORI=0TO7:FORJ=0TO7
2504 IF A%(I,J)=0THENA%(I,J)=1:GOTO2508
2506 A%(I,J)=0
2508 NEXTJ,I
2510 GOSUB2300:GOTO410
2600 GOSUB2200
2602 FORI=0TO7:FORJ=0TO7
2604 IFPEEK(S+I*80+J*2)=81THENA%(7-J,7-I)=1:GOTO26
08
2606 A%(7-J,7-I)=0
2608 NEXTJ,I
2610 FORI=0TO7:FORJ=7TO0STEP-1
2612 IF A%(I,J)=1THENPOKES+I*80+(7-J)*2,81:GOTO2616
2614 POKES+I*80+(7-J)*2,46
2616 NEXTJ,I:GOTO410

```

READY.

Although this program doesn't alter the top of memory, it does store the data for the character set only about 10K above the start of Basic, and so you may wish to move this if you want to use your own characters in an adventure game.

Lines 175 and 205 are the ones to alter, with location 56334 being the villain of the piece.

The rest of the program is fairly straightforward. A visual display of the characters that you can alter is shown on the screen, and by moving the 'cursor' around with the normal cursor control keys you can select which character you wish to change. Then, in lines 1200-1300 you are presented with a list of available options of things to do to the character selected.

Adding and erasing dots are obvious enough when you run the program, updating a character gives you the collection of data values which go to make up that character image, reviewing the situation takes you back to the main menu, while quit and add data should be easy enough to understand. Mirror, invert and turn a character respectively give you a mirror image of the character, turn off all the dots that go to make a character and vice versa, and turn the character through ninety degrees at a time.

Those last three are most useful when it comes to designing graphical characters, because they give you the opportunity to see what those characters look like under a variety of different circumstances.

When you've finished with the program all you'll need are the data statements for each individual character, and the opening lines which place those statements into memory and tell the 64 to get its new information from there and not from ROM.

If you can work out a way of getting rid of those lines without the user of your adventure knowing what is happening, you will then have a large amount of memory left for your game and, at the same time, a completely new character set to play with. Thus you can not only print out your room descriptions in, say, gothic text, but you can also use the other characters to produce some interesting and exciting room displays.

This has many advantages over using the existing character set, not least of which being that it makes your adventures look different from everyone else's!

Sprites

On the other hand, you could resort to using sprites to produce some sort of animation in your game. If you're writing a western adven-

ture and someone types in the command:

DRAW GUN

wouldn't it be nice to have a spritely gun being drawn on to the screen before the program responds to the actual request from the player? Sprites are ideal for this sort of thing, and since multi-colour sprites are much more versatile than ordinary two-colour ones, the following program allows you to design and save your own multi-colour sprite displays.

```
10 REM MULTI-COLOUR SPRITE GENERATOR
29 REM
30 REM IF ANY SPRITE DATA,SET UP SPRITE
32 POKE 53272,21
40 POKE 828,0
45 V=53248:NO=13:SP=13:POKE V+21,0
50 READ FF
60 IF FF>0THEN 804
69 REM
70 REM NO MORE SPRITE DATA
71 REM
80 POKE53281,0:POKE53280,0:PRINT"[YEL]
85 GOSUB 3000
90 DEFFNA(ZZ)=1065+R*40+C
110 XL=0:YL=1:XG=16:SE=21:XY=23:XX=29
120 SC=39:PRINT"[CLR]"
130 POKE 2040,13:POKE V+SE,1:POKE V+XY,0
140 POKE V+XX,0:POKE V+XL,255:POKE V+YL,190
150 POKE V+XG,0
160 X=255:Y=190
165 POKE V+28,1
169 REM
170 REM SET UP DISPLAY
171 REM
172 FORI=0TO7:AX(I)=2^I:NEXT
180 PRINT"[HOME,YEL,SP,24CBMP]"
181 FORI=0TO20
182 PRINT"[CBMN,24SP,CBMH]"
183 NEXTI:PRINT"[SP,24CBMY]"
184 GOSUB1000
185 LOC=64*NO:PRINT"[HOME]"
190 FORI=LOCTOLOC+62STEP3:PRINT"[CR]";
195 A$="":C$="[BLK,WHT,RED,CYN,PUR,GRN,BLU,YEL,ORN
G,BRN,PINK,GRY1,GRY2,LGRN]"
196 C$=C$+"[LBLU,GRY3]"
200 FORJ=0TO2
210 ZZ=PEEK(I+J)
220 FORK=7TO0STEP-1
230 A=INT((ZZANDA%(K))/A%(K))
```

```

240 B$=B$+STR$(A):C=C+1
250 IFC=2THENC=0:IFVAL(B$)>0THENGOSUB291:GOTO260
255 IFC=0THENB$="":PRINT[YEL]..":
260 NEXTK
270 NEXTJ
280 PRINT
290 NEXTI:GOTO300
291 IFB$=" 0 1"THENPRINTMID$(C$,A2+1,1)"[RVS,2SP,0
FF]":GOTO294
292 IFB$=" 1 1"THENPRINTMID$(C$,A3+1,1)"[RVS,2SP,0
FF]":GOTO294
293 IFB$=" 1 0"THENPRINTMID$(C$,A4+1,1)"[RVS,2SP,0
FF]":
294 B$="":RETURN
300 GOSUB1000
309 REM
310 REM SPRITE SET UP ON THE SCREEN
320 REM INPUT CHANGES
321 REM
330 R=0:C=0
340 Z=FNA(0)
342 PC=PEEK(Z):PD=PEEK(Z+1):PK=Z
343 IFPC=160THENRV=128:GOTO345
344 RV=0
345 POKE Z,24+RV:POKE Z+1,24+RV
346 KN=PEEK(Z+54272):KM=PEEK(Z+54273):IFKN=0THENKN
=1:KM=1
350 POKEZ+54272,KN:POKE Z+54273,KM
360 GETA$:IFA$=""THEN360
370 POKEZ+54272,KN:POKE Z+54273,KM:POKEPK,PC:POKEP
K+1,PD
380 IFA$="Q"THEN1600
390 IFA$="[CR]"ANDC=22THENC=0:GOTO340
400 IFA$="[CR]"THENC=C+2:GOTO340
410 IFA$="[CL]"ANDC=0THENC=22:GOTO340
420 IFA$="[CL]"THENC=C-2:GOTO340
430 IFA$="[CD]"ANDR=20THENR=0:GOTO340
440 IFA$="[CD]"THENR=R+1:GOTO340
450 IFA$="[CU]"ANDR=0THENR=20:GOTO340
460 IFA$="[CU]"THENR=R-1:GOTO340
470 IFA$="[HOME]"THENR=0:C=0:GOTO340
480 IFA$="[CLR]"THENGOSUB1150:GOTO340
490 IFA$="+ "THEN580
500 IFA$="- "THEN730
510 IFA$="M"THEN1210
520 IFA$="B"THEN1450
530 IFA$="C" THEN 1400
540 IFA$="E"THEN900
570 GOTO 340
574 REM
575 REM ADD POINT

```



```

576 REM
580 Z=FNA(0)
590 IFCS=A2THENK1=224:K2=160
595 IFCS=A4THENK1=160:K2=224
600 IFCS=A3THENK1=160:K2=160
610 POKEZ,K1:POKE Z+1,K2:POKE55296+Z-1024,CS:POKE
55296+Z-1023,CS
620 BYTE=INT(C/8)+R*3:GOSUB625:GOTO340
625 CH=1065+40*R+8*INT(C/8):TC=0:BI=8
626 FORL1=CHTOCH+7:P=PEEK(L1):BI=BI-1:IFP=160ORP=2
4THENTC=TC+2^BI
627 NEXTL1:POKEBYTE+NO*64,TC:RETURN
724 REM
725 REM DELETE POINT
726 REM
730 Z=FNA(0)
740 Z1=PEEK(Z)
760 POKE Z,46:POKE Z+1,46:POKE54272+Z,7:POKE54273+
Z,7
770 BYTE=INT(C/8)+R*3:GOSUB625:GOTO340
804 REM
805 REM IF ANY DATA. SET SPRITES UP
806 REM
808 READ A1,A2,A3:IFA1=0THENPOKE V+28,PEEK(V+28)AN
D255-2^(0):GOTO810
809 POKE V+28,PEEK(V+28)OR2^(0):POKE V+37,A2:POKE
V+38,A3
810 READ A4:POKE V+39,A4
815 LOC=SP*64
820 FOR I=LOC TO LOC+62
830 READ A:POKE I,A
840 NEXT I
850 GOTO50
894 REM
895 REM INPUT FOR EXPAND
896 REM
900 PRINT"[HOME,23CD,10CR]PRESS X OR Y"
910 GETA$:IFA$<>"X"AND A$<>"Y"THEN900
920 IFA$="X"THEN960
930 IFPEEK(V+XY)=1THENPOKEV+XY,0:GOTO980
940 POKEV+XY,1
950 GOTO980
960 IFPEEK(V+XX)=1THENPOKEV+XX,0:GOTO980
970 POKEV+XX,1
980 PRINT"[HOME,23CD,10CR,12SP]"
990 GOTO 340
994 REM
995 REM DISPAY CONTROL OPTIONS
996 REM
1000 PRINT"[HOME,YEL]SPC(27)"[RVS]CONTROLS[OFF]"
1005 PRINTSPC(26)"SPRITE # 0"

```

```

1030 PRINTSPC(26)"[RVS]M[OFF]OVE SPRITE"
1040 PRINTSPC(26)"[RVS]C[OFF]HANGE COLOUR"
1050 PRINTSPC(26)"[RVS]E[OFF]XPAND"
1060 PRINTSPC(26)"[RVS]+[OFF] ADD DOT"
1070 PRINTSPC(26)"[RVS]-[OFF] REMOVE DOT"
1080 PRINTSPC(26)"[RVS]B[OFF]ASIC DATA"
1090 PRINTSPC(26)"[RVS]Q[OFF]UIT"
1100 PRINT:PRINTSPC(26)"USE CURSOR"
1110 PRINTSPC(26)"CONTROL TO"
1120 PRINTSPC(26)"POSITION"
1130 PRINTSPC(26)"CURSOR."
1140 RETURN
1144 REM
1145 REM CLEAR PRESENT SPRITE
1146 REM
1150 FORI=0T062:POKEN0*64+I,0:NEXTI
1160 FORI=0T020
1170 FORJ=1T024
1180 POKE1064+I*40+J,46:POKE55336+I*40+J,7
1190 NEXTJ,I:R=0:C=0
1200 RETURN
1204 REM
1205 REM MOVE SPRITE AROUND SCREEN
1206 REM
1210 PRINT"[HOME,23CD]USE CURSOR KEYS/RETURN TO EX
IT."
1230 GETA$:IFA$=""THEN1230
1240 IFA$="[CR]"ANDX<319THENX=X+2
1250 IFA$="[CL]"ANDX>1THENX=X-2
1260 IFA$="[CD]"ANDY<254THENY=Y+2
1270 IFA$="[CU]"ANDY>1THENY=Y-2
1280 POKE V+YL,Y
1290 POKE V+XG,INT(X/255)
1300 POKE V+XL,X-INT(X/255)*255
1310 IF A$=CHR$(13)THEN1330
1320 GOTO1210
1330 POKE V+XL,255
1340 POKE V+YL,190
1350 POKE V+XG,0
1360 X=255:Y=190
1370 PRINT"[HOME,23CD,36SP]"
1390 GOTO 340
1394 REM
1395 REM CHANGE SPRITE COLOUR
1396 REM
1400 PRINT"[HOME,23CD,YEL]PRESS 1, 2 OR 3 TO MAKE
YOUR CHOICE."
1402 GETCG$:IFCG$=""THEN1402
1403 IFCG$="1"THENC$=A2:GOTO1420
1404 IFCG$="2"THENC$=A3:GOTO1420
1405 IFCG$="3"THENC$=A4:GOTO1420

```

```

1406 GOTO 1402
1420 POKE V+36+VAL(CG$),CS
1430 PRINT"[HOME,23CD,37SP,HOME]";
1440 GOTO 340
1444 REM
1445 REM CREATE DATA STATEMENTS FOR
1446 REM PRESENT SPRITE
1447 REM
1450 PP=PP+11:PRINT"[CLR,3CD]";PP+30000;"DATA";SP;
",1,";A2;"",":A3;"",":A4;"
1460 FORI=0TO8
1470 PRINTPP+I+30001"DATA";
1480 FORJ=0TO6
1490 BB=PEEK(NO*64+I*7+J)
1500 BB$=RIGHT$(STR$(BB),LEN(STR$(BB))-1)
1510 PRINTBB$;"",":";
1520 NEXT J
1530 PRINT"[CL] "
1540 NEXT I
1550 PRINTPP+10+30000;"DATA -1":NO=NO+1:SP=SP+1
1560 PRINT"GOTO45[HOME]
1570 POKE 198,12
1580 FORI=0TO11:POKE631+I,13:NEXT I
1590 END
1600 PRINT"[CLR,2CD]BYE.":END
3000 PRINT"[CLR]#MULTI-COLOUR SPRITE DESIGNER."
3002 PRINT"[2CD]YOU CAN DISPLAY UP TO FOUR COLOURS
PER SPRITE NOW. THESE ARE :="
3004 PRINT"[2CD]SPRITE MULTI-COLOUR ONE,
3006 PRINT"SPRITE MULTI-COLOUR TWO,
3008 PRINT"ORDINARY SPRITE COLOUR,"
3010 PRINT"AND THE SCREEN COLOUR."
3012 PRINT"[2CD]THIS LIMITS OUR SPRITE RESOLUTION
TO 12 BY 21 CHARACTERS."
3014 PRINT:PRINT"NEVERTHELESS, WITH A LITTLE BIT O
F IMAGINATION, SOME STAGGERING ";
3016 PRINT"RESULTS CANBE ACHIEVED."
3018 GOSUB 4000
3020 PRINT"[CLR,YEL]FIRST OF ALL, YOU MUST CHOOSE
THE COLOURFROM THE USUAL 16.
3022 PRINT"[HOME,3CD]"TAB(28)CHR$(18)"BLACK - 0"
3024 PRINTTAB(28)CHR$(18)"WHITE - 1"
3025 PRINTTAB(28)CHR$(18)"RED - 2"
3026 PRINTTAB(28)CHR$(18)"CYAN - 3"
3027 PRINTTAB(28)CHR$(18)"PURPLE - 4"
3028 PRINTTAB(28)CHR$(18)"GREEN - 5"
3029 PRINTTAB(28)CHR$(18)"BLUE - 6"
3030 PRINTTAB(28)CHR$(18)"YELLOW - 7"
3031 PRINTTAB(28)CHR$(18)"ORANGE - 8"
3032 PRINTTAB(28)CHR$(18)"BROWN - 9"
3033 PRINTTAB(28)CHR$(18)"PINK -10"

```

```

3034 PRINTTAB(28)CHR$(18)"GREY1  -11"
3035 PRINTTAB(28)CHR$(18)"GREY2  -12"
3036 PRINTTAB(28)CHR$(18)"L.BLUE  -13"
3037 PRINTTAB(28)CHR$(18)"L.GRN   -14"
3038 PRINTTAB(28)CHR$(18)"GREY3   -15"
3040 INPUT "SPRITE MULTI-COLOUR ZERO":A2:IFA2<ODRA
2>15THENPRINT"[2CU]":GOTO3040
3042 INPUT "SPRITE MULTI-COLOUR ONE":A3:IFA3<ODRA3
>15THENPRINT"[2CU]":GOTO3042
3044 INPUT "SPRITE ORDINARY COLOUR":A4:IFA4<ODRA4>
15THENPRINT"[2CU]":GOTO3044
3045 CS=A4
3046 GOSUB 4000
3048 POKE V+37,A2:POKE V+38,A3:POKE V+39,A4
3050 PRINT"[CLR]MOVE THE CURSOR AROUND THE SCREEN
WITH THE CURSOR":
3051 PRINT" KEYS. DITTO FOR THE SPRITE USING MEN
U OPTION M.
3053 PRINT"[CD]WHEN SATISFIED, TURN IT INTO DATA
STATEMENTS BY USING OPTION B.
3054 PRINT"THESE ARE LISTED AS LINES 30011-30020."
3055 PRINT"[CD]THE REST OF THE INSTRUCTIONS ARE
DISPLAYED ON THE SCREEN."
3056 PRINT"[CD]APART FROM...! PRESS HOME TO MOVE
THE CURSOR BACK TO THE TOP LEFT ":
3057 PRINT", AND CLR/HOME TO ERASE THE SPRIT
E COMPLETELY."
3058 PRINT"[CD]ALSO, WHEN CHANGING COLOURS, PRESSI
NG '1' GIVES YOU MULTI-COLOUR ":
3059 PRINT"ZERO, '2' GIVES YOU MULTI-COLOUR ONE
,AND '3' GIVESTHE ORDINARY ";
3060 PRINT"SPRITE COLOUR."
3065 GOSUB 4000
3070 RETURN
4000 PRINT"[2CD]PRESS 'SPACE' TO CONTINUE"
4002 GETSP$:IFSP$<>" THEN4002
4004 RETURN
29997 REM
29998 REM SPRITE DATA STORED FROM HERE
29999 REM
30011 DATA 13 ,1, 1 , 9 , 12
30012 DATA0,2,128,0,2,128,0
30013 DATA2,128,0,2,128,0,2
30014 DATA128,10,170,128,8,0,0
30015 DATA8,0,0,12,0,0,12
30016 DATA68,0,29,1,0,79,16
30017 DATA64,31,192,16,7,255,244
30018 DATA1,255,253,0,127,244,0
30019 DATA31,208,0,7,64,0,1
30020 DATA0,0,0,0,0,0,0
30021 DATA -1

```

READY.

The first part of the program sets up a sprite, if any data statements already exist, and then goes on to draw up the grid on which you can design and alter your sprites. The instructions are printed up via a subroutine starting at line 3000, and the menu options while designing sprites are shown in lines 1000 to 1140.

As you can see from those lines a number of interesting options are available, and they should be sufficient for you to produce any kind of display that your game might require. Playing an adventure the other day, I noticed that both the title page and various parts of the program, while featuring no other graphics, were brightened up by the inclusion of a number of sprites. This can enhance the 'playability' of a game if done well, and certainly wakes you up at three o'clock in the morning when you think that you're never going to get anywhere with a particular game.

However, more and more people seem to be resorting to split screens and high-resolution graphics to flavour their games, so let's move on to a number of routines that will help us out in those particular areas.

Splitting screens

Sitting in the 64, at memory locations 788 and 789, is something called a hardware interrupt vector. If you PEEK those locations you'll see that the normal values stored there are 49 and 234 respectively. Don't, whatever you do, alter those values without having some other code in the machine as well to deal with the alteration. Those two locations tell the 64 that roughly fifty times a second it has got to jump to an internal ROM routine starting at memory location 234, 49, or in more understandable hexadecimal terms \$EA31 (49 being the decimal equivalent of \$31 and 234 being the decimal equivalent of \$EA).

This routine performs a lot of useful actions that keep the 64 ticking over, and any untoward alteration of the contents of locations 788 and 789 could lead to disaster. However, with a little bit of careful thought we can get those locations to do something very useful for us. If they are altered to point to another machine-code routine in memory (since they are in RAM they're easily altered by a simple POKE command), then fifty times a second the 64 will race off to that routine. As long as we tell the 64 at the end of the routine that it has then got to go to location \$EA31 everything will be all right.

The following short program uses this principle to give us a split screen.

```

90 POKE 53265,27
100 FORJ=828TO864:READX
110 T=T+X:POKEJ,X
120 NEXTJ
200 DATA 169,1,141,25,208,162,194,160,0,173,18,208
,16,4,162,1
210 DATA 160,7,142,18,208,140,33,208,173,13,220
220 DATA 41,1,240,3,76,49,234,76,188,254
300 POKE56333,127
310 POKE 788,60:POKE789,3
320 POKE53274,129

```

READY.

As you can see, this is a short Basic loader for a very useful machine-code routine. As well as altering a couple of locations controlled by the 64's video chip, the program puts a small routine into memory, starting at location 828. Line 310 of the program then instructs the 64 to go to location \$0360, or 828 in decimal, before trundling off to location \$EA31 again. The machine-code listing makes that easier to follow:

```

B*
      PC  SR  AC  XR  YR  SP
.:77B2 33 00 9A 00 F6
.
033C A9 01          LDA #$01
033E BD 19 D0      STA $D019
0341 A2 C2          LDX #$C2
0343 A0 00          LDY #$00
0345 AD 12 D0      LDA $D012
0348 10 04          BPL $034E
034A A2 01          LDX #$01
034C A0 07          LDY #$07
034E BE 12 D0      STX $D012
0351 BC 21 D0      STY $D021
0354 AD 0D DC      LDA $DC0D
0357 29 01          AND #$01
0359 F0 03          BEQ $035E
035B 4C 31 EA      JMP $EA31
035E 4C BC FE      JMP $FEBC
0361 00          BRK
.
.

```

If you type in and run the program, you'll see that the top two-thirds of the screen are now black, and the bottom third is in yellow. To change those colours, alter the ninth data value (currently a zero) and the eighteenth data value (currently a seven). For example, if you want red and green instead you'll have to change the zero and seven to two and five.

To alter the point at which the split occurs, alter the seventh data value. This determines how far down the screen the dividing line will occur.

In order to make use of this you'll need to be able to put some high-resolution graphics at the top of the screen and some normal text at the bottom, so the following program is an introduction to producing high-resolution graphical displays.

High resolution

This is only a start, it isn't the finished product, but it will enable you to produce high-resolution displays with a reasonable amount of ease.

```
1 B=0:T=0
2 READA:IFAK<>-1THENPOKE49152+B.A:B=B+1:T=T+A:GOTO2
3 PRINT:PRINT"DATA READ IN ...":IFT<>47642THENPRIN
T:PRINT"BUT THERE'S AN ERROR SOMEWHERE.":STOP
4 PRINT:PRINT"CORRECTLY! WELL DONE."
10 DATA76.30,192.76.61,192.76
20 DATA84,192.76,113,192.76.139
30 DATA192.76.142,192.76.82,193
40 DATA76.58,193,76.98,193.234
50 DATA234,234.173,17,208,141,114
60 DATA193,173,24,208,141,115,193
70 DATA169,59.141,17.208,169,24
80 DATA141,24,208,32,61,192,162
90 DATA16.32.90,192,96.160.0
100 DATA169,32,132,253,133,254.152
110 DATA145,253,200,208,251,230.254
120 DATA165.254.201,64,208,242,96
130 DATA32,253,174,32.158,183.160
140 DATA0,169,4,132,253,133,254
150 DATA138,145,253,200,208.251.230
160 DATA254,165,254,201.8,208,242
170 DATA96,160,0,169,32.132.253
180 DATA133,254,177,253,73,255.145
190 DATA253,200,208,247,230,254.165
200 DATA254,201,64,208,239,96,169
```

210 DATA0,44,169,128,133,151,32
 220 DATA253,174,32,235,183,224,200
 230 DATA176,238,165,21,201,1,144
 240 DATA8,208,230,165,20,201,64
 250 DATA176,224,138,74,74,74,168
 260 DATA185,33,193,141,117,193,185
 270 DATA8,193,141,118,193,138,41
 280 DATA7,24,109,117,193,141,117
 290 DATA193,165,20,41,248,141,116
 300 DATA193,24,169,0,109,117,193
 310 DATA133,253,169,32,109,118,193
 320 DATA133,254,24,165,253,109,116
 330 DATA193,133,253,165,254,101,21
 340 DATA133,254,165,20,41,7,73
 350 DATA7,170,169,1,202,48,3
 360 DATA10,208,250,160,0,36,151
 370 DATA16,5,73,255,49,253,44
 380 DATA17,253,145,253,96,0,0
 390 DATA1,2,3,5,6,7,8
 400 DATA10,11,12,13,15,16,17
 410 DATA18,20,21,22,23,25,27
 420 DATA28,29,0,64,128,192,0
 430 DATA64,128,192,0,64,128,192
 440 DATA0,64,128,192,0,64,128
 450 DATA192,0,64,128,192,0,32
 460 DATA253,174,32,212,225,162,0
 470 DATA160,64,169,0,133,253,169
 480 DATA32,133,254,169,253,32,216
 490 DATA255,96,32,253,174,32,212
 500 DATA225,169,97,133,185,169,0
 510 DATA32,213,255,96,173,114,193
 520 DATA141,17,208,173,115,193,141
 530 DATA24,208,32,68,229
 540 DATA96,-1

READY.

Or, for those of you who prefer the machine-code approach:

B*

```

      PC  SR  AC  XR  YR  SP
.:97FE 30 00 00 00 F6
.
C000 4C 1E C0      JMP $C01E
C003 4C 3D C0      JMP $C03D
C006 4C 54 C0      JMP $C054
C009 4C 71 C0      JMP $C071
C00C 4C 8B C0      JMP $C08B
C00F 4C 8E C0      JMP $C08E
C012 4C 52 C1      JMP $C152
C015 4C 3A C1      JMP $C13A
  
```


C018	4C	62	C1	JMP	\$C162
C01B	EA			NOP	
C01C	EA			NOP	
C01D	EA			NOP	
C01E	AD	11	D0	LDA	\$D011
C021	8D	72	C1	STA	\$C172
C024	AD	18	D0	LDA	\$D018
C027	8D	73	C1	STA	\$C173
C02A	A9	3B		LDA	#\$3B
C02C	8D	11	D0	STA	\$D011
C02F	A9	18		LDA	#\$18
C031	8D	18	D0	STA	\$D018
C034	20	3D	C0	JSR	\$C03D
C037	A2	10		LDX	#\$10
C039	20	5A	C0	JSR	\$C05A
C03C	60			RTS	
C03D	A0	00		LDY	##00
C03F	A9	20		LDA	##20
C041	84	FD		STY	\$FD
C043	85	FE		STA	\$FE
C045	98			TYA	
C046	91	FD		STA	(\$FD),Y
C048	C8			INY	
C049	D0	FB		BNE	\$C046
C04B	E6	FE		INC	\$FE
C04D	A5	FE		LDA	\$FE
C04F	C9	40		CMP	##40
C051	D0	F2		BNE	\$C045
C053	60			RTS	
C054	20	FD	AE	JSR	\$AEFD
C057	20	9E	B7	JSR	\$B79E
C05A	A0	00		LDY	##00
C05C	A9	04		LDA	##04
C05E	84	FD		STY	\$FD
C060	85	FE		STA	\$FE
C062	8A			TXA	
C063	91	FD		STA	(\$FD),Y
C065	C8			INY	
C066	D0	FB		BNE	\$C063
C068	E6	FE		INC	\$FE
C06A	A5	FE		LDA	\$FE
C06C	C9	08		CMP	##08
C06E	D0	F2		BNE	\$C062
C070	60			RTS	
C071	A0	00		LDY	##00
C073	A9	20		LDA	##20
C075	84	FD		STY	\$FD
C077	85	FE		STA	\$FE
C079	B1	FD		LDA	(\$FD),Y
C07B	49	FF		EOR	##FF
C07D	91	FD		STA	(\$FD),Y

C07F	C8		INY
C080	D0	F7	BNE \$C079
C082	E6	FE	INC \$FE
C084	A5	FE	LDA \$FE
C086	C9	40	CMP ##40
C088	D0	EF	BNE \$C079
C08A	60		RTS
C08B	A9	00	LDA ##00
C08D	2C	A9 B0	BIT \$B0A9
C090	85	97	STA \$97
C092	20	FD AE	JSR \$AEFD
C095	20	EB B7	JSR \$B7EB
C098	E0	C8	CPX ##C8
C09A	B0	EE	BCS \$C08A
C09C	A5	15	LDA \$15
C09E	C9	01	CMP ##01
COA0	90	08	BCC \$COAA
COA2	D0	E6	BNE \$C08A
COA4	A5	14	LDA \$14
COA6	C9	40	CMP ##40
COAB	B0	E0	BCS \$C08A
COAA	8A		TXA
COAB	4A		LSR
COAC	4A		LSR
COAD	4A		LSR
COAE	A8		TAY
COAF	B9	21 C1	LDA \$C121.Y
COB2	8D	75 C1	STA \$C175
COB5	B9	08 C1	LDA \$C108.Y
COB8	8D	76 C1	STA \$C176
COBB	8A		TXA
COBC	29	07	AND ##07
COBE	18		CLC
COBF	6D	75 C1	ADC \$C175
COC2	8D	75 C1	STA \$C175
COC5	A5	14	LDA \$14
COC7	29	F8	AND ##F8
COC9	8D	74 C1	STA \$C174
COCC	18		CLC
COCD	A9	00	LDA ##00
COCF	6D	75 C1	ADC \$C175
COD2	85	FD	STA \$FD
COD4	A9	20	LDA ##20
COD6	6D	76 C1	ADC \$C176
COD9	85	FE	STA \$FE
CODB	18		CLC
CODC	A5	FD	LDA \$FD
CODE	6D	74 C1	ADC \$C174
COE1	85	FD	STA \$FD
COE3	A5	FE	LDA \$FE
COE5	65	15	ADC \$15

COE7 85 FE	STA \$FE
COE9 A5 14	LDA \$14
COEB 29 07	AND #\$07
COED 49 07	EOR #\$07
COEF AA	TAX
COF0 A9 01	LDA #\$01
COF2 CA	DEX
COF3 30 03	BMI \$COFB
COF5 0A	ASL
COF6 D0 FA	BNE \$COF2
COFB A0 00	LDY #\$00
COFA 24 97	BIT \$97
COFC 10 05	BPL \$C103
COFE 49 FF	EOR \$FFF
C100 31 FD	AND (\$FD),Y
C102 2C 11 FD	BIT \$FD11
C105 91 FD	STA (\$FD),Y
C107 60	RTS
C108 00	BRK
C109 00	BRK
C10A 01 02	ORA (\$02,X)
C10C 03	???
C10D 05 06	ORA \$06
C10F 07	???
C110 08	PHP
C111 0A	ASL
C112 0B	???
C113 0C	???
C114 0D 0F 10	ORA \$100F
C117 11 12	ORA (\$12),Y
C119 14	???
C11A 15 16	ORA \$16,X
C11C 17	???
C11D 19 1B 1C	ORA \$1C1B,Y
C120 1D 00 40	ORA \$4000,X
C123 80	???
C124 C0 00	CPY #\$00
C126 40	RTI
C127 80	???
C128 C0 00	CPY #\$00
C12A 40	RTI
C12B 80	???
C12C C0 00	CPY #\$00
C12E 40	RTI
C12F 80	???
C130 C0 00	CPY #\$00
C132 40	RTI
C133 80	???
C134 C0 00	CPY #\$00
C136 40	RTI
C137 80	???

C138	C0	00	CPY	#\$00
C13A	20	FD AE	JSR	\$AEFD
C13D	20	D4 E1	JSR	\$E1D4
C140	A2	00	LDX	#\$00
C142	A0	40	LDY	#\$40
C144	A9	00	LDA	#\$00
C146	85	FD	STA	\$FD
C148	A9	20	LDA	##20
C14A	85	FE	STA	\$FE
C14C	A9	FD	LDA	##FD
C14E	20	DB FF	JSR	##FFDB
C151	60		RTS	
C152	20	FD AE	JSR	\$AEFD
C155	20	D4 E1	JSR	\$E1D4
C158	A9	61	LDA	##61
C15A	85	B9	STA	\$B9
C15C	A9	00	LDA	##00
C15E	20	D5 FF	JSR	##FFD5
C161	60		RTS	
C162	AD	72 C1	LDA	\$C172
C165	8D	11 D0	STA	\$D011
C168	AD	73 C1	LDA	\$C173
C16B	8D	18 D0	STA	\$D018
C16E	20	44 E5	JSR	\$E544
C171	60		RTS	
C172	00		BRK	

This adds a number of extra 'commands' to Basic, although all of them need to be accessed by SYStem calls rather than by using new Basic keywords.

In order, they produce the following effects:

SYS49152 initialises the package and sets up a high-resolution screen.

SYS49155 clears a high resolution screen.

SYS49158,n sets the colour of the screen, with n ranging as usual from 0 to 15.

SYS49161 inverts the graphics screen, so that every dot which was on is now off, and vice versa.

SYS49164,x,y plots a point at the co-ordinates x and y.

SYS49167,x,y unplots a point at the co-ordinates x and y.

SYS49170, "filename", dn loads a previously saved high-resolution screen called "filename" from device number dn (1 for tape, 8 for disk).

SYS49173, "filename", dn saves a high-resolution screen to device number dn under the filename "filename".

SYS49176 turns off the high-resolution screen and returns everything to normal again. (Well, as normal as you ever get on the 64).

Points to note

You may have noticed, if you've flipped through this and some of the other chapters in this book, that a large number of the machine-code routines that we are discussing start at the same location in memory, namely \$C000, or decimal 49152. There is a good reason for this, since this is the start of the spare 4K in the middle of the 64's ROM.

However, if you want to use them all at the same time you're in trouble! - well, not really, since most of the routines are completely relocatable (in other words, instead of living from locations \$C000-\$C100 they can live from \$C200-\$C300, or whatever). Just remember to alter the SYStem calls accordingly.

The only trouble with something like this is that you are using a complete 8K of screen. This is all very well if you want to produce displays which can be turned on or off by repeated pressing of, say, the CBM logo key, so that pressing it once produces the high-resolution screen and pressing it again produces the text underneath it, but what about if you want to produce windowed displays, i.e., you want the top ten lines in high-resolution mode but the bottom fifteen in ordinary text mode?

Ah me, such a demanding lot you are out there. Let's take a look at the next listing, which sorts out that particular problem.

Interrupts

As we saw with the split screen program, interrupting the 64 can produce some interesting and downright useful effects, as the next program will illustrate.

```

10 SE = 49152 : EE = 49688
20 :
30 FOR I = SE TO EE : READ A : POKE I , A : NEXT
40 :
50000 DATA 076.009,192.076,091.193.076,146.193.120
.169,127.141.013
50001 DATA 220.169,001,141.026,208,173,090.193.133
.251,173,142,192
50002 DATA 141.018,208.169,024.141,017.208.173,020
.003,141,060,003
50003 DATA 173.021,003.141,061.003.169,065.141,020
,003,169.192,141
50004 DATA 021,003,088,169,016,141.067,003.096.173
,025,208,141,025
50005 DATA 208,041,001,240,058,198,251,016,007,173
,090,193,133,251
50006 DATA 198,251,166,251,189,192,192,141,032,208
,189,167,192,141
50007 DATA 033,208,189,064,193,141,136,002,189,217
,192,141,017,208
50008 DATA 189,242,192,141,022,208,189,012,193,141
,024,208,189,142
50009 DATA 192,141,018,208,138,240,006,104,168,104
,170,104,064,076
50010 DATA 049,234,001,241,233,225,217,209,201,193
,185,177,169,161
50011 DATA 153,145,137,129,121,113,105,097,089,081
,073,065,057,000
50012 DATA 000,000,000,000,000,000,000,000,000,000
,000,000,000,000
50013 DATA 000,000,000,000,000,000,000,000,000,000
,012,012,012,012
50014 DATA 012,012,012,012,012,012,012,012,012,012
,012,012,012,012
50015 DATA 012,012,012,012,012,012,012,012,027,027,027
,027,027,027,027
50016 DATA 027,027,027,027,027,027,027,027,027,027
,027,027,027,027
50017 DATA 027,027,027,027,008,008,008,008,008,008
,008,008,008,008
50018 DATA 008,008,008,008,008,008,008,008,008,008
,008,008,008,008
50019 DATA 008,008,020,020,020,020,020,020,020,020
,020,020,020,020
50020 DATA 020,020,020,020,020,020,020,020,020,020
,020,020,020,003
50021 DATA 003,003,003,003,003,003,003,003,003,003
,003,003,003,003
50022 DATA 003,003,003,003,003,003,003,003,003,003
,003,003,004,004
50023 DATA 004,004,004,004,004,004,004,004,004,004

```

```
,004,004,004,004
50024 DATA 004,004,004,004,004,004,004,004,004,004
,025,032,015,194
50025 DATA 165,020,041,015,141,061,003,032,015,194
,165,020,041,015
50026 DATA 141,062,003,160,000,173,061,003,153,167
,192,173,062,003
50027 DATA 153,192,192,169,027,153,217,192,169,008
,153,242,192,169
50028 DATA 020,153,012,193,200,192,025,208,224,096
,032,015,194,165
50029 DATA 020,201,025,144,002,169,024,141,066,003
,169,024,056,237
50030 DATA 066,003,141,066,003,032,015,194,165,020
,141,061,003,032
50031 DATA 015,194,165,020,141,062,003,032,015,194
,165,020,201,003
50032 DATA 144,002,169,000,141,063,003,032,015,194
,165,020,201,002
50033 DATA 144,002,169,000,141,064,003,032,015,194
,165,020,141,065
50034 DATA 003,172,066,003,120,173,061,003,153,167
,192,173,062,003
50035 DATA 153,192,192,174,063,003,189,010,194,153
,217,192,174,064
50036 DATA 003,189,013,194,153,242,192,173,067,003
,013,065,003,153
50037 DATA 012,193,088,096,027,091,059,008,024,032
,253,174,032,138
50038 DATA 173,032,247,183,096
```

READY.

Or again, for all you machine-code freaks out there:

B*

```
PC SR AC XR YR SP
.:8FEB 33 00 D3 00 F6
.
C000 4C 09 C0      JMP $C009
C003 4C 5B C1      JMP $C15B
C006 4C 92 C1      JMP $C192
C009 7B           SEI
C00A A9 7F        LDA #$7F
C00C 8D 0D DC      STA $DC0D
C00F A9 01        LDA #$01
C011 8D 1A D0      STA $D01A
C014 AD 5A C1      LDA $C15A
C017 B5 FB        STA $FB
C019 AD BE C0      LDA $C0BE
C01C 8D 12 D0      STA $D012
```

C01F	A9	18		LDA	#\$18
C021	8D	11	D0	STA	\$D011
C024	AD	14	03	LDA	\$0314
C027	8D	3C	03	STA	\$033C
C02A	AD	15	03	LDA	\$0315
C02D	8D	3D	03	STA	\$033D
C030	A9	41		LDA	#\$41
C032	8D	14	03	STA	\$0314
C035	A9	C0		LDA	#\$C0
C037	8D	15	03	STA	\$0315
C03A	58			CLI	
C03B	A9	10		LDA	#\$10
C03D	8D	43	03	STA	\$0343
C040	60			RTS	
C041	AD	19	D0	LDA	\$D019
C044	8D	19	D0	STA	\$D019
C047	29	01		AND	#\$01
C049	F0	3A		BEQ	\$C085
C04B	C6	FB		DEC	\$FB
C04D	10	07		BPL	\$C056
C04F	AD	5A	C1	LDA	\$C15A
C052	85	FB		STA	\$FB
C054	C6	FB		DEC	\$FB
C056	A6	FB		LDX	\$FB
C058	8D	C0	C0	LDA	\$C0C0,X
C05B	8D	20	D0	STA	\$D020
C05E	8D	A7	C0	LDA	\$C0A7,X
C061	8D	21	D0	STA	\$D021
C064	8D	40	C1	LDA	\$C140,X
C067	8D	88	02	STA	\$0288
C06A	8D	D9	C0	LDA	\$C0D9,X
C06D	8D	11	D0	STA	\$D011
C070	8D	F2	C0	LDA	\$C0F2,X
C073	8D	16	D0	STA	\$D016
C076	8D	0C	C1	LDA	\$C10C,X
C079	8D	18	D0	STA	\$D018
C07C	8D	8E	C0	LDA	\$C08E,X
C07F	8D	12	D0	STA	\$D012
C082	8A			TXA	
C083	F0	06		BEQ	\$C08B
C085	68			PLA	
C086	AB			TAY	
C087	68			PLA	
C08B	AA			TAX	
C089	68			PLA	
C08A	40			RTI	
C08B	4C	31	EA	JMP	\$EA31
C08E	01	F1		ORA	(\$F1,X)
C090	E9	E1		SBC	#\$E1
C092	D9	D1	C9	CMP	\$C9D1,Y
C095	C1	B9		CMP	(\$B9,X)

C097	B1	A9	LDA (\$A9),Y
C099	A1	99	LDA (\$99,X)
C09B	91	89	STA (\$89),Y
C09D	81	79	STA (\$79,X)
C09F	71	69	ADC (\$69),Y
C0A1	61	59	ADC (\$59,X)
C0A3	51	49	EOR (\$49),Y
C0A5	41	39	EOR (\$39,X)
C0A7	00		BRK
C0A8	00		BRK
C0A9	00		BRK
C0AA	00		BRK
C0AB	00		BRK
C0AC	00		BRK
C0AD	00		BRK
C0AE	00		BRK
C0AF	00		BRK
COB0	00		BRK
COB1	00		BRK
COB2	00		BRK
COB3	00		BRK
COB4	00		BRK
COB5	00		BRK
COB6	00		BRK
COB7	00		BRK
COB8	00		BRK
COB9	00		BRK
COBA	00		BRK
COBB	00		BRK
COBC	00		BRK
COBD	00		BRK
COBE	00		BRK
COBF	00		BRK
COC0	0C		???
COC1	0C		???
COC2	0C		???
COC3	0C		???
COC4	0C		???
COC5	0C		???
COC6	0C		???
COC7	0C		???
COC8	0C		???
COC9	0C		???
COCA	0C		???
COCB	0C		???
COCC	0C		???
COCd	0C		???
COCE	0C		???
COCF	0C		???
COD0	0C		???
COD1	0C		???

COD2	0C	???
COD3	0C	???
COD4	0C	???
COD5	0C	???
COD6	0C	???
COD7	0C	???
COD8	0C	???
COD9	1B	???
CODA	1B	???
CODB	1B	???
CODC	1B	???
CODD	1B	???
CODE	1B	???
CODF	1B	???
COE0	1B	???
COE1	1B	???
COE2	1B	???
COE3	1B	???
COE4	1B	???
COE5	1B	???
COE6	1B	???
COE7	1B	???
COE8	1B	???
COE9	1B	???
COEA	1B	???
COEB	1B	???
COEC	1B	???
COED	1B	???
COEE	1B	???
COEF	1B	???
COF0	1B	???
COF1	1B	???
COF2	0B	PHP
COF3	0B	PHP
COF4	0B	PHP
COF5	0B	PHP
COF6	0B	PHP
COF7	0B	PHP
COF8	0B	PHP
COF9	0B	PHP
COFA	0B	PHP
COFB	0B	PHP
COFC	0B	PHP
COFD	0B	PHP
COFE	0B	PHP
COFF	0B	PHP
C100	0B	PHP
C101	0B	PHP
C102	0B	PHP
C103	0B	PHP
C104	0B	PHP

C105	08	PHP
C106	08	PHP
C107	08	PHP
C108	08	PHP
C109	08	PHP
C10A	08	PHP
C10B	08	PHP
C10C	14	???
C10D	14	???
C10E	14	???
C10F	14	???
C110	14	???
C111	14	???
C112	14	???
C113	14	???
C114	14	???
C115	14	???
C116	14	???
C117	14	???
C118	14	???
C119	14	???
C11A	14	???
C11B	14	???
C11C	14	???
C11D	14	???
C11E	14	???
C11F	14	???
C120	14	???
C121	14	???
C122	14	???
C123	14	???
C124	14	???
C125	03	???
C126	03	???
C127	03	???
C128	03	???
C129	03	???
C12A	03	???
C12B	03	???
C12C	03	???
C12D	03	???
C12E	03	???
C12F	03	???
C130	03	???
C131	03	???
C132	03	???
C133	03	???
C134	03	???
C135	03	???
C136	03	???
C137	03	???

C138	03		???
C139	03		???
C13A	03		???
C13B	03		???
C13C	03		???
C13D	03		???
C13E	03		???
C13F	03		???
C140	04		???
C141	04		???
C142	04		???
C143	04		???
C144	04		???
C145	04		???
C146	04		???
C147	04		???
C148	04		???
C149	04		???
C14A	04		???
C14B	04		???
C14C	04		???
C14D	04		???
C14E	04		???
C14F	04		???
C150	04		???
C151	04		???
C152	04		???
C153	04		???
C154	04		???
C155	04		???
C156	04		???
C157	04		???
C158	04		???
C159	04		???
C15A	19	20 0F	DRA \$0F20.Y
C15D	C2		???
C15E	A5	14	LDA \$14
C160	29	0F	AND #\$0F
C162	8D	3D 03	STA \$033D
C165	20	0F C2	JSR \$C20F
C168	A5	14	LDA \$14
C16A	29	0F	AND #\$0F
C16C	8D	3E 03	STA \$033E
C16F	A0	00	LDY #\$00
C171	AD	3D 03	LDA \$033D
C174	99	A7 C0	STA \$C0A7.Y
C177	AD	3E 03	LDA \$033E
C17A	99	C0 C0	STA \$C0C0.Y
C17D	A9	1B	LDA #\$1B
C17F	99	D9 C0	STA \$C0D9.Y
C182	A9	0B	LDA #\$0B

C184	99	F2	C0	STA	\$C0F2.Y
C187	A9	14		LDA	#\$14
C189	99	0C	C1	STA	\$C10C.Y
C18C	C8			INY	
C18D	C0	19		CPY	#\$19
C18F	D0	E0		BNE	\$C171
C191	60			RTS	
C192	20	0F	C2	JSR	\$C20F
C195	A5	14		LDA	\$14
C197	C9	19		CMP	#\$19
C199	90	02		BCC	\$C19D
C19B	A9	18		LDA	#\$18
C19D	8D	42	03	STA	\$0342
C1A0	A9	18		LDA	#\$18
C1A2	38			SEC	
C1A3	ED	42	03	SBC	\$0342
C1A6	8D	42	03	STA	\$0342
C1A9	20	0F	C2	JSR	\$C20F
C1AC	A5	14		LDA	\$14
C1AE	8D	3D	03	STA	\$033D
C1B1	20	0F	C2	JSR	\$C20F
C1B4	A5	14		LDA	\$14
C1B6	8D	3E	03	STA	\$033E
C1B9	20	0F	C2	JSR	\$C20F
C1BC	A5	14		LDA	\$14
C1BE	C9	03		CMP	#\$03
C1C0	90	02		BCC	\$C1C4
C1C2	A9	00		LDA	#\$00
C1C4	8D	3F	03	STA	\$033F
C1C7	20	0F	C2	JSR	\$C20F
C1CA	A5	14		LDA	\$14
C1CC	C9	02		CMP	#\$02
C1CE	90	02		BCC	\$C1D2
C1D0	A9	00		LDA	#\$00
C1D2	8D	40	03	STA	\$0340
C1D5	20	0F	C2	JSR	\$C20F
C1D8	A5	14		LDA	\$14
C1DA	8D	41	03	STA	\$0341
C1DD	AC	42	03	LDY	\$0342
C1E0	78			SEI	
C1E1	AD	3D	03	LDA	\$033D
C1E4	99	A7	C0	STA	\$C0A7.Y
C1E7	AD	3E	03	LDA	\$033E
C1EA	99	C0	C0	STA	\$C0C0.Y
C1ED	AE	3F	03	LDX	\$033F
C1F0	BD	0A	C2	LDA	\$C20A.X
C1F3	99	D9	C0	STA	\$C0D9.Y
C1F6	AE	40	03	LDX	\$0340
C1F9	BD	0D	C2	LDA	\$C20D.X
C1FC	99	F2	C0	STA	\$C0F2.Y
C1FF	AD	43	03	LDA	\$0343

```

C202 0D 41 03    ORA $0341
C205 99 0C C1    STA $C10C,Y
C208 5B          CLI
C209 60          RTS
C20A 1B          ???
C20B 5B          ???
C20C 3B          ???
C20D 0B          PHP
C20E 1B          CLC
C20F 20 FD AE    JSR $AEFD
C212 20 BA AD    JSR $ADBA
C215 20 F7 B7    JSR $B7F7
C218 60          RTS
C219 FF          ???
.
.

```

So, what does this little lot do?

With this program residing in memory we can, on a very Basic level, switch on what are termed 'raster interrupts', and change the colour of the screen and the border.

To switch on the interrupts, we need to:

SYS 49152

and to alter the screen and border colours we have to:

SYS 49155,sc,bc

where sc and bc represent the screen and border colours respectively, each ranging from 0 to 15 as usual.

Nothing very special so far, but by turning on the raster interrupts we can now alter every individual line on the screen. In other words, each of the 25 lines on the screen can be in any particular mode that takes our fancy. Turning on raster interrupts is just a fancy way of saying that we have now altered that part of the 64 which checks to see what each particular part of the screen is doing, be it displaying normal characters, lower-case characters, extended background colour mode characters, multi-colour characters or high-resolution characters.

This, as you might imagine, gives us the very powerful ability to produce just about anything we like on the screen of the 64, wherever we like and whenever we like. Naturally enough, using high-

resolution mode means that we have to set some memory aside for the high-resolution display, but how much memory is to be set aside now depends on how much of the screen we want to be in each particular mode.

Let's take a look at the options available to us.

Using the package

Apart from the two simple SYS commands given earlier, there is also a third catered for by this routine. This gives us the power to alter selectively any line of the screen display.

SYS 49158,ln,sc,bc,a,b,c

where these parameters can take the following values:

ln: line number, which can lie in the range 0 to 24.

sc: screen colour, which can lie in the range 0 to 15, and takes the usual colour values.

bc: border colour, which can lie in the range 0 to 15, and takes the usual colour values.

a: this controls the first screen display register, and can take values in the range 0 to 2, where:

a=0: gives normal colour mode.

a=1: gives extended background colour mode.

a=2: gives high-resolution mode.

b: this controls the second screen display register, and has the value of 0 or 1.

b=0: as with a, this gives normal colour mode.

b=1: gives multi-colour mode.

c: this controls the character memory, and determines where character and high-resolution memory areas are to be stored.

c=2: character area starts at 2048, high-resolution area starts at 2048.

c=4: character area starts at 4096, high-resolution area starts at 4096.

c=6: character area starts at 6144, high-resolution area starts at 6144.

c = 8: character area starts at 8192, high-resolution area starts at 8192.

c = 10: character area starts at 10240, high-resolution area is unavailable.

c = 12: character area starts at 12288, high-resolution area is unavailable.

c = 14: character area starts at 14336, high-resolution area is unavailable.

Note that when c=6 the character area is the normal upper-case character set, and when c=8 the character area is the normal lower-case character set.

So how does this work in practice?

Well, an example is always the best way, so having got the program typed in *correctly*, issue the following series of commands:

```
SYS 49155,7,9
```

```
SYS 49152
```

```
FORI = 0T07:SYS49158,1,7,9,2,0,8:NEXT
```

and you should see a high-resolution area appearing at the top of the screen. To clear this area, you'll need to type in something like this:

```
FORI = 8192T010752:POKEI,0:NEXT
```

This is because our high-resolution screen now starts at memory location 8192, and we are affecting 8 times 320 positions on the screen. 8 times 320 is 2560, and 2560 plus 8192 is equal to 10752.

To give some colour to the display, we now need to type in:

```
FORI = 1024T01344:POKEI,5:NEXT
```

This is because the screen memory starts at location 1024, and we are looking at 8 rows of the screen. Finally, to put something on to this high-resolution screen, the following will produce a series of high-resolution lines:

```
FORI = 8192T010752:POKEI,255:NEXT
```


Quite what you display on your high-resolution screen is up to you, but I hope the above will give you some idea of what this powerful program can do.

Many thanks to Allen Webb for a lot of the information contained in this last section.

Saving memory

Having gone to the trouble of designing some high-resolution screen displays, it would be rather handy to save them to tape or disk. The following program, which resides in memory from locations 679 to 722, does the trick. The instructions, starting at line 10000, explain how to operate it.

```
1 GOSUB10000
3 POKE53281,0:POKE53280,9:POKE646,2:PRINT"[CLR,YEL
]
4 PRINTTAB(11)"[RVS]      SAVER  64      "
5 DATA169,54,133,1
10 DATA169,8,162,8,160,255,32,186,255
20 DATA169,2,162,211,160,2,32,189,255
30 DATA169,1,133,247,169,8,133,248,162,255,160,127
,169,247,32,216,255
35 DATA169,55,133,1,96
40 FORI=679TO722:READA:POKEI,A:NEXT
50 N$="":DEV=1:S=2049:E=32767
60 INPUT"[CD]FILE NAME      ":N$:N$=LEFT$(N$,16)
62 IFN$=""THEN60
70 INPUT"DEVICE (T/D) ":DEV$
71 IFDEV$="T"THENDEV=1:GOTO80
72 IFDEV$="D"THENDEV=8:GOTO80
74 GOTO70
80 S=49152
90 E=53247
100 K=S:GOSUB1000:POKE702,A1:POKE706,A2
110 K=E:GOSUB1000:POKE710,A1:POKE712,A2
120 SA=255:POKE688,SA
130 K=LEN(N$):POKE693,K
140 FORI=1TOK
150 POKE722+I,ASC(MID$(N$,I,1)):NEXT
155 POKE684,DEV:POKE686,DEV
156 PRINT"[CD]LOAD IN MACHINE CODE &"
160 PRINT"TYPE 'SYS 679' TO SAVE"
170 END
1000 A2=INT(K/256):A1=K-A2*256:RETURN
10000 POKE53280,9:POKE53281,7:PRINT"[CLR,BLK]MACHI
NE CODE SAVER."
```

```

10010 PRINT"[CD]THIS SHORT MACHINE CODE PROGRAM R
ESIDES IN MEMORY FROM ";
10012 PRINT"LOCATIONS 679 TO 722."
10014 PRINT"[CD]ON RUNNING IT, YOU'LL BE ASKED FOR
A FILE NAME, AND A DEVICE TO ";
10016 PRINT"SAVE THE MACHINE CODE TO."
10018 PRINT"[CD]ALTERING THE VALUES OF S AND E IN
LINES 80 AND 90 DETERMINES ";
10020 PRINT"WHICH PART OF THE MACHINE'S MEMORY WI
LL BE SAVED. S AND E";
10022 PRINT"ARE THE START AND END ADDRESSES RESPEC
T-IVELY OF ANY CODE TO BE SAVED.
10024 PRINT"[CD]JUST RUN IT, ANSWER THE QUESTIONS,
AND THEN WHEN YOU'RE READY ";
10026 PRINT"TO SAVE YOUR CODEENTER (IN DIRECT OR P
ROGRAM MODE) [RVS]SYS679"
10028 PRINT"[CD]PRESS 'SPACE' TO START THE PROGRAM
."
10030 GETSP$:IFSP$<>" "THEN10030
10032 RETURN

```

READY.

And finally, we'll take a look at (listen to?) putting sound into your adventures.

Sounding out adventures

The question of sound in an adventure is almost as vexed as the question of graphics, but is a lot easier for the player to decide whether to continue with it or not. After all, televisions do have a volume control.

If you're at all familiar with the 64, you'll know about using the SID chip to produce a wide variety of noises, for doors creaking open, gates slamming shut, snakes hissing, and so on. Indeed, the 'Enormous Cave' listing given in Chapter 9 uses some simple sound effects from time to time. However, there is one fault in using sound in this way, and that is that it stops program operation. If you're going to use sound at all it makes sense to let the program carry on running while the sound is being made, and the following (interrupt-driven again) programs show just one way in which this might be done.

```

2 A=A+1:IFA=1THENLOAD "MUSICDEMO",8,1
5 POKES3280,9:POKES3281,7:POKES3272,23:PRINT"[HOME
,BLK]MUSICAL INTERRUPTS."
10 S=54272:POKES+17,255:POKES+19,142:POKES+20,150
20 POKES+24,15:POKE49216,10:POKE49218,0:POKE49219,
65:POKE251,178:POKE252,192
30 POKE253,178:POKE254,192
100 POKES+10,255:POKES+12,142:POKES+13,150
110 POKE51264,10:POKE51266,0:POKE51267,65:POKE247,
178:POKE248,200
120 POKE249,178:POKE250,200:SYS49152
130 PRINT"[CLR]AS YOU CAN NOW HEAR, IF YOU'VE TURN
ED UP THE VOLUME ON YOUR T.V.,";
140 PRINT" THERE IS SOME MUSIC PLAYING IN THE BAC
KGROUND."
150 PRINT"[CD]PROGRAM OPERATION CAN CONTINUE AS NO
RMALBUT THE TUNE WILL CARRY ON ";
160 PRINT"PLAYING UNTILWE DECIDE TO STOP IT."
170 PRINT"[CD]HOW IS ALL THIS DONE. AND HOW DO WE
ALTER THE TUNE?"
172 PRINT"[CD]PRESS 'F1' TO FIND OUT."
174 GETA$:IFA$<>"[F1]"THEN174
176 PRINT"[CLR]THE MACHINE CODE ROUTINE LIES IN ME
MORY FROM $C000 TO $C914. BUT ";
178 PRINT"THERE IS A LOT OF EMPTY SPACE IN THERE T
O ALLOW FOR THESTORING ";
180 PRINT"OF LONGER TUNES."
182 PRINT"[CD]VOICE THREE IS CONTROLLED BY THE ROU
TINEFROM $C000 TO $C0B2. AND ";
184 PRINT"VOICE TWO BY THE ROUTINE FROM $C800 TO
$C8B2."
186 PRINT"[CD]THE TUNES PLAYED ARE, FOR VOICE THRE
E, AT $C0B3 TO $C114, AND FOR ";
188 PRINT"VOICE TWO AT $C8B3 TO $C914."
190 PRINT"[CD]NOTES ARE STORED IN HI-LO FORMAT. WI
TH TWO '00'S INDICATING A PAUSE.
192 PRINT"[CD]TO TERMINATE A TUNE. ENTER A VALUE O
F 255. OR $FF."
194 PRINT"[2CD]PRESS 'F1' AGAIN FOR MORE."
196 GETB$:IFB$<>"[F1]"THEN196
198 PRINT"[CLR]LINES 10 TO 120 SET UP THE ADSR SET
TINGSFOR BOTH VOICES."
200 PRINT"[CD]TO TURN THE THING OFF. ENTER SYS4917
1."
202 PRINT"[CD]AND TO START IT AGAIN. ENTER SYS4915
2."
204 PRINT"[CD]BUT REMEMBER, WHEN YOU TURN IT OFF.
THE NOTES WILL TAKE SOME TIME ";
206 PRINT"TO DIE AWAY BECAUSE OF THE ADSR SETTIN
GS."
208 PRINT"[2CD]PRESS 'F1' TO QUIT. OR 'F7' TO REVI

```

EW THESE INSTRUCTIONS.

210 GETC\$: IFC\$="[F7]" THEN 130

212 IFC\$<>"[F1]" THEN 210

214 FORI=0 TO 24: POKE I, 0: NEXT

216 POKE 49219, 0: POKE 51267, 0: SYS 49171: POKE 53272, 21

218 END

READY.

10 SE = 49152 : EE = 49429

20 :

30 FORI = SE TO EE : READ A : POKE I, A : NEXT

40 :

50000 DATA 234,234,234,120,169,080,141,020,003,169
.192,141,021,003

50001 DATA 088,234,234,096,120,169,049,141,020,003
.169,234,141,021

50002 DATA 003,088,234,096,234,222,222,222,222,222
.222,222,222,222

50003 DATA 222,222,222,222,222,222,222,222,222,222
.222,222,222,222

50004 DATA 222,222,222,222,222,222,222,222,222,010,010
.000,000,000,000

50005 DATA 000,000,000,000,000,000,065,000,000,000,000
.234,234,234,173

50006 DATA 066,192,240,004,076,049,234,234,206,065
.192,208,247,234

50007 DATA 234,234,173,064,192,141,065,192,234,230
.251,208,002,230

50008 DATA 252,234,160,000,177,251,201,255,240,040
.234,234,234,234

50009 DATA 160,000,141,015,212,230,251,208,002,230
.252,234,177,251

50010 DATA 141,014,212,173,067,192,072,169,000,141
.018,212,104,141

50011 DATA 018,212,234,234,076,100,200,234,234,234
.165,253,133,251

50012 DATA 165,254,133,252,076,100,192,000,000,000
.000,002,036,002

50013 DATA 179,003,054,003,154,003,209,003,154,003
.054,002,179,002

50014 DATA 036,002,179,003,054,003,154,003,209,003
.154,003,054,002

50015 DATA 179,002,220,003,154,004,073,004,207,005
.024,004,207,004

50016 DATA 073,003,154,002,036,002,179,003,054,003
.154,003,209,003

50017 DATA 154,003,054,002,179,003,054,004,011,004
.207,005,102,005

50018 DATA 184,005,102,004,207,004,011,002,220,003
.154,004,073,004

50019 DATA 207,005,024,004,207,004,073,003,154,255
,255,000

READY.

```
10 SE = 51200 :EE = 51477
20 :
30 FOR I = SE TO EE : READ A : POKE I , A : NEXT
40 :
50000 DATA 234,234,234,120,169,080,141,020,003,169
,200,141,021,003
50001 DATA 088,234,234,096,120,169,049,141,020,003
,169,234,141,021
50002 DATA 003,088,234,096,234,222,222,222,222,222
,222,222,222,222
50003 DATA 222,222,222,222,222,222,222,222,222,222
,222,222,222,222
50004 DATA 222,222,222,222,222,222,222,222,010,010
,000,000,000,000
50005 DATA 000,000,000,000,000,000,065,000,000,000,000
,234,234,234,173
50006 DATA 066,200,240,004,076,049,234,234,206,065
,200,208,247,234
50007 DATA 234,234,173,064,200,141,065,200,234,230
,247,208,002,230
50008 DATA 248,234,160,000,177,247,201,255,240,040
,234,234,234,234
50009 DATA 160,000,141,008,212,230,247,208,002,230
,248,234,177,247
50010 DATA 141,007,212,173,067,200,072,169,000,141
,011,212,104,141
50011 DATA 011,212,234,234,076,049,234,234,234,234
,165,249,133,247
50012 DATA 165,250,133,248,076,100,200,000,000,000
,000,002,036,002
50013 DATA 179,003,054,003,154,003,209,003,154,003
,054,002,179,002
50014 DATA 036,002,179,003,054,003,154,003,209,003
,154,003,054,002
50015 DATA 179,002,220,003,154,004,073,004,207,005
,024,004,207,004
50016 DATA 073,003,154,002,036,002,179,003,054,003
,154,003,209,003
50017 DATA 154,003,054,002,179,003,054,004,011,004
,207,005,102,005
50018 DATA 184,005,102,004,207,004,011,002,220,003
,154,004,073,004
50019 DATA 207,005,024,004,207,004,073,003,154,255
,255,255
```

READY.

B*

PC SR AC XR YR SP
.:67BC 33 00 74 00 F6

```

C000 EA          NOP
C001 EA          NOP
C002 EA          NOP
C003 78          SEI
C004 A9 50       LDA #$50
C006 8D 14 03    STA $0314
C009 A9 C0       LDA #$C0
C00B 8D 15 03    STA $0315
C00E 58          CLI
C00F EA          NOP
C010 EA          NOP
C011 60          RTS
C012 78          SEI
C013 A9 31       LDA #$31
C015 8D 14 03    STA $0314
C018 A9 EA       LDA #$EA
C01A 8D 15 03    STA $0315
C01D 58          CLI
C01E EA          NOP
C01F 60          RTS
C020 EA          NOP
C021 DE DE DE    DEC $DEDE,X
C024 DE DE DE    DEC $DEDE,X
C027 DE DE DE    DEC $DEDE,X
C02A DE DE DE    DEC $DEDE,X
C02D DE DE DE    DEC $DEDE,X
C030 DE DE DE    DEC $DEDE,X
C033 DE DE DE    DEC $DEDE,X
C036 DE DE DE    DEC $DEDE,X
C039 DE DE DE    DEC $DEDE,X
C03C DE DE DE    DEC $DEDE,X
C03F DE 0A 0A    DEC $0A0A,X
C042 00          BRK
C043 00          BRK
C044 00          BRK
C045 00          BRK
C046 00          BRK
C047 00          BRK
C048 00          BRK
C049 00          BRK
C04A 00          BRK
C04B 41 00       EOR ($00,X)
C04D 00          BRK
C04E 00          BRK
C04F 00          BRK
C050 EA          NOP
C051 EA          NOP

```

C052	EA			NOP
C053	AD	42	C0	LDA \$C042
C056	F0	04		BEQ \$C05C
C058	4C	31	EA	JMP \$EA31
C05B	EA			NOP
C05C	CE	41	C0	DEC \$C041
C05F	D0	F7		BNE \$C058
C061	EA			NOP
C062	EA			NOP
C063	EA			NOP
C064	AD	40	C0	LDA \$C040
C067	8D	41	C0	STA \$C041
C06A	EA			NOP
C06B	E6	FB		INC \$FB
C06D	D0	02		BNE \$C071
C06F	E6	FC		INC \$FC
C071	EA			NOP
C072	A0	00		LDY #\$00
C074	B1	FB		LDA (\$FB),Y
C076	C9	FF		CMP #\$FF
C078	F0	28		BEQ \$C0A2
C07A	EA			NOP
C07B	EA			NOP
C07C	EA			NOP
C07D	EA			NOP
C07E	A0	00		LDY #\$00
C080	8D	0F	D4	STA \$D40F
C083	E6	FB		INC \$FB
C085	D0	02		BNE \$C089
C087	E6	FC		INC \$FC
C089	EA			NOP
C08A	B1	FB		LDA (\$FB),Y
C08C	8D	0E	D4	STA \$D40E
C08F	AD	43	C0	LDA \$C043
C092	48			PHA
C093	A9	00		LDA #\$00
C095	8D	12	D4	STA \$D412
C098	68			PLA
C099	8D	12	D4	STA \$D412
C09C	EA			NOP
C09D	EA			NOP
C09E	4C	64	C8	JMP \$C864
COA1	EA			NOP
COA2	EA			NOP
COA3	EA			NOP
COA4	A5	FD		LDA \$FD
COA6	85	FB		STA \$FB
COA8	A5	FE		LDA \$FE
COAA	85	FC		STA \$FC
COAC	4C	64	C0	JMP \$C064
COAF	00			BRK

COB0	00	BRK
COB1	00	BRK
COB2	00	BRK
COB3	02	???
COB4	24 02	BIT #02
COB6	B3	???
COB7	03	???
COB8	36 03	ROL #03,X
COBA	9A	TXS
COBB	03	???
COBC	D1 03	CMP (#03),Y
COBE	9A	TXS
COBF	03	???
COC0	36 02	ROL #02,X
COC2	B3	???
COC3	02	???
COC4	24 02	BIT #02
COC6	B3	???
COC7	03	???
COC8	36 03	ROL #03,X
COCA	9A	TXS
COCB	03	???
COCC	D1 03	CMP (#03),Y
COCE	9A	TXS
COCF	03	???
COD0	36 02	ROL #02,X
COD2	B3	???
COD3	02	???
COD4	DC	???
COD5	03	???
COD6	9A	TXS
COD7	04	???
COD8	49 04	EOR #04
CODA	CF	???
CODB	05 18	ORA #18
CODD	04	???
CODE	CF	???
CODF	04	???
COE0	49 03	EOR #03
COE2	9A	TXS
COE3	02	???
COE4	24 02	BIT #02
COE6	B3	???
COE7	03	???
COE8	36 03	ROL #03,X
COEA	9A	TXS
COEB	03	???
COEC	D1 03	CMP (#03),Y
COEE	9A	TXS
COEF	03	???
COF0	36 02	ROL #02,X

COF2	B3	???
COF3	03	???
COF4	36 04	RDL \$04,X
COF6	0B	???
COF7	04	???
COF8	CF	???
COF9	05 66	ORA \$66
COFB	05 BB	ORA \$BB
COFD	05 66	ORA \$66
COFF	04	???
C100	CF	???
C101	04	???
C102	0B	???
C103	02	???
C104	DC	???
C105	03	???
C106	9A	TXS
C107	04	???
C108	49 04	EOR \$\$04
C10A	CF	???
C10B	05 1B	ORA \$1B
C10D	04	???
C10E	CF	???
C10F	04	???
C110	49 03	EOR \$\$03
C112	9A	TXS
C113	FF	???
C114	FF	???

.		
C800	EA	NOP
C801	EA	NOP
C802	EA	NOP
C803	7B	SEI
C804	A9 50	LDA \$\$50
C806	BD 14 03	STA \$0314
C809	A9 CB	LDA \$\$CB
C80B	BD 15 03	STA \$0315
C80E	5B	CLI
C80F	EA	NOP
C810	EA	NOP
C811	60	RTS
C812	7B	SEI
C813	A9 31	LDA \$\$31
C815	BD 14 03	STA \$0314
C81B	A9 EA	LDA \$\$EA
C81A	BD 15 03	STA \$0315
C81D	5B	CLI
C81E	EA	NOP
C81F	60	RTS
C820	EA	NOP

C821	DE	DE	DE	DEC	\$DEDE,X
C824	DE	DE	DE	DEC	\$DEDE,X
C827	DE	DE	DE	DEC	\$DEDE,X
C82A	DE	DE	DE	DEC	\$DEDE,X
C82D	DE	DE	DE	DEC	\$DEDE,X
C830	DE	DE	DE	DEC	\$DEDE,X
C833	DE	DE	DE	DEC	\$DEDE,X
C836	DE	DE	DE	DEC	\$DEDE,X
C839	DE	DE	DE	DEC	\$DEDE,X
C83C	DE	DE	DE	DEC	\$DEDE,X
C83F	DE	0A	0A	DEC	\$0A0A,X
C842	00			BRK	
C843	00			BRK	
C844	00			BRK	
C845	00			BRK	
C846	00			BRK	
C847	00			BRK	
C848	00			BRK	
C849	00			BRK	
C84A	00			BRK	
C84B	41	00		EOR	(\$00,X)
C84D	00			BRK	
C84E	00			BRK	
C84F	00			BRK	
C850	EA			NOP	
C851	EA			NOP	
C852	EA			NOP	
C853	AD	42	C8	LDA	\$C842
C856	F0	04		BEQ	\$C85C
C858	4C	31	EA	JMP	\$EA31
C85B	EA			NOP	
C85C	CE	41	C8	DEC	\$C841
C85F	D0	F7		BNE	\$C858
C861	EA			NOP	
C862	EA			NOP	
C863	EA			NOP	
C864	AD	40	C8	LDA	\$C840
C867	8D	41	C8	STA	\$C841
C86A	EA			NOP	
C86B	E6	F7		INC	\$F7
C86D	D0	02		BNE	\$C871
C86F	E6	F8		INC	\$F8
C871	EA			NOP	
C872	A0	00		LDY	#\$00
C874	B1	F7		LDA	(\$F7),Y
C876	C9	FF		CMP	#\$FF
C878	F0	28		BEQ	\$C8A2
C87A	EA			NOP	
C87B	EA			NOP	
C87C	EA			NOP	
C87D	EA			NOP	

C87E	A0	00		LDY	#\$00
C880	8D	0B	D4	STA	\$D40B
C883	E6	F7		INC	\$F7
C885	D0	02		BNE	\$C889
C887	E6	F8		INC	\$F8
C889	EA			NOP	
C88A	B1	F7		LDA	(\$F7),Y
C88C	8D	07	D4	STA	\$D407
C88F	AD	43	CB	LDA	\$CB43
C892	4B			PHA	
C893	A9	00		LDA	#\$00
C895	8D	0B	D4	STA	\$D40B
C898	6B			PLA	
C899	8D	0B	D4	STA	\$D40B
C89C	EA			NOP	
C89D	EA			NOP	
C89E	4C	31	EA	JMP	\$EA31
C8A1	EA			NOP	
C8A2	EA			NOP	
C8A3	EA			NOP	
C8A4	A5	F9		LDA	\$F9
C8A6	85	F7		STA	\$F7
C8A8	A5	FA		LDA	\$FA
C8AA	85	F8		STA	\$F8
C8AC	4C	64	CB	JMP	\$CB64
C8AF	00			BRK	
C8B0	00			BRK	
C8B1	00			BRK	
C8B2	00			BRK	
C8B3	02			???	
C8B4	24	02		BIT	\$02
C8B6	B3			???	
C8B7	03			???	
C8B8	36	03		ROL	\$03,X
C8BA	9A			TXS	
C8BB	03			???	
C8BC	D1	03		CMP	(\$03),Y
C8BE	9A			TXS	
C8BF	03			???	
C8C0	36	02		ROL	\$02,X
C8C2	B3			???	
C8C3	02			???	
C8C4	24	02		BIT	\$02
C8C6	B3			???	
C8C7	03			???	
C8C8	36	03		ROL	\$03,X
C8CA	9A			TXS	
C8CB	03			???	
C8CC	D1	03		CMP	(\$03),Y
C8CE	9A			TXS	
C8CF	03			???	

C8D0	36	02	ROL	\$02,X
C8D2	B3		???	
C8D3	02		???	
C8D4	DC		???	
C8D5	03		???	
C8D6	9A		TXS	
C8D7	04		???	
C8D8	49	04	EOR	#\$04
C8DA	CF		???	
C8DB	05	18	ORA	\$18
C8DD	04		???	
C8DE	CF		???	
C8DF	04		???	
C8E0	49	03	EOR	#\$03
C8E2	9A		TXS	
C8E3	02		???	
C8E4	24	02	BIT	\$02
C8E6	B3		???	
C8E7	03		???	
C8E8	36	03	ROL	\$03,X
C8EA	9A		TXS	
C8EB	03		???	
C8EC	D1	03	CMP	(\$03),Y
C8EE	9A		TXS	
C8EF	03		???	
C8F0	36	02	ROL	\$02,X
C8F2	B3		???	
C8F3	03		???	
C8F4	36	04	ROL	\$04,X
C8F6	0B		???	
C8F7	04		???	
C8F8	CF		???	
C8F9	05	66	ORA	\$66
C8FB	05	B8	ORA	\$B8
C8FD	05	66	ORA	\$66
C8FF	04		???	
C900	CF		???	
C901	04		???	
C902	0B		???	
C903	02		???	
C904	DC		???	
C905	03		???	
C906	9A		TXS	
C907	04		???	
C908	49	04	EOR	#\$04
C90A	CF		???	
C90B	05	18	ORA	\$18
C90D	04		???	
C90E	CF		???	
C90F	04		???	
C910	49	03	EOR	#\$03

C912 9A	TXS
C913 FF	???
C914 00	BRK
.	
.	

The first program contains all the instructions for using and altering the remaining Basic or machine-code listings. Quite simply, you determine which note you want to play, say the third C note that the 64 can play, and by consulting your 64 manual you'll find that this converts to hi-lo values of 8 and 147 respectively. Converting these to hexadecimal gives you the numbers \$8 and \$93. These are then POKEd into memory (or simply typed in using an assembler) starting at locations \$C0B3 and \$C0B4 for voice three, or \$C8B3 and \$C8B4 for voice two. The next note is then stored at locations \$C0B5 and \$C0B6 (for voice three), and so on. When you come to the end of a tune, put the value 255, or hexadecimal \$FF, into memory, and the tune will start again from the beginning.

ADSR settings are POKEd in at the start of the program.

Of course, you don't have to play a tune. Moody, atmospheric, occasional notes can do much to enhance an adventure.

Conclusion

We have explored many graphical avenues in this chapter, and hopefully you'll be able to use one or more of them in your own adventures - if you want to, that is! Graphics *can* be useful and, if used wisely, can improve an otherwise lacklustre adventure. Whether they do so or not is up to you and your imagination. Mind you, after all is said and done, I suppose you could always use disk drives and go and buy a *Koala Pad*!

Talking about disk drives ...

3

Tape or Disk?

Introduction

When sitting down to think about writing an adventure, one of the first decisions you must make is whether or not you are going to use tape or disk as a storage medium. If you only have one or the other then there's not much of a decision involved, but if you're in the fortunate position of having both you'll have to decide which one to use.

By use, I don't just mean whether you're going to save the program on disk or tape. If you've got a disk drive there seems little point in developing a program and continually saving it to or loading it from tape. The time spent doing this alone is prohibitive.

No, the decision to be made is whether you load the program from tape and have it all residing in the machine at the same time, or whether you load the main program from disk and then continually access the disk to print up messages, room descriptions and the like.

Using disk has one great obvious advantage: much more memory, which means a bigger game, a larger vocabulary, and richer room descriptions and messages. On the other hand, unless you use some machine-code disk-handling routines and either write your main program in machine code or compile it, you'll spend a lot of time listening to the whirring of the Commodore disk drive as it strives to find the correct piece of information on disk. Commodore drives are not the fastest in the world, and although a time of (say) five seconds to access any part of a file may not seem like a very long time, if you've got something like 200 room descriptions and 400 messages stored away, you'll find that at the end of a complete game featuring all descriptions and messages you'll have wasted some fifty minutes on disk access time.

Consider a case in point. Some time ago Jim Butterfield produced a disk-based version of the original Colossal Cave, and very good it was too: it certainly sparked off my interest in this particular genre of game. Recently I produced a tape-based version of the same game, using the same story line, objects, treasures and so on. The only difference (apart from the program of course) was that Jim's kept accessing the disk and mine resided in memory at all times, although to be truthful I did have to cut down on some of the text. A reviewer in a user-group magazine, comparing the two versions of the game, preferred mine to Jim's for precisely that reason: he wasn't continually waiting for files to be hauled in from disk.

There is an example of both types of game in this book. Chapters 5 and 8 cover a tape-based one, and Chapter 9 covers a disk-based one. Although the tape one is a deliberately small game to give newcomers a gentle introduction into the art of writing adventures, the disk program contains ten times as many locations, with generally longer descriptions, a much wider-ranging vocabulary, and a whole host of messages that combine to produce a much more interesting game.

Looking at this question from a commercial viewpoint is a different matter, since software houses seem reluctant to release adventure games on disk, always preferring to use tape-only games. Quite why this is so I have no idea, since I'm sure there are sufficient disk users around to warrant releasing games on disk, but ours is not to reason why, ours is to produce an adventure game or die, so as usual I'll leave the decision up to you.

Whichever you choose, let's take a look at some typical uses of both tape and disk when producing adventure games.

Saving on memory

When using either tape or disk there are some simple rules which, if correctly observed, will make a great difference to the size of your Basic program. Since adventure games tend to eat up memory, this can be no bad thing.

(1) Programs look much more legible if lines are spaced out, as in the following example:

```
10 FOR I = 1 TO 10 : PRINT I : NEXT I
```

However, that takes up a lot more space in memory than:

```
10 FOR I = 1 TO 10:PRINT I:NEXT
```

Not only have we removed the I parameter after the NEXT statement (although you can't always do this if more than one loop is operative at a time), but we've also got rid of the additional 10 superfluous spaces. The line may now be harder to read, but ten spaces is ten bytes, and taken over a long program can save at least as much as a K of memory.

(2) Each line number takes up a certain amount of memory, whereas putting colons between separate statements with the same line number takes up less. Thus, it might look neater to have:

```
10 A = 2
20 B = 5
30 C = A*B
40 PRINT C
```

But far less wasteful is:

```
10 A = 2:B = 5:C = A*B:PRINT C
```

(3) Variables should always be declared at the start of a program. Furthermore, whenever a numerical constant is to be used more than once (e.g. the mathematical constant PI), it saves an awful lot of memory to define a variable to be equal to PI. Thus we might have something like:

```
10 PI = 3.14159
20 PRINT "THE AREA OF A CIRCLE WITH RADIUS 10 IS ";
30 PRINT PI*10*10
```

Note the use of 10*10 rather than 10 to the power 2. It's a lot quicker to execute.

Since so many routines on the 64 depend on the use of sound (starting memory location 54272) and sprites (53248), it makes a lot of sense to define a couple of variables SO = 54272 and SP = 53248, and then just refer to other locations as SO plus something or SP plus something.

(4) String variables take up space as soon as they're declared, regardless of the length of that string. Make the string longer, and

yet more space is eaten up, but making it shorter still leaves us with a certain amount of memory occupied. Obviously it isn't possible to avoid strings at all times, but their use should be kept to a sensible minimum.

(5) Matrices eat up memory, and if you're going to define a two-dimensional matrix of any size, make sure that every element of the array is used. Whether it is used or not, the Commodore 64 still reserves some space for it, so if it remains unused it's a criminal waste of memory.

(6) Subroutines certainly save on memory, since one subroutine and five GOSUBs to it take up far less space than duplicating the code five times. But beware. If a subroutine is used only once in a program, it will occupy more space than merely typing out the code, since the commands GOSUB and RETURN themselves occupy memory.

(7) It is sometimes convenient to separate out mathematical expressions with extra brackets to make them more legible. But brackets take up memory space, and it is advisable to let operator precedence sort the expression out rather than use unnecessary brackets.

(8) Wherever possible always use integer variables instead of ordinary ones: the saving on memory can be quite dramatic. Try it for yourself by declaring an array P(250,3), see how much memory is left, perform a CLR, declare an array P%(250,3) and again see how much memory is left. Quite a difference.

(9) And finally, the biggest saving of all: using data files!

Data conservation

When writing adventure games, somewhere in the program you'll have to have a collection of data that forms the map for the game. Since I usually only use the four cardinal compass points: North, South, East and West, and rely on movements in other directions (such as up and down) being sorted out by the program where appropriate, in a 250 location adventure I'll use a two-dimensional array, like this:

DIMP(249,3)

Remember that the first element of an array is always referred to

as element zero. Now dimensioning an array like this takes up an awful lot of memory: 5009 bytes, in fact. What it also does is to reserve a section of memory at the top of the area set aside for Basic, which will be occupied no matter whether we use all the elements of the array or not.

A typical element in our P(249,3) array will be a two-digit number. Since our room numbers run from 0 to 249, and a lot of the rooms will not allow you to go in particular directions, on average each element will be about two digits long.

Since this data has to be read into the program at some point, you might think that a vast assortment of data statements will be necessary somewhere along the way, and that these will be used to get the data into our array. But just think for a moment. We've got 1000 numbers to type in, each on average about two digits long, so together with line numbers, DATA statements, and associated commas between each data element, incorporating a collection of data statements in the main body of the program will occupy around 3K of memory, which is a lot, especially for a complex adventure program.

The solution is to store all this data on tape or disk, and read it into the program as soon as we start running it. This way, although we still lose our 5009 bytes for the array, we don't lose that additional 3K for however many data statements it takes to define everything. True, they have got to be typed in at some point, but that's just some work on my part: the user never gets to see it. So before even considering the rest of the program, there is one shorter program that consists of thousands of data statements (well, tens of them anyway), and a short program to file it all on to tape or disk, when developing the program.

```
10 DATA 0,1,2,3,4,4,3,4,1,0,5,3 ....
:
:
:
:
1000 OPEN 2,8,2,"0:ROOM DATA,S,W"
1002 FOR I = 0 TO 249:FOR J = 0 TO 3
1004 PRINT #2,P(I,J);CHR$(13);
1006 NEXT J,I
```

This takes each element of the array in turn and stores it on disk in the file ROOM DATA. Note the CHR\$(13), a carriage return,

separating each item of data. Then that program can be deleted and the rest of the adventure sorted out. It's usually more of an adventure getting the program to work than it is solving the thing!

When it comes to playing the completed game, the first part of the program contains a routine to read in all the data from disk, rather like this:

```
2000 OPEN2,8,2,"0:ROOM DATA,S,R"
2002 FORI=0TO249:FORJ=0TO3
2004 INPUT #2,P(I,J)
2006 NEXT J,I
2008 RETURN
```

A saving of around 3K is not to be sniffed at, and any programmer who uses vast amounts of data like this would be well advised to haul everything in from tape or disk, rather than having it stored in the main program. The computer sets aside some memory anyway, there's no need for you to set aside even more.

Unfortunately those short programming examples won't really give you much of an idea about how data is really stored in a sequential file on tape or disk, so let's take a look at some longer programming examples.

```
3000 PRINT"[CLR]INSERT TAPE AND PRESS 'SPACE' WHEN
      READY"
3002 GETPP$:IFPP$<>" "THEN3002
3004 PRINT"OK."
3006 OPEN1,1,1,"DATA"
3007 FORI=1TOLO:PRINT#1,OB$(I);CR$;
3008 NEXTI
3009 PRINT#1,CP;CR$;TB;CR$;GF;CR$;PD;CR$;ZZ;CR$;SC
;CR$;DF;CR$;BR;CR$;NP;CR$;
3010 PRINT#1,P$(10);CR$;P$(60);CR$;P$(50);CR$;P$(2
7);CR$;P$(15);CR$;P$(21);CR$;
3012 PRINT#1,P$(4);CR$;P$(93);CR$;P$(79);CR$;P$(84
);CR$;P$(74);CR$;
3014 PRINT#1,P$(69);CR$;P$(42);CR$;OB$(30);CR$;OB$
(3);CR$;OB$(31);CR$;
3016 PRINT#1,P%(45,1);CR$;P%(10,3);CR$;P%(3,0);CR$
;P%(60,1);CR$;P%(50,3);CR$;
3018 PRINT#1,P%(27,0);CR$;P%(15,1);CR$;P%(15,2);CR
$;P%(21,2);CR$;P%(4,3);CR$;
3020 PRINT#1,P%(93,0);CR$;P%(79,2);CR$;P%(84,2);CR
$;P%(84,3);CR$;P%(74,3);CR$;
3022 PRINT#1,P%(60,0);CR$;P%(60,3);CR$;P%(69,3);CR
```

```

$;P%(42,1);CR$;
3024 CLOSE1
3026 RETURN
3200 PRINT"[CLR]INSERT TAPE AND PRESS 'SPACE' WHEN
      READY"
3202 GETPP$;IFPP$<>" THEN3202
3204 PRINT"OK."
3206 OPEN1,1,0,"DATA"
3207 FORI=1TOLO:INPUT#1,OB$(I)
3208 NEXTI
3209 INPUT#1,CP,TB,GF,PD,ZZ,SC,DF,BR,NP
3210 INPUT#1,P$(10),P$(60),P$(50),P$(27),P$(15),P$
      (21)
3212 INPUT#1,P$(4),P$(93),P$(79),P$(84),P$(74)
3214 INPUT#1,P$(69),P$(42),OB$(30),OB$(3),OB$(31)
3216 INPUT#1,P%(45,1),P%(10,3),P%(3,0),P%(60,1),P%
      (50,3)
3218 INPUT#1,P%(27,0),P%(15,1),P%(15,2),P%(21,2),P
      %(4,3)
3220 INPUT#1,P%(93,0),P%(79,2),P%(84,2),P%(84,3),P
      %(74,3)
3222 INPUT#1,P%(60,0),P%(60,3),P%(69,3),P%(42,1)
3224 CLOSE1
3226 RETURN

```

READY.

```

10 REM *****
11 REM *      EXAMPLE      *
15 REM *  READ AND WRITE A  *
20 REM *  SEQUENTIAL DATA *
25 REM *  FILE ON 1541 DRIVE *
30 REM *****
35 POKE53281,7;POKE53280,9;PRINTCHR$(14);PRINT"[CLR,
      BLK]"
40 DIM A$(25),B(25)
45 OPEN15,8,15;REM OPEN COMMAND CHANNEL
50 PRINT#15,"IO";REM INITIALISE DRIVE
55 GOSUB245;REM CHECK ERROR CHANNEL
60 CR$=CHR$(13)
65 PRINT"[RVS]WRITE TEST FILE"
95 OPEN2,8,2,"@0:TEST FILE,S,W";REM OPEN FILE
105 GOSUB245
110 INPUT"A$,B";A$,B;REM GET DATA FROM USER
115 IFA$="END"THEN135
120 PRINT#2,A$,"STR$(B)CR$;REM WRITE DATA
125 GOSUB245
130 GOTO110
135 CLOSE2;REM CLOSE CHANNEL
170 OPEN2,8,2,"0:TEST FILE,S,R";REM READ FILE
175 GOSUB245

```

```

180 INPUT#2,A$(I),B(I):REM INPUT DATA AGAIN
185 RS=ST:REM STORE DISK STATUS
190 GOSUB245
195 PRINTA$(I),B(I)
200 IFRS=64THEN220:REM CHECK END OF FILE
205 IFRS<>0THEN230:REM CHECK FOR ERROR IN FILE STA
TUS
210 I=I+1
215 GOTO180
220 CLOSE2
225 END
230 PRINT"[RV$]BAD DISK STATUS IS"RS
235 CLOE2
240 END
245 INPUT#15,EN$,EM$,ES$,ET$:REM GET ERROR
285 IFEN$="00"THENRETURN
290 PRINT"[RV$]ERROR ON DISK ":EM$
295 CLOSE15:CLOSE2
300 END

```

READY.

Looking at the tape program first, note how everything is separated by a character CR\$. Since this sub-listing is taken from a real adventure, CR\$ was defined earlier in the program to be equal to a carriage return. Note also how there is no restriction on what we can file away: simple variables, integer variables, parts of a string array, parts of an integer array, anything can be filed on tape.

When reading the data back in again, it is vitally important that it is read in in the same order as it was originally filed out. If not, chaos! To avoid this problem, what I usually do is type in all the lines for storing the data on tape first, and then using the Commodore editing system list (in this program anyway) lines 3009-3022 on the screen, renumber it as 3209-3222. Then, change all the PRINT # statements to INPUT #, and finally just remove all the CR\$ characters. It's a quicker way of entering the lines anyway, and you can virtually guarantee that you'll get everything in the right order.

The format for disk is very similar to the format for tape, the only real difference being the manner in which a file is opened for either reading or writing data. To see this, look at lines 3206 and 3006 in the tape program, and lines 95 and 170 in the disk one.

Storing messages

The following program, alas, applies only to disk. It could in theory be used on tape as well, but in practice it would be a bit ridiculous, and I don't really recommend it.

What we are going to do is look at just one way in which we can store messages and room descriptions on disk, and then, according to the state of the game, retrieve and display those messages on the screen.

```
0 DATA162,6,32,198,255,32,228,255,201,42,208,249,3
2,228,255,205,155,3,208
1 DATA241,76,204,255
2 FORI=900TO922:READA:POKEI,A:NEXT
4 READA:IFA=-1THENSTOP
5 A$=STR$(A):A$=MID$(A$,2):A$="COL"+A$
10 OPEN2,8,2,"@0:"+A$+",S,W"
20 READA$:FORI=1TOLEN(A$):Z$=MID$(A$,I,1)
21 IFZ$<>"^"THENPRINT#2,Z$::NEXT:PRINT#2,CHR$(13);
:GOTO20
22 CLOSE2
23 GOTO4
25 B$="COL3":F2=0
26 OPEN6,8,6,"0:"+B$+",S,R"
27 PRINT:PRINT:INPUT "NO.":A$:F3=F2:IFA$="-1"THENC
LOSE6:END
28 IFLEN(A$)=1THENF$=A$:A$="COL0":GOTO31
29 IFLEN(A$)=2THENF$=MID$(A$,2):A$="COL"+LEFT$(A$,
1):GOTO31
30 F$=RIGHT$(A$,1):A$="COL"+LEFT$(A$,2)
31 F2=VAL(F$):POKE923,F2+48
32 IFF2<=F3THENCLOSE6:OPEN6,8,6,"0:"+A$+",S,R":B$=
A$:GOTO35
33 IFA$<>B$THENCLOSE6:OPEN6,8,6,"0:"+A$+",S,R":B$=
A$
35 SYS900
37 GET#6,Z$
41 IFZ$="J"THENFORI=1TO2000:NEXT:PRINT:GOTO37
42 IFZ$="#"THEN27
43 PRINTZ$:GOTO37
100 DATA21
102 DATA"OYOU ARE READING A STRING USED TO CREATE
104 DATA"AN ADVENTURE TEXT FILE ON DISK.#"
110 DATA"1JUST ANOTHER EXAMPLE OF A STRING, WHICH
112 DATA"COULD BE MADE A LOT LONGER IF I WANTED I
```

```
T TO BE.#^  
120 DATA-1
```

READY.

A careful explanation of this program is called for, since it is an extremely useful utility for the would-be adventure writer.

Explanation

Lines 0-2: machine-code routine to get a character from a file stored on disk. The equivalent of GET #2,A\$, but about ten times as fast. Here it's stored starting at memory location 900, but you can move it anywhere you like really. Just remember to change the SYStem call as appropriate.

Line 4: read some numeric data (in this case from line 100), and if it comes across a '-1' then stop everything.

Line 5: take the string equivalent of the number, take everything from the second character onwards in that string (remembering that STR\$ always creates a leading space at the start of a string), and add the word COL to the string. COL is just used by this program, you can call your files whatever you like, and use whatever numbers you like, but it makes sense to have COL0, COL1, COL2 and so on.

Line 10: open a sequential write file to disk, and call it by whatever is in A\$. In this listing it will be COL21.

Line 20: read a string (in this case from line 102), and then start a loop which will selectively store each character in A\$ in Z\$.

Line 21: if Z\$ is not equal to the up-arrow key (a printer quirk), then print the character to the file and carry on around the loop until it's all finished. Then, print a carriage return to the file and trundle back for another string.

Line 22: we've encountered in the string an up-arrow key, so we want to close this file and start another new one.

Line 23: which is what this line starts.

That's the end of the sequence to store the files on disk. If you look at the data you'll see that each new description or message starts

with an asterisk ('*') followed by a number. In using this program, you could have room descriptions ranging from *0 to *9, but not beyond that, for reasons which we'll come to in a minute. The up-arrow key, which is what the odd symbol at the end of line 112 is meant to be, signifies the end of a file.

Line 25: declare a dummy string to be one of the files on the disk, and set the variable F2.

Line 26: open a sequential file to read data from disk.

Line 27: request the user to enter a number, although in your own adventures this would be the room or message number. Then set the variable F3 and check for the user entering '-1' to end the program run.

Line 28: if only a single digit was entered, then set F\$ equal to A\$, A\$ equal to COL0 (since if only one digit was entered we know that we're looking for one of the first nine entries in file 0), and go to line 31.

Line 29: if only two digits were entered, set F\$ equal to the right hand character in A\$, A\$ equal to COL plus the left hand character, and off to 31 again.

Line 30: assumes more than two digits (the maximum number that this can handle is 999), so F\$ is set to the right most character again, A\$ is set to COL plus the two left hand characters.

Line 31: set F2 equal to the value of the number stored in F\$, and POKE the ASCII value of that number into memory location 923 for use by the machine-code routine. This will also need to be altered if you move the routine.

Line 32: if F2 (set in line 31) is less than or equal to F3 (from line 27) then we have to close one file and open another one, before setting B\$ equal to A\$. In other words, we're either moving to a different file or we're still in the same file but reading an earlier part of it.

Line 33: same sort of thing if A\$ (new file name) is not equal to B\$ (old file name).

Line 35: call to the machine-code routine which gets characters from a file until it finds an asterisk and the number poked into memory

in line 31.

Line 37: conventionally getting a character from disk.

Line 41: if it's a square right hand bracket then pause for a while. You can use this to put some tension in your messages.

Line 42: if it's a hash sign then that's the end of the file.

Line 43: otherwise print the character and go and get another one.

Then there's just the data.

Conclusion

This utility will come into its own in Chapter 8, where it's used in a serious (well, slightly serious) adventure.

Whether you use disk or tape, it's an easy matter to save memory and speed up your Basic by following a few simple rules and using either storage medium to its best advantage.

The next chapter contains some more utilities to help you in your adventure quest!

4

Useful Routines

Introduction

Before compiling these routines, I tried to think of a variety of programs that would be useful to both writers and players of adventure games, i.e. a selection of programs that would improve the user-friendliness of adventure games, while at the same time making them easier to write. These four routines fall nicely into that category, and the first two concern our old friends the function keys.

Key defining

A number of programs have been written that re-define the function keys, but this utility can be made to work from within another program to define each key to produce what you want it to produce, up to a maximum of 128 characters for all eight keys. At 16 characters per key, there should be enough room to perform a variety of actions.

First of all, the Basic loader:

```
10 SE = 49272 : EE = 49414
20 :
30 FOR I = SE TO EE : READ A : POKE I , A : NEXT
40 :
50000 DATA 120,169,144,141,020,003,169,192,141,021
,003,088,096,166
50001 DATA 251,224,255,208,002,230,252,230,251,096
,165,197,197,254
50002 DATA 240,058,201,003,048,054,201,007,016,050
,133,254,201,003
50003 DATA 208,003,024,105,004,024,105,129,174,141
,002,240,003,024
50004 DATA 105,004,133,253,160,000,169,194,133,252
```

```
,132,251,177,251
50005 DATA 197,253,240,019,200,208,247,230,252,165
,252,201,197,208
50006 DATA 239,076,049,234,133,254,076,049,234,200
,208,008,230,252
50007 DATA 165,252,201,197,240,242,177,251,201,013
,208,010,230,198
50008 DATA 166,198,157,119,002,076,213,192,201,000
,240,222,201,133
50009 DATA 048,007,201,141,016,003,076,049,234,032
,210,255,076,213
50010 DATA 192,000,255
```

READY.

This just reads in the 145 bytes of data necessary for the routine (in fact only 143 are used) and POKEs them into memory starting at location 49272. If you prefer to see these things in machine code, and perhaps increase your understanding of how machine code works, type in the following listing:

B*

```
      FC  SR  AC  XR  YR  SP
.:97FE 30 00 00 00 F6
.
C078 78          SEI
C079 A9 90      LDA #$90
C07B 8D 14 03   STA $0314
C07E A9 C0      LDA #$C0
C080 8D 15 03   STA $0315
C083 5B         CLI
C084 60         RTS
C085 A6 FB      LDX $FB
C087 E0 FF      CPX #$FF
C089 D0 02      BNE $C0BD
C08B E6 FC      INC $FC
C08D E6 FB      INC $FB
C08F 60         RTS
C090 A5 C5      LDA $C5
C092 C5 FE      CMP $FE
C094 F0 3A      BEQ $C0D0
C096 C9 03      CMP #$03
C098 30 36      BMI $C0D0
C09A C9 07      CMP #$07
C09C 10 32      BPL $C0D0
C09E 85 FE      STA $FE
C0A0 C9 03      CMP #$03
C0A2 D0 03      BNE $C0A7
C0A4 1B         CLC
C0A5 69 04      ADC #$04
```

COA7	18		CLC
COA8	69	81	ADC #81
COAA	AE	8D 02	LDX \$028D
COAD	F0	03	BEQ \$COB2
COAF	18		CLC
COB0	69	04	ADC #\$04
COB2	85	FD	STA \$FD
COB4	A0	00	LDY #\$00
COB6	A9	C2	LDA #\$C2
COB8	85	FC	STA \$FC
COBA	84	FB	STY \$FB
COBC	B1	FB	LDA (\$FB).Y
COBE	C5	FD	CMP \$FD
COC0	F0	13	BEQ \$COD5
COC2	CB		INY
COC3	D0	F7	BNE \$COBC
COC5	E6	FC	INC \$FC
COC7	A5	FC	LDA \$FC
COC9	C9	C5	CMP #\$C5
COCB	D0	EF	BNE \$COBC
COCd	4C	31 EA	JMP \$EA31
COD0	85	FE	STA \$FE
COD2	4C	31 EA	JMP \$EA31
COD5	CB		INY
COD6	D0	08	BNE \$COE0
COD8	E6	FC	INC \$FC
CODA	A5	FC	LDA \$FC
CODC	C9	C5	CMP #\$C5
CODE	F0	F2	BEQ \$COD2
COE0	B1	FB	LDA (\$FB).Y
COE2	C9	0D	CMP #\$0D
COE4	D0	0A	BNE \$COF0
COE6	E6	C6	INC \$C6
COE8	A6	C6	LDX \$C6
COEA	9D	77 02	STA \$0277.X
COED	4C	D5 C0	JMP \$COD5
COF0	C9	00	CMP #\$00
COF2	F0	DE	BEQ \$COD2
COF4	C9	85	CMP #\$85
COF6	30	07	BMI \$COFF
COF8	C9	8D	CMP #\$8D
COFA	10	03	BPL \$COFF
COFC	4C	31 EA	JMP \$EA31
COFF	20	D2 FF	JSR \$FFD2
C102	4C	D5 C0	JMP \$COD5
C105	00		BRK
.			
.			

If you look at the first few lines of the machine-code listing, you'll see that we're invoking our old acquaintance, the interrupt, to take charge of everything by diverting the hardware interrupt vector (as discussed in Chapter 2) to look at our routine starting at location \$C090 50 times a second before jumping to where it should have gone in memory in the first place.

However, before we can activate the program, some data needs to be placed in memory, and that is the purpose of the following program:

```
0 FORJ=0T07
1 READA:POKE49664+Z+C.A:READB$:FORI=1TOLEN(B$):D=A
  SC(MID$(B$,I,1)):IFD=95THEND=13
2 POKE49665+Z+C.D
3 Z=Z+1:NEXT
4 C=C+1
5 NEXTJ
100 DATA133,NORTH ,134,SOUTH ,135,EAST ,136,WEST ,
137,SCORE ,138,INVENT
110 DATA139.SAVE ,140,LOAD
```

READY.

This looks at the data statements and POKEs the ASCII values of the keys into memory, followed by the ASCII value of the word we want that particular key to produce. In this particular instance we've used a selection of verbs that come in handy when playing adventures. The strange symbol to the right of each word is meant to be the left-arrow key, but my printer was up to its usual tricks and couldn't cope. This is to tell the machine-code program that, when (for example) F1 is pressed we want the word NORTH to appear followed by a carriage return. If you don't want a carriage return performed, just leave out the left-arrow key after the particular word.

Having got the words into memory, the program is activated with a SYS49272 call. After that, whenever F1 is pressed the word NORTH will appear, followed by a carriage return. F3 produces the word SOUTH and a carriage return, and so on. The nice thing about this is not only that it makes the adventure easier to play, but the keywords can be re-defined in the middle of a program to cope with a variety of situations.

To use your own words, just use different data statements, remembering the limit of 128 characters for all eight keys.

Changing colours

One of my very few criticisms of the Zork trilogy of games is that they use the most boring of background, border and text colours. They are in fact the ones displayed by the 64 at power-on, and I for one see enough of those without wanting to spend hours playing a game looking at the same drab colours.

How nice it would have been if they'd incorporated the following short machine-code routine, which allows the user to select which background colour he wants, which border colour, and also which colour he wants the text to be displayed in.

```
10 SE = 49152 : EE = 49245
20 :
30 FOR I = SE TO EE : READ A : POKE I , A : NEXT
40 :
50000 DATA 234,169,013,141,020,003,169,192,141,021
,003,234,096,165
50001 DATA 197,201,004,240,011,201,005,240,013,201
,006,240,015,076
50002 DATA 049,234,238,032,208,076,049,234,238,033
,208,076,049,234
50003 DATA 174,000,193,232,138,162,000,157,000,216
,232,224,250,208
50004 DATA 248,162,000,157,250,216,232,224,250,208
,248,162,000,157
50005 DATA 244,217,232,224,250,208,248,162,000,157
,234,218,232,224
50006 DATA 254,208,248,238,000,193,076,049,234,255
```

READY.

As before, this just POKes the required number of bytes for the routine into memory: all 94 of them in fact.

Again, for machine-code enthusiasts, amateur or professional, here is the machine-code listing:

```
B*
      PC  SR  AC  XR  YR  SP
.: 97FE 30 00 00 00 F6
.
C000 EA                NOP
C001 A9 0D            LDA #$0D
```

```

C003 8D 14 03      STA $0314
C006 A9 C0         LDA #$C0
C008 8D 15 03      STA $0315
C00B EA           NOP
C00C 60           RTS
C00D A5 C5         LDA $C5
C00F C9 04         CMP #$04
C011 F0 0B         BEQ $C01E
C013 C9 05         CMP #$05
C015 F0 0D         BEQ $C024
C017 C9 06         CMP #$06
C019 F0 0F         BEQ $C02A
C01B 4C 31 EA      JMP $EA31
C01E EE 20 D0      INC $D020
C021 4C 31 EA      JMP $EA31
C024 EE 21 D0      INC $D021
C027 4C 31 EA      JMP $EA31
C02A AE 00 C1      LDX $C100
C02D EB           INX
C02E 8A           TXA
C02F A2 00         LDX #$00
C031 9D 00 DB      STA $DB00,X
C034 EB           INX
C035 E0 FA         CPX #$FA
C037 D0 FB         BNE $C031
C039 A2 00         LDX #$00
C03B 9D FA DB      STA $DBFA,X
C03E EB           INX
C03F E0 FA         CPX #$FA
C041 D0 FB         BNE $C03B
C043 A2 00         LDX #$00
C045 9D F4 D9      STA $D9F4,X
C048 EB           INX
C049 E0 FA         CPX #$FA
C04B D0 FB         BNE $C045
C04D A2 00         LDX #$00
C04F 9D EA DA      STA $DAEA,X
C052 EB           INX
C053 E0 FE         CPX #$FE
C055 D0 FB         BNE $C04F
C057 EE 00 C1      INC $C100
C05A 4C 31 EA      JMP $EA31
C05D 00           BRK
.
.

```

Nothing too complicated here. Pressing F1 changes the border colour, F3 changes the background colour, and F5 changes the colour of every character displayed on the screen.

To invoke the routine, use SYS 49152.

Before you use it, however, a word of warning. As the hardware interrupt vector operates at about 50 cycles per second, this means that this routine is checked every fiftieth of a second. Thus if you keep your finger on the key for more than a fraction of a second you'll probably witness about twenty changes in colour before you've finished. So either just stab at the key, or perhaps you'd like to try your hand at changing the program slightly so that it only allows one key press at a time.

Improving screen displays

I came across this routine in a magazine and thought it might be useful, so I duly typed it in. Lo and behold it didn't work, but in amended form it is now a very useful addition to the 64's repertoire of commands.

Most computers have, hidden away somewhere in their Basic, the ability to make a section of the screen flash on and off. Alas, the 64 has no such routine, and if you want to achieve this effect you have to write a cumbersome Basic routine yourself. This is all very well, but it does tie up the machine so that it can't do anything else while flashing. The following program overcomes this fault by using our old friend the interrupt vector again.

```
10 SE = 49152 : EE = 49335
20 :
30 FOR I = SE TO EE : READ A : POKE I , A : NEXT
40 :
50000 DATA 001,003,007,006,010,000,002,000,255,000
,255,017,016,255
50001 DATA 024,040,076,072,178,169,006,141,009,192
,032,138,173,032
50002 DATA 247,183,165,021,208,238,165,020,172,009
,192,217,009,192
50003 DATA 176,228,153,000,192,206,009,192,240,006
,032,253,174,076
50004 DATA 024,192,133,254,120,169,073,141,020,003
,169,192,141,021
50005 DATA 003,088,096,169,216,133,252,169,000,133
,251,168,173,000
50006 DATA 192,208,001,200,185,002,192,133,253,172
,006,192,240,019
50007 DATA 169,040,024,101,251,133,251,176,006,136
,208,244,076,117
```


50008 DATA 192,230,252,208,246,173,005,192,024,101
 ,251,133,251,144
 50009 DATA 002,230,252,165,253,201,016,208,003,173
 ,033,208,145,251
 50010 DATA 200,204,004,192,208,248,206,001,192,240
 ,003,076,049,234
 50011 DATA 173,000,192,073,001,141,000,192,165,254
 ,141,001,192,076
 50012 DATA 049,234,120,169,049,141,020,003,169,234
 ,141,021,003,088
 50013 DATA 096,000

READY.

Or in machine code:

B*

	PC	SR	AC	XR	YR	SP	
..:	97FE	30	00	00	00	F6	
.							
C000	01	03					ORA (#03,X)
C002	07						???
C003	06	0A					ASL #0A
C005	00						BRK
C006	02						???
C007	00						BRK
C008	FF						???
C009	00						BRK
C00A	FF						???
C00B	11	10					ORA (#10),Y
C00D	FF						???
C00E	18						CLC
C00F	28						PLP
C010	4C	48	B2				JMP \$B248
C013	A9	06					LDA #\$06
C015	8D	09	C0				STA \$C009
C018	20	8A	AD				JSR \$AD8A
C01B	20	F7	B7				JSR \$B7F7
C01E	A5	15					LDA \$15
C020	D0	EE					BNE \$C010
C022	A5	14					LDA \$14
C024	AC	09	C0				LDY \$C009
C027	D9	09	C0				CMP \$C009,Y
C02A	B0	E4					BCS \$C010
C02C	99	00	C0				STA \$C000,Y
C02F	CE	09	C0				DEC \$C009
C032	F0	06					BEQ \$C03A
C034	20	FD	AE				JSR \$AEFD
C037	4C	18	C0				JMP \$C018
C03A	B5	FE					STA \$FE

C03C	78		SEI
C03D	A9	49	LDA #\$49
C03F	8D	14 03	STA \$0314
C042	A9	C0	LDA #\$C0
C044	8D	15 03	STA \$0315
C047	58		CLI
C048	60		RTS
C049	A9	DB	LDA #\$DB
C04B	85	FC	STA \$FC
C04D	A9	00	LDA #\$00
C04F	85	FB	STA \$FB
C051	A8		TAY
C052	AD	00 C0	LDA \$C000
C055	D0	01	BNE \$C058
C057	C8		INY
C058	B9	02 C0	LDA \$C002,Y
C05B	85	FD	STA \$FD
C05D	AC	06 C0	LDY \$C006
C060	F0	13	BEQ \$C075
C062	A9	28	LDA #\$28
C064	18		CLC
C065	65	FB	ADC \$FB
C067	85	FB	STA \$FB
C069	B0	06	BCS \$C071
C06B	88		DEY
C06C	D0	F4	BNE \$C062
C06E	4C	75 C0	JMP \$C075
C071	E6	FC	INC \$FC
C073	D0	F6	BNE \$C06B
C075	AD	05 C0	LDA \$C005
C078	18		CLC
C079	65	FB	ADC \$FB
C07B	85	FB	STA \$FB
C07D	90	02	BCC \$C081
C07F	E6	FC	INC \$FC
C081	A5	FD	LDA \$FD
C083	C9	10	CMP #\$10
C085	D0	03	BNE \$C08A
C087	AD	21 D0	LDA \$D021
C08A	91	FB	STA (\$FB),Y
C08C	C8		INY
C08D	CC	04 C0	CPY \$C004
C090	D0	F8	BNE \$C08A
C092	CE	01 C0	DEC \$C001
C095	F0	03	BEQ \$C09A
C097	4C	31 EA	JMP \$EA31
C09A	AD	00 C0	LDA \$C000
C09D	49	01	EOR #\$01
C09F	8D	00 C0	STA \$C000
C0A2	A5	FE	LDA \$FE
C0A4	8D	01 C0	STA \$C001

COA7	4C	31	EA	JMP	\$EA31
COAA	7B			SEI	
COAB	A9	31		LDA	#\$31
COAD	8D	14	03	STA	\$0314
COB0	A9	EA		LDA	#\$EA
COB2	8D	15	03	STA	\$0315
COB5	5B			CLI	
COB6	60			RTS	
COB7	00			BRK	
.					
.					

As usual, the first program POKes the various values into memory, the second one lets you see how the routine works. To use it, we must observe the following syntax:

SYS(49171)a,b,c,d,e,f

where:

a = first row (0-24) of text to be flashed.

b = first column (0-39) of text to be flashed.

c = number of characters after the first one that are going to be flashed.

d = first colour that text will be displayed in.

e = second colour that text will be displayed in before alternating between first and second colours.

f = duration of flash in 50ths of a second. Thus if f = 50 the flash will occur once every second, or thereabouts.

To turn the thing off you must use **SYS49322**.

The routine can be used to highlight prompts, message displays, warnings, and so on. With a little bit of thought you can get some quite clever effects.

And for the programmer

So far we've concentrated on helping out the player of adventures more than the programmer, making displays brighter and

changeable, and keywords easy to use. But now, for the Basic adventure programmer, here is a renumber program. Renumbering a large Basic adventure can sometimes save up to half a kilobyte of memory, which comes in handy when you start pushing the machine's memory to the limits.

```
10 SE = 50000 : EE = 51199
20 :
30 FOR I = SE TO EE : READ A : POKE I , A : NEXT
40 :
50000 DATA 032,229,199,032,161,199,032,138,199,032
,037,199,160,000
50001 DATA 132,249,132,250,132,158,132,180,132,164
,120,173,012,200
50002 DATA 013,013,200,240,105,056,173,010,200,237
,012,200,141,000
50003 DATA 200,173,011,200,237,013,200,141,001,200
,162,006,032,013
50004 DATA 199,165,095,133,247,166,096,134,248,172
,008,200,132,020
50005 DATA 172,009,200,132,021,133,095,134,096,160
,001,177,095,240
50006 DATA 083,200,200,177,095,145,253,133,170,136
,177,095,145,253
50007 DATA 133,169,165,021,197,170,144,066,208,006
,165,020,197,169
50008 DATA 144,058,024,160,000,173,000,200,109,012
,200,145,253,141
50009 DATA 000,200,173,001,200,109,013,200,201,250
,144,006,032,037
50010 DATA 199,076,072,178,200,145,253,141,001,200
,230,249,208,002
50011 DATA 230,250,032,235,198,177,095,170,136,177
,095,024,144,163
50012 DATA 169,255,133,164,165,249,005,250,208,002
,088,096,165,095
50013 DATA 133,251,165,096,133,252,173,007,200,205
,011,200,208,008
50014 DATA 173,006,200,205,010,200,240,091,176,045
,169,002,133,158
50015 DATA 162,010,032,013,199,032,107,199,165,252
,197,177,208,004
50016 DATA 165,251,197,176,176,065,162,000,230,158
,230,158,032,013
50017 DATA 199,176,007,162,000,032,096,199,240,065
,024,144,143,162
50018 DATA 000,032,013,199,102,165,032,107,199,165
,177,197,248,208
50019 DATA 004,165,176,197,247,176,004,036,165,048
,227,230,158,162
```

50020 DATA 010,144,205,032,013,199,176,216,162,071
 ,032,096,199,208
 50021 DATA 209,173,001,200,197,170,208,005,173,000
 ,200,197,169,144
 50022 DATA 004,036,164,016,191,165,158,201,003,144
 ,040,165,251,229
 50023 DATA 247,133,167,165,252,229,248,133,168,165
 ,045,133,090,165
 50024 DATA 046,133,091,032,171,198,176,160,165,045
 ,133,088,165,046
 50025 DATA 133,089,032,191,163,166,158,224,003,144
 ,093,208,003,032
 50026 DATA 075,199,160,000,177,247,145,176,230,176
 ,208,002,230,177
 50027 DATA 230,247,208,002,230,248,162,202,032,026
 ,199,144,233,165
 50028 DATA 247,229,167,133,247,165,248,229,168,133
 ,248,177,251,145
 50029 DATA 247,230,251,208,002,230,252,230,247,208
 ,002,230,248,162
 50030 DATA 000,032,026,199,208,233,032,050,199,032
 ,051,165,162,131
 50031 DATA 165,158,201,003,240,003,032,050,199,032
 ,050,199,165,176
 50032 DATA 133,247,165,177,133,248,032,129,199,032
 ,138,199,032,037
 50033 DATA 199,032,147,199,160,000,177,253,072,200
 ,177,253,200,200
 50034 DATA 145,247,136,104,145,247,136,177,247,170
 ,136,177,247,133
 50035 DATA 247,134,248,032,235,198,032,116,199,208
 ,221,032,154,199
 50036 DATA 169,000,133,096,169,043,133,095,160,000
 ,177,095,170,200
 50037 DATA 177,095,134,095,133,096,177,095,240,041
 ,160,003,132,015
 50038 DATA 200,177,095,240,231,201,034,208,008,165
 ,015,073,255,133
 50039 DATA 015,208,239,036,015,048,235,201,143,240
 ,211,162,006,221
 50040 DATA 249,199,240,011,202,208,248,240,219,032
 ,037,199,076,116
 50041 DATA 164,024,152,101,095,133,122,133,090,166
 ,096,144,001,232
 50042 DATA 134,123,134,091,032,115,000,144,010,201
 ,171,240,004,201
 50043 DATA 164,208,183,240,059,032,107,169,160,003
 ,032,194,198,176
 50044 DATA 073,160,001,032,194,198,176,017,165,095
 ,072,165,096,072
 50045 DATA 032,019,166,104,133,096,104,133,095,176

,022,160,002,177
 50046 DATA 095,170,200,177,095,032,205,189,169,032
 ,032,210,255,169
 50047 DATA 249,162,255,208,038,032,121,000,170,056
 ,165,122,229,095
 50048 DATA 168,138,201,044,240,153,201,171,240,250
 ,201,164,240,246
 50049 DATA 170,076,103,197,160,000,032,147,199,177
 ,253,170,200,177
 50050 DATA 253,133,098,134,099,032,154,199,162,144
 ,056,032,073,188
 50051 DATA 032,221,189,217,000,001,240,003,200,208
 ,248,165,122,229
 50052 DATA 090,133,159,152,056,229,159,133,146,240
 ,098,162,077,032
 50053 DATA 249,198,036,146,048,043,165,045,133,088
 ,165,046,133,089
 50054 DATA 162,000,032,249,198,162,002,032,249,198
 ,160,000,177,088
 50055 DATA 164,146,145,088,165,088,208,002,198,089
 ,198,088,162,045
 50056 DATA 032,064,199,208,233,240,046,165,122,133
 ,088,165,123,133
 50057 DATA 089,162,000,032,249,198,162,002,032,249
 ,198,056,169,000
 50058 DATA 229,146,133,146,164,146,177,088,160,000
 ,145,088,230,088
 50059 DATA 208,002,230,089,162,000,032,064,199,208
 ,235,160,001,185
 50060 DATA 000,001,240,005,145,090,200,208,246,165
 ,146,240,003,032
 50061 DATA 051,165,076,231,197,024,165,045,101,167
 ,168,165,046,101
 50062 DATA 168,197,056,208,002,196,055,176,004,132
 ,045,133,046,096
 50063 DATA 032,138,199,032,147,199,132,180,032,129
 ,199,164,180,165
 50064 DATA 021,209,253,144,017,208,007,136,165,020
 ,209,253,240,009
 50065 DATA 032,235,198,032,116,199,208,231,024,032
 ,154,199,096,024
 50066 DATA 165,253,105,004,133,253,165,254,105,000
 ,133,254,096,024
 50067 DATA 181,045,101,146,149,045,160,255,036,146
 ,048,001,200,152
 50068 DATA 117,046,149,046,096,189,000,200,133,020
 ,189,001,200,133
 50069 DATA 021,076,019,166,181,046,197,252,208,004
 ,181,045,197,251
 50070 DATA 096,165,045,133,047,133,049,165,046,133
 ,048,133,050,096

50071 DATA 056,181,045,229,167,149,045,181,046,229
 ,168,149,046,096
 50072 DATA 165,089,213,046,208,004,165,088,213,045
 ,096,162,000,032
 50073 DATA 082,199,162,004,024,181,247,101,167,149
 ,247,181,248,101
 50074 DATA 168,149,248,096,165,096,213,177,208,004
 ,165,095,213,176
 50075 DATA 096,165,095,133,176,165,096,133,177,096
 ,165,189,208,002
 50076 DATA 198,190,198,189,165,189,005,190,096,165
 ,249,133,189,165
 50077 DATA 250,133,190,096,169,000,133,253,169,160
 ,133,254,096,165
 50078 DATA 001,041,254,133,001,096,165,001,009,001
 ,133,001,096,032
 50079 DATA 006,226,032,218,199,140,010,200,141,011
 ,200,032,006,226
 50080 DATA 032,218,199,140,012,200,141,013,200,032
 ,006,226,032,218
 50081 DATA 199,140,006,200,141,007,200,032,006,226
 ,169,171,032,255
 50082 DATA 174,032,017,226,032,221,199,140,008,200
 ,141,009,200,096
 50083 DATA 032,014,226,032,107,169,165,021,164,020
 ,096,162,007,189
 50084 DATA 241,199,157,006,200,202,016,247,096,000
 ,000,255,249,100
 50085 DATA 000,010,000,000,155,138,167,137,141,203

READY.

Again, for those of you who want to see how it works rather than just typing it in and using it:

B*

	PC	SR	AC	XR	YR	SP	
.	:	97FE	30	00	00	00	F6
.							
C350	20	E5	C7				JSR \$C7E5
C353	20	A1	C7				JSR \$C7A1
C356	20	8A	C7				JSR \$C78A
C359	20	25	C7				JSR \$C725
C35C	A0	00					LDY #\$00
C35E	84	F9					STY \$F9
C360	84	FA					STY \$FA
C362	84	9E					STY \$9E
C364	84	B4					STY \$B4
C366	84	A4					STY \$A4
C368	78						SEI
C369	AD	0C	C8				LDA \$C80C

C36C	0D	0D	C8	ORA	\$C80D
C36F	F0	69		BEQ	\$C3DA
C371	38			SEC	
C372	AD	0A	C8	LDA	\$C80A
C375	ED	0C	C8	SBC	\$C80C
C378	8D	00	C8	STA	\$C800
C37B	AD	0B	C8	LDA	\$C80B
C37E	ED	0D	C8	SBC	\$C80D
C381	8D	01	C8	STA	\$C801
C384	A2	06		LDX	#\$06
C386	20	0D	C7	JSR	\$C70D
C389	A5	5F		LDA	\$5F
C38B	85	F7		STA	\$F7
C38D	A6	60		LDX	\$60
C38F	86	F8		STX	\$F8
C391	AC	08	C8	LDY	\$C808
C394	84	14		STY	\$14
C396	AC	09	C8	LDY	\$C809
C399	84	15		STY	\$15
C39B	85	5F		STA	\$5F
C39D	86	60		STX	\$60
C39F	A0	01		LDY	#\$01
C3A1	B1	5F		LDA	(\$5F),Y
C3A3	F0	53		BEQ	\$C3F8
C3A5	C8			INY	
C3A6	C8			INY	
C3A7	B1	5F		LDA	(\$5F),Y
C3A9	91	FD		STA	(\$FD),Y
C3AB	85	AA		STA	\$AA
C3AD	88			DEY	
C3AE	B1	5F		LDA	(\$5F),Y
C3B0	91	FD		STA	(\$FD),Y
C3B2	85	A9		STA	\$A9
C3B4	A5	15		LDA	\$15
C3B6	C5	AA		CMP	\$AA
C3B8	90	42		BCC	\$C3FC
C3BA	D0	06		BNE	\$C3C2
C3BC	A5	14		LDA	\$14
C3BE	C5	A9		CMP	\$A9
C3C0	90	3A		BCC	\$C3FC
C3C2	18			CLC	
C3C3	A0	00		LDY	#\$00
C3C5	AD	00	C8	LDA	\$C800
C3C8	6D	0C	C8	ADC	\$C80C
C3CB	91	FD		STA	(\$FD),Y
C3CD	8D	00	C8	STA	\$C800
C3D0	AD	01	C8	LDA	\$C801
C3D3	6D	0D	C8	ADC	\$C80D
C3D6	C9	FA		CMP	#\$FA
C3D8	90	06		BCC	\$C3E0
C3DA	20	25	C7	JSR	\$C725

C3DD	4C	4B	B2	JMP	\$B24B
C3E0	C8			INY	
C3E1	91	FD		STA	(\$FD).Y
C3E3	8D	01	C8	STA	\$C801
C3E6	E6	F9		INC	\$F9
C3E8	D0	02		BNE	\$C3EC
C3EA	E6	FA		INC	\$FA
C3EC	20	EB	C6	JSR	\$C6EB
C3EF	B1	5F		LDA	(\$5F).Y
C3F1	AA			TAX	
C3F2	8B			DEY	
C3F3	B1	5F		LDA	(\$5F).Y
C3F5	18			CLC	
C3F6	90	A3		BCC	\$C39B
C3F8	A9	FF		LDA	#\$FF
C3FA	85	A4		STA	\$A4
C3FC	A5	F9		LDA	\$F9
C3FE	05	FA		ORA	\$FA
C400	D0	02		BNE	\$C404
C402	58			CLI	
C403	60			RTS	
C404	A5	5F		LDA	\$5F
C406	85	FB		STA	\$FB
C408	A5	60		LDA	\$60
C40A	85	FC		STA	\$FC
C40C	AD	07	C8	LDA	\$C807
C40F	CD	0B	C8	CMP	\$C80B
C412	D0	0B		BNE	\$C41C
C414	AD	06	C8	LDA	\$C806
C417	CD	0A	C8	CMP	\$C80A
C41A	F0	5B		BEQ	\$C477
C41C	B0	2D		BCS	\$C44B
C41E	A9	02		LDA	#\$02
C420	85	9E		STA	\$9E
C422	A2	0A		LDX	#\$0A
C424	20	0D	C7	JSR	\$C70D
C427	20	6B	C7	JSR	\$C76B
C42A	A5	FC		LDA	\$FC
C42C	C5	B1		CMP	\$B1
C42E	D0	04		BNE	\$C434
C430	A5	FB		LDA	\$FB
C432	C5	B0		CMP	\$B0
C434	B0	41		BCS	\$C477
C436	A2	00		LDX	#\$00
C438	E6	9E		INC	\$9E
C43A	E6	9E		INC	\$9E
C43C	20	0D	C7	JSR	\$C70D
C43F	B0	07		BCS	\$C44B
C441	A2	00		LDX	#\$00
C443	20	60	C7	JSR	\$C760
C446	F0	41		BEQ	\$C489

C448	18		CLC
C449	90	8F	BCC \$C3DA
C44B	A2	00	LDX #\$00
C44D	20	0D	JSR \$C70D
C450	66	A5	ROR \$A5
C452	20	6B	JSR \$C76B
C455	A5	B1	LDA \$B1
C457	C5	F8	CMP \$F8
C459	D0	04	BNE \$C45F
C45B	A5	B0	LDA \$B0
C45D	C5	F7	CMP \$F7
C45F	B0	04	BCS \$C465
C461	24	A5	BIT \$A5
C463	30	E3	BMI \$C448
C465	E6	9E	INC \$9E
C467	A2	0A	LDX #\$0A
C469	90	CD	BCC \$C438
C46B	20	0D	JSR \$C70D
C46E	B0	D8	BCS \$C448
C470	A2	47	LDX #\$47
C472	20	60	JSR \$C760
C475	D0	D1	BNE \$C448
C477	AD	01	LDA \$C801
C47A	C5	AA	CMP \$AA
C47C	D0	05	BNE \$C483
C47E	AD	00	LDA \$C800
C481	C5	A9	CMP \$A9
C483	90	04	BCC \$C489
C485	24	A4	BIT \$A4
C487	10	BF	BPL \$C448
C489	A5	9E	LDA \$9E
C48B	C9	03	CMP #\$03
C48D	90	28	BCC \$C4B7
C48F	A5	FB	LDA \$FB
C491	E5	F7	SBC \$F7
C493	85	A7	STA \$A7
C495	A5	FC	LDA \$FC
C497	E5	F8	SBC \$F8
C499	85	AB	STA \$AB
C49B	A5	2D	LDA \$2D
C49D	85	5A	STA \$5A
C49F	A5	2E	LDA \$2E
C4A1	85	5B	STA \$5B
C4A3	20	AB	JSR \$C6AB
C4A6	B0	A0	BCS \$C448
C4A8	A5	2D	LDA \$2D
C4AA	85	58	STA \$58
C4AC	A5	2E	LDA \$2E
C4AE	85	59	STA \$59
C4B0	20	BF	JSR \$A3BF
C4B3	A6	9E	LDX \$9E

C4B5	E0	03		CPX	#\$03
C4B7	90	5D		BCC	#\$C516
C4B9	D0	03		BNE	#\$C4BE
C4BB	20	4B	C7	JSR	#\$C74B
C4BE	A0	00		LDY	#\$00
C4C0	B1	F7		LDA	(\$F7),Y
C4C2	91	B0		STA	(\$B0),Y
C4C4	E6	B0		INC	#\$B0
C4C6	D0	02		BNE	#\$C4CA
C4C8	E6	B1		INC	#\$B1
C4CA	E6	F7		INC	#\$F7
C4CC	D0	02		BNE	#\$C4D0
C4CE	E6	F8		INC	#\$F8
C4D0	A2	CA		LDX	#\$CA
C4D2	20	1A	C7	JSR	#\$C71A
C4D5	90	E9		BCC	#\$C4C0
C4D7	A5	F7		LDA	#\$F7
C4D9	E5	A7		SBC	#\$A7
C4DB	85	F7		STA	#\$F7
C4DD	A5	F8		LDA	#\$F8
C4DF	E5	AB		SBC	#\$AB
C4E1	85	F8		STA	#\$F8
C4E3	B1	FB		LDA	(\$FB),Y
C4E5	91	F7		STA	(\$F7),Y
C4E7	E6	FB		INC	#\$FB
C4E9	D0	02		BNE	#\$C4ED
C4EB	E6	FC		INC	#\$FC
C4ED	E6	F7		INC	#\$F7
C4EF	D0	02		BNE	#\$C4F3
C4F1	E6	F8		INC	#\$F8
C4F3	A2	00		LDX	#\$00
C4F5	20	1A	C7	JSR	#\$C71A
C4F8	D0	E9		BNE	#\$C4E3
C4FA	20	32	C7	JSR	#\$C732
C4FD	20	33	A5	JSR	#\$A533
C500	A2	83		LDX	#\$83
C502	A5	9E		LDA	#\$9E
C504	C9	03		CMP	#\$03
C506	F0	03		BEQ	#\$C50B
C508	20	32	C7	JSR	#\$C732
C50B	20	32	C7	JSR	#\$C732
C50E	A5	B0		LDA	#\$B0
C510	85	F7		STA	#\$F7
C512	A5	B1		LDA	#\$B1
C514	85	F8		STA	#\$F8
C516	20	81	C7	JSR	#\$C781
C519	20	8A	C7	JSR	#\$C78A
C51C	20	25	C7	JSR	#\$C725
C51F	20	93	C7	JSR	#\$C793
C522	A0	00		LDY	#\$00
C524	B1	FD		LDA	(\$FD),Y

C526	48		PHA
C527	C8		INY
C528	B1	FD	LDA (\$FD),Y
C52A	C8		INY
C52B	C8		INY
C52C	91	F7	STA (\$F7),Y
C52E	88		DEY
C52F	68		PLA
C530	91	F7	STA (\$F7),Y
C532	88		DEY
C533	B1	F7	LDA (\$F7),Y
C535	AA		TAX
C536	88		DEY
C537	B1	F7	LDA (\$F7),Y
C539	85	F7	STA \$F7
C53B	86	F8	STX \$F8
C53D	20	EB C6	JSR \$C6EB
C540	20	74 C7	JSR \$C774
C543	D0	DD	BNE \$C522
C545	20	9A C7	JSR \$C79A
C548	A9	00	LDA #\$00
C54A	85	60	STA \$60
C54C	A9	2B	LDA #\$2B
C54E	85	5F	STA \$5F
C550	A0	00	LDY #\$00
C552	B1	5F	LDA (\$5F),Y
C554	AA		TAX
C555	C8		INY
C556	B1	5F	LDA (\$5F),Y
C558	86	5F	STX \$5F
C55A	85	60	STA \$60
C55C	B1	5F	LDA (\$5F),Y
C55E	F0	29	BEQ \$C589
C560	A0	03	LDY #\$03
C562	84	0F	STY \$0F
C564	C8		INY
C565	B1	5F	LDA (\$5F),Y
C567	F0	E7	BEQ \$C550
C569	C9	22	CMP #\$22
C56B	D0	0B	BNE \$C575
C56D	A5	0F	LDA \$0F
C56F	49	FF	EOR #\$FF
C571	85	0F	STA \$0F
C573	D0	EF	BNE \$C564
C575	24	0F	BIT \$0F
C577	30	EB	BMI \$C564
C579	C9	8F	CMP #\$8F
C57B	F0	D3	BEQ \$C550
C57D	A2	06	LDX #\$06
C57F	DD	F9 C7	CMP \$C7F9.X
C582	F0	0B	BEQ \$C58F

C584	CA		DEX
C585	D0	F8	BNE \$C57F
C587	F0	DB	BEQ \$C564
C589	20	25 C7	JSR \$C725
C58C	4C	74 A4	JMP \$A474
C58F	18		CLC
C590	98		TYA
C591	65	5F	ADC \$5F
C593	85	7A	STA \$7A
C595	85	5A	STA \$5A
C597	A6	60	LDX \$60
C599	90	01	BCC \$C59C
C59B	E8		INX
C59C	86	7B	STX \$7B
C59E	86	5B	STX \$5B
C5A0	20	73 00	JSR \$0073
C5A3	90	0A	BCC \$C5AF
C5A5	C9	AB	CMP #\$AB
C5A7	F0	04	BEQ \$C5AD
C5A9	C9	A4	CMP #\$A4
C5AB	D0	B7	BNE \$C564
C5AD	F0	3B	BEQ \$C5EA
C5AF	20	6B A9	JSR \$A96B
C5B2	A0	03	LDY #\$03
C5B4	20	C2 C6	JSR \$C6C2
C5B7	B0	49	BCS \$C602
C5B9	A0	01	LDY #\$01
C5BB	20	C2 C6	JSR \$C6C2
C5BE	B0	11	BCS \$C5D1
C5C0	A5	5F	LDA \$5F
C5C2	48		PHA
C5C3	A5	60	LDA \$60
C5C5	48		PHA
C5C6	20	13 A6	JSR \$A613
C5C9	68		PLA
C5CA	85	60	STA \$60
C5CC	68		PLA
C5CD	85	5F	STA \$5F
C5CF	B0	16	BCS \$C5E7
C5D1	A0	02	LDY #\$02
C5D3	B1	5F	LDA (\$5F),Y
C5D5	AA		TAX
C5D6	C8		INX
C5D7	B1	5F	LDA (\$5F),Y
C5D9	20	CD BD	JSR \$BDCD
C5DC	A9	20	LDA #\$20
C5DE	20	D2 FF	JSR \$FFD2
C5E1	A9	F9	LDA #\$F9
C5E3	A2	FF	LDX \$FF
C5E5	D0	26	BNE \$C60D
C5E7	20	79 00	JSR \$0079

C5EA	AA		TAX
C5EB	38		SEC
C5EC	A5	7A	LDA \$7A
C5EE	E5	5F	SBC \$5F
C5F0	A8		TAY
C5F1	8A		TXA
C5F2	C9	2C	CMP #\$2C
C5F4	F0	99	BEQ \$C58F
C5F6	C9	AB	CMP #\$AB
C5F8	F0	FA	BEQ \$C5F4
C5FA	C9	A4	CMP #\$A4
C5FC	F0	F6	BEQ \$C5F4
C5FE	AA		TAX
C5FF	4C	67 C5	JMP \$C567
C602	A0	00	LDY #\$00
C604	20	93 C7	JSR \$C793
C607	B1	FD	LDA (\$FD),Y
C609	AA		TAX
C60A	C8		INY
C60B	B1	FD	LDA (\$FD),Y
C60D	85	62	STA \$62
C60F	86	63	STX \$63
C611	20	9A C7	JSR \$C79A
C614	A2	90	LDX #\$90
C616	38		SEC
C617	20	49 BC	JSR \$BC49
C61A	20	DD BD	JSR \$BDDD
C61D	D9	00 01	CMP \$0100,Y
C620	F0	03	BEQ \$C625
C622	C8		INY
C623	D0	F8	BNE \$C61D
C625	A5	7A	LDA \$7A
C627	E5	5A	SBC \$5A
C629	85	9F	STA \$9F
C62B	98		TYA
C62C	38		SEC
C62D	E5	9F	SBC \$9F
C62F	85	92	STA \$92
C631	F0	62	BEQ \$C695
C633	A2	4D	LDX #\$4D
C635	20	F9 C6	JSR \$C6F9
C638	24	92	BIT \$92
C63A	30	2B	BMI \$C667
C63C	A5	2D	LDA \$2D
C63E	85	58	STA \$58
C640	A5	2E	LDA \$2E
C642	85	59	STA \$59
C644	A2	00	LDX #\$00
C646	20	F9 C6	JSR \$C6F9
C649	A2	02	LDX #\$02
C64B	20	F9 C6	JSR \$C6F9

C64E	A0	00		LDY	#\$00
C650	B1	58		LDA	(\$58),Y
C652	A4	92		LDY	\$92
C654	91	58		STA	(\$58),Y
C656	A5	58		LDA	\$58
C658	D0	02		BNE	\$C65C
C65A	C6	59		DEC	\$59
C65C	C6	58		DEC	\$58
C65E	A2	2D		LDX	##2D
C660	20	40	C7	JSR	\$C740
C663	D0	E9		BNE	\$C64E
C665	F0	2E		BEQ	\$C695
C667	A5	7A		LDA	\$7A
C669	85	58		STA	\$58
C66B	A5	7B		LDA	\$7B
C66D	85	59		STA	\$59
C66F	A2	00		LDX	##00
C671	20	F9	C6	JSR	\$C6F9
C674	A2	02		LDX	##02
C676	20	F9	C6	JSR	\$C6F9
C679	38			SEC	
C67A	A9	00		LDA	##00
C67C	E5	92		SBC	\$92
C67E	85	92		STA	\$92
C680	A4	92		LDY	\$92
C682	B1	58		LDA	(\$58),Y
C684	A0	00		LDY	##00
C686	91	58		STA	(\$58),Y
C688	E6	58		INC	\$58
C68A	D0	02		BNE	\$C68E
C68C	E6	59		INC	\$59
C68E	A2	00		LDX	##00
C690	20	40	C7	JSR	\$C740
C693	D0	EB		BNE	\$C680
C695	A0	01		LDY	##01
C697	B9	00	01	LDA	\$0100,Y
C69A	F0	05		BEQ	\$C6A1
C69C	91	5A		STA	(\$5A),Y
C69E	C8			INY	
C69F	D0	F6		BNE	\$C697
C6A1	A5	92		LDA	\$92
C6A3	F0	03		BEQ	\$C6A8
C6A5	20	33	A5	JSR	\$A533
C6A8	4C	E7	C5	JMP	\$C5E7
C6AB	18			CLC	
C6AC	A5	2D		LDA	\$2D
C6AE	65	A7		ADC	\$A7
C6B0	A8			TAY	
C6B1	A5	2E		LDA	\$2E
C6B3	65	A8		ADC	\$A8

C6B5	C5	38		CMP	\$38
C6B7	D0	02		BNE	\$C6BB
C6B9	C4	37		CPY	\$37
C6BB	B0	04		BCS	\$C6C1
C6BD	B4	2D		STY	\$2D
C6BF	B5	2E		STA	\$2E
C6C1	60			RTS	
C6C2	20	8A	C7	JSR	\$C78A
C6C5	20	93	C7	JSR	\$C793
C6C8	B4	B4		STY	\$B4
C6CA	20	81	C7	JSR	\$C781
C6CD	A4	B4		LDY	\$B4
C6CF	A5	15		LDA	\$15
C6D1	D1	FD		CMP	(\$FD),Y
C6D3	90	11		BCC	\$C6E6
C6D5	D0	07		BNE	\$C6DE
C6D7	88			DEY	
C6D8	A5	14		LDA	\$14
C6DA	D1	FD		CMP	(\$FD),Y
C6DC	F0	09		BEQ	\$C6E7
C6DE	20	EB	C6	JSR	\$C6EB
C6E1	20	74	C7	JSR	\$C774
C6E4	D0	E7		BNE	\$C6CD
C6E6	18			CLC	
C6E7	20	9A	C7	JSR	\$C79A
C6EA	60			RTS	
C6EB	18			CLC	
C6EC	A5	FD		LDA	\$FD
C6EE	69	04		ADC	##04
C6F0	85	FD		STA	\$FD
C6F2	A5	FE		LDA	\$FE
C6F4	69	00		ADC	##00
C6F6	85	FE		STA	\$FE
C6F8	60			RTS	
C6F9	18			CLC	
C6FA	B5	2D		LDA	\$2D.X
C6FC	65	92		ADC	\$92
C6FE	95	2D		STA	\$2D.X
C700	A0	FF		LDY	##FF
C702	24	92		BIT	\$92
C704	30	01		BMI	\$C707
C706	C8			INY	
C707	98			TYA	
C708	75	2E		ADC	\$2E.X
C70A	95	2E		STA	\$2E.X
C70C	60			RTS	
C70D	BD	00	C8	LDA	\$C800.X
C710	85	14		STA	\$14
C712	BD	01	C8	LDA	\$C801.X
C715	85	15		STA	\$15
C717	4C	13	A6	JMP	\$A613

C71A	B5	2E		LDA	\$2E,X
C71C	C5	FC		CMP	\$FC
C71E	D0	04		BNE	\$C724
C720	B5	2D		LDA	\$2D,X
C722	C5	FB		CMP	\$FB
C724	60			RTS	
C725	A5	2D		LDA	\$2D
C727	B5	2F		STA	\$2F
C729	B5	31		STA	\$31
C72B	A5	2E		LDA	\$2E
C72D	B5	30		STA	\$30
C72F	B5	32		STA	\$32
C731	60			RTS	
C732	38			SEC	
C733	B5	2D		LDA	\$2D,X
C735	E5	A7		SBC	\$A7
C737	95	2D		STA	\$2D,X
C739	B5	2E		LDA	\$2E,X
C73B	E5	A8		SBC	\$A8
C73D	95	2E		STA	\$2E,X
C73F	60			RTS	
C740	A5	59		LDA	\$59
C742	D5	2E		CMP	\$2E,X
C744	D0	04		BNE	\$C74A
C746	A5	58		LDA	\$58
C748	D5	2D		CMP	\$2D,X
C74A	60			RTS	
C74B	A2	00		LDX	#\$00
C74D	20	52	C7	JSR	\$C752
C750	A2	04		LDX	#\$04
C752	18			CLC	
C753	B5	F7		LDA	\$F7,X
C755	65	A7		ADC	\$A7
C757	95	F7		STA	\$F7,X
C759	B5	F8		LDA	\$F8,X
C75B	65	A8		ADC	\$A8
C75D	95	F8		STA	\$F8,X
C75F	60			RTS	
C760	A5	60		LDA	\$60
C762	D5	B1		CMP	\$B1,X
C764	D0	04		BNE	\$C76A
C766	A5	5F		LDA	\$5F
C768	D5	B0		CMP	\$B0,X
C76A	60			RTS	
C76B	A5	5F		LDA	\$5F
C76D	B5	B0		STA	\$B0
C76F	A5	60		LDA	\$60
C771	B5	B1		STA	\$B1
C773	60			RTS	
C774	A5	BD		LDA	\$BD
C776	D0	02		BNE	\$C77A

C77B	C6	BE	DEC	\$BE
C77A	C6	BD	DEC	\$BD
C77C	A5	BD	LDA	\$BD
C77E	05	BE	ORA	\$BE
C780	60		RTS	
C781	A5	F9	LDA	\$F9
C783	85	BD	STA	\$BD
C785	A5	FA	LDA	\$FA
C787	85	BE	STA	\$BE
C789	60		RTS	
C78A	A9	00	LDA	#\$00
C78C	85	FD	STA	\$FD
C78E	A9	A0	LDA	#\$A0
C790	85	FE	STA	\$FE
C792	60		RTS	
C793	A5	01	LDA	\$01
C795	29	FE	AND	#\$FE
C797	85	01	STA	\$01
C799	60		RTS	
C79A	A5	01	LDA	\$01
C79C	09	01	ORA	#\$01
C79E	85	01	STA	\$01
C7A0	60		RTS	
C7A1	20	06 E2	JSR	\$E206
C7A4	20	DA C7	JSR	\$C7DA
C7A7	8C	0A C8	STY	\$C80A
C7AA	8D	0B C8	STA	\$C80B
C7AD	20	06 E2	JSR	\$E206
C7B0	20	DA C7	JSR	\$C7DA
C7B3	8C	0C C8	STY	\$C80C
C7B6	8D	0D C8	STA	\$C80D
C7B9	20	06 E2	JSR	\$E206
C7BC	20	DA C7	JSR	\$C7DA
C7BF	8C	06 C8	STY	\$C806
C7C2	8D	07 C8	STA	\$C807
C7C5	20	06 E2	JSR	\$E206
C7C8	A9	AB	LDA	#\$AB
C7CA	20	FF AE	JSR	\$AEFF
C7CD	20	11 E2	JSR	\$E211
C7D0	20	DD C7	JSR	\$C7DD
C7D3	8C	08 C8	STY	\$C808
C7D6	8D	09 C8	STA	\$C809
C7D9	60		RTS	
C7DA	20	0E E2	JSR	\$E20E
C7DD	20	6B A9	JSR	\$A96B
C7E0	A5	15	LDA	\$15
C7E2	A4	14	LDY	\$14
C7E4	60		RTS	
C7E5	A2	07	LDX	#\$07
C7E7	BD	F1 C7	LDA	\$C7F1,X
C7EA	9D	06 C8	STA	\$C806,X

C7ED CA	DEX
C7EE 10 F7	BPL \$C7E7
C7F0 60	RTS
C7F1 00	BRK
.	
.	

The syntax to be observed here is:

SYS 50000,a,b,c-d

where:

a = first new line number.

b = gap between new line numbers.

c = first line number to be renumbered.

d = last line number to be renumbered.

If any of these are omitted the default values are used instead, which are respectively 100,10,0,63999.

Finally, if any GOTOs or GOSUBs trundle off to a non-existent line, the line number will be displayed and the line referenced will be changed to 63999.

So, rather like compiling a program, make sure that your program is in perfect working order before renumbering it. There's no such thing as an un-compiler, and you can't un-renumber a program either!

Conclusion

I hope you find some or all of these useful, and at the same time see how easy it is to produce a couple of short machine-code routines to enhance an adventure game. Little touches with the screen display can make all the difference, and if a player likes the display on one game he's more likely to come back and play another one.

If you're in the business of selling adventure games, getting a player to buy another one is what it's all about, so a little time and care over the final product can turn an average game into a company winner!

Talking about playing adventure games, it's about time we got started.

5

Getting Started

Here and in Chapter 8 we are going to construct a very simple adventure. Simple, for the benefit of those of you who've never done this kind of thing before, and in two chapters to avoid you closing the book in disgust at the sight of an extremely long listing.

Seriously, the adventure is divided into two chapters because we're going to split the game up into two parts. The first part will cover all the donkey work of drawing maps, making a vocabulary list, working out what problems to have, and so on; the second part will be all the verbs: a similar approach to that used in the 'Exploring Adventures' series. However, this time it is being used to encourage newcomers and revive old memories for earlier readers before getting on to the main listing in the book - 'Enormous Cave' (a kind of sequel to 'Colossal Cave': the first and best).

Graphics or not

One of the first questions to be settled, and one which determines to a great extent how the game must be written, is whether or not to use graphics.

As we saw in Chapter 2, there are many ways of incorporating graphics into a game. Personally I don't like graphical adventures, and so I've never written one. What's more, I don't intend to start now! However, the choice of graphics is very much up to the individual, and in the game that we'll be using as our first example there is sufficient memory left over for you to put some graphics in in one form or another for a first experiment.

Disk or tape

When it comes to producing adventures for yourself, or for a circle

of friends, the decision isn't really that important. If you've all got disk drives, then use a disk as your storage medium. On the other hand, if you're a collection of tape users then there's little point using a disk to store program and data.

When writing my own adventures, I usually use the disk drive as a development tool, and then convert everything to tape later. The time saved is enormous. But when it comes to releasing the game, other forces come into play. According to various spokesmen, this mysterious thing called 'the market' doesn't want disk-based games, it wants tape-only ones. I think there's an equal market for both, but the people who buy programs for the shops (not the ones who buy the finished product) decide that they won't take disk games and that you won't buy them.

So, from a commercial point of view, it is better to release (or submit material to software houses with a view to publication) on tape. As a result there are I'm sure some excellent disk-only games that never get to see the light of day. Bigger adventures, longer passages of text, more devious problems to solve, and so on. They exist in the United States, so why they don't get released over here I'll never know.

Still, apart from the commercial reasons there are others to be considered as well. Although the bulk of the programming techniques involved remain the same, there are some vital differences, and for a first adventure I think we ought to stick to tape-only. It's easier for us both.

Screen layouts

Remembering the simple fact that whoever is going to be playing the game will be spending a lot of time looking at the screen, an attractive screen layout is essential. The mighty Infocom use the power-up colours of the 64, which to me seems a little bit lacking in imagination, to say the least. Dark backgrounds, so I'm told by people who play a lot of adventure games, give more character and feel to an underground scenario than do light backgrounds. As long as the display doesn't end up looking too garish, it's as well to add some colour to it too, just to liven things up a little.

To use Infocom as an example again, when they display information on the screen they just scroll everything up and don't worry about clearing the screen, using different colours, or anything at all

really. Again, the choice of what to do is very much a personal one, although I like to see the screen cleared at a new location and the list of visible objects and obvious exits displayed as well. It saves having to type LOOK in whenever you arrive at a different location. Most good adventure players do that as a matter of course, but when you *have* to type LOOK in at every location it does get a mite tedious after a while.

One stumbling block when designing screen layouts is the spacing of everything. If this book was laid out with a random number of spaces between each paragraph it would look a bit odd, wouldn't it? So, no matter how much space you decide to leave between, say, room descriptions and a list of visible objects, always keep it the same.

Designing a map

This is as important when designing a game as it is when solving one. My original maps tend to look a bit of a mess, quite honestly, but at least I can refer to them while going through the program so that I know where I am at any particular time. To try to write an adventure without first producing a map is ludicrous, if not impossible.

A tidy version of the map for the game we're going to be playing is given on p.102. There are only 25 locations, to keep things nice and simple.

There are no hard and fast rules for drawing a map. I usually decide on the number of locations that the game will have, start in the middle of a sheet of paper and draw a lot of numbers to represent each location. As seems to be the case with things like this, the scale on one side of my sheet of paper is dramatically different from the scale on the other side, as I find myself running out of room, but that doesn't matter. As long as it's down on paper in one form or another, you'll be all right.

You'll notice that there's a large gap between locations 15 and 16. This is because a magic word is needed to transport you from one location to the other, rather like the Scott Adams' game 'Pirate Adventure'. You'll have to find the word out for yourself though!

Connections between all the other locations are fairly obvious: you can move reasonably freely between all the locations that are linked

$$\begin{array}{ccccccc}
 & & & & 1 \\
 & & & & + \\
 & & & & 2 \\
 & & & & + \\
 & & & & 3 \\
 & & & & + \\
 6 & + & 5 & + & 4 & + & 7 & + & 8 & + & 9 \\
 & & & & & & & & & & + \\
 & & & & & & & & 14 & & 10 \\
 & & & & & & & & + & & + \\
 & & & & & & & & 13+12+11 \\
 & & & & & & & & + \\
 & & & & & & & & 15 \\
 \\
 16+17+18 \\
 & & & & + \\
 & & & & 19 \\
 & & & & + \\
 24+21+20 \\
 & & + & + \\
 25+23+22
 \end{array}$$

by plus ('+') signs. I said 'reasonably', because there are a number of problems to surmount along the way. This, having drawn up a simple map, is what we must turn our attention to next.

Designing the problems

This is probably the most crucial part of designing an adventure game. It is usually based on the map that we've just drawn, and to a very large extent it influences the next section that we're going to look at, namely drawing up a vocabulary list of understandable words.

Make a game too hard, and people will soon get fed up with it. Make it too simple, and people will solve it in five minutes flat. Somewhere in between is a fine dividing line between very hard and very easy, and that's what you've got to aim at. Similarly, you've got to decide how many problems to have. Too many or too few is just as bad as making them too easy or too hard, so it is usually a good idea to stick to a ratio of one problem to every three or four locations. They don't need to be spaced out equally, but you must give the player the chance to progress.

For this particular game, there are just eight main problems to solve (I told you it was a simple game). Going back to the map, I usually put a circle around those locations that are going to have problems contained within them, and then alongside the map write down a brief description of what those problems are.

Locations one to three are all set inside your London flat, so the first problem is how to get out of the flat at location three. Having done that, there is a policeman at location four, so our second problem is dealing with him. Locations four to ten are all on the streets just outside your flat, and problem number three arises at the bank, where you've got to use your cashcard to draw out some money. Locations eleven to fifteen are all in the local library, and problem number four involves getting into the library in the first place.

Problem number five is set in the library, where you have to search for a specific book. This helps you solve problem number six, which is how to get from location fifteen to location sixteen and back again. Locations sixteen to twenty-five are all set on Hampstead Heath, and problem number seven arrives at location eighteen, where there's a man selling ice-creams. You've got to get an ice-cream from him.

The eighth and final problem is at location twenty-five, where you find a small child holding your wallet. Since retrieving the wallet is the purpose of the game, you've got to get it back from him. However, being an adventure child he can run faster than you and so you've got to bribe him in some way. Having got the wallet, that's the end of the game!

In this game most of the solutions to the various problems are quite obvious really, but as the purpose of the program is not to be a difficult adventure but to show you how a small game is put together, this is of no importance.

Having sorted out the map and the problems, and got a rough idea of where each location lies, we can now go on to work out the vocabulary list for the game.

Vocabulary lists

Another very important feature of any adventure game is the list (and number) of words that the program can recognise and act upon.

As 'The Lost Wallet' is a very small game, it doesn't have the largest of vocabularies. Nevertheless it can understand and react to around 65 words. So how do we generate this list of words in the first place?

Taking the verbs first, a lot of them will be common ones that occur in just about every adventure game. Words such as GET and DROP, movement commands like GO, or more simply N, S, E and W, INVENTORY (to see what you're carrying), LOOK and REDESCRIBE (to let you read a room description that might have scrolled off the screen), SEARCH and EXAMINE (to help you find out something more about an object or room) and more besides, are all to be found in the majority of adventure games.

A large number of words will come from the room descriptions themselves, so a read through those will probably bring some more verbs to light. For instance, in location nine in 'The Lost Wallet' you're told that there's a cashpoint card, so it is reasonable to assume that players will type in INSERT CARD or something like that. Thus another verb becomes INSERT. Part of the action takes place in a library, so you can bet your life that people will type in READ BOOK, and so READ is added to our list of understandable words. So it goes on, and for this game at least I ended up with a list of 36 verbs, as follows:

QUIT	FIND
HELP	WEAR
SCORE	DISCARD
INVENTORY	BUY
LOOK	READ
REDESCRIBE	TAKE
GO	PICK
JUMP	INSERT
N	CHASE
S	LOAD
E	SAVE
W	TALK
GET	GIVE
DROP	OFFER
OPEN	SEARCH
CLOSE	LISTEN
CROSS	ENTER
EXAMINE	AWAY

You'll have noticed that there are no rude words in that list: this is a family book, after all. Sadly, however, people will type in the

odd (sometimes very odd) swear word, and so the program should really be able to respond to them. There are ways of doing that without having the words resident in the listing, and we'll look at that in Chapter 8.

Having got all the verbs, we now want a list of objects. Again, these should be obvious enough from the problems, solutions to problems, and room descriptions. If one of the problems involves opening a door, you will probably need an object like a KEY to do it. If someone types in GO NORTH, then NORTH becomes another object. If you have to insert a card into the magic money machine to acquire some funds, then CARD becomes yet another object.

The complete list of objects for this adventure is as follows:

N	BANK
S	CARD
E	MONEY
W	TICKET
KEY	ICE-CREAM
NORTH	POLICEMAN
SOUTH	CHILD
EAST	KID
WEST	FLAT
RECORD	DISGUISE
DOOR	ROOM
WALLET	BOOK
ROAD	RUBBISH
LIBRARY	

Now then, all these have to be stored in the program somewhere, as indeed do all the room descriptions, so we'll look at the first part of our adventure game listing now.

```
3999 REM START OF ALL ROOM DESCRIPTIONS
4000 PRINT"YOU ARE IN THE BEDROOM OF YOUR FLAT IN
4002 PRINT"THE HEART OF LONDON.  THROUGH THE
4004 PRINT"WINDOW YOU CAN SEE THE BUSY STREET
4006 PRINT"BELOW, AND A POLICEMAN STROLLING AROUND
4007 PRINT"KEEPING AN EYE ON THINGS (AND YOU!)"
4008 PRINT:PRINT"PERHAPS YOU'RE WANTED FOR SOMETHI
NG?"
4009 PRINT:PRINT"AN EXIT LEADS SOUTH TO THE FRONT
ROOM.":GOTO3002
4010 PRINT"YOU ARE IN THE FRONT ROOM OF YOUR FLAT.
4011 PRINT"SHELVES OF BOOKS LINE THE WALLS, AND A
```

```

4012 PRINT"JUMBLE OF RECORDS HIDES AN UNTIDY PILE
4013 PRINT"OF DIRTY CLOTHES.
4014 PRINT:PRINT"AN OLD SETTEE SITS OVER A GRIMY C
ARPET,
4015 PRINT"AND YOUR STEREO KEEPS PLAYING THE SAME
4016 PRINT"RECORD OVER AND OVER AGAIN.
4017 PRINT:PRINT"DOORS LEAD OFF TO NORTH AND SOUTH
.":GOTO3002
4020 PRINT"YOU ARE IN THE FRONT HALLWAY OF YOUR
4021 PRINT"FLAT. A DOOR LEADS BACK TO THE UNTIDY
4022 PRINT"FRONT ROOM, AND BEFORE YOU LIES AN
4023 PRINT"IMPOSING WOODEN DOOR, LETTER BOX
4024 PRINT"RESPLENDENT WITH LETTER CATCHER AND
4025 PRINT"UNPAID BILLS. THROUGH THE FROSTED
4026 PRINT"GLASS IN THE TOP HALF OF THE DOOR, YOU
4027 PRINT"CAN STILL SEE A POLICEMAN WALKING
ABOUT.
4028 PRINT:PRINT"A COLLECTION OF OLD RUBBISH LIES
AT YOURFEET.":GOTO3002
4030 PRINT"YOU ARE OUTSIDE YOUR FLAT. THE ROAD IS
4031 PRINT"SO BUSY THAT YOU HAVEN'T A HOPE OF
4032 PRINT"CROSSING IT, AND SO YOU MUST GO EAST OR
4033 PRINT"WEST ALONG THE PAVEMENT, DODGING THE
OBVIOUS AS YOU WALK.":IFD0=0THEN4036
4034 PRINT:PRINT"AS YOUR DOOR SLAMS SHUT BEHIND YO
U, YOU MUST DECIDE WHAT TO DO.
4035 FORI=1TO2000:NEXT:DO=0
4036 IFWD=0THEN4038
4037 PRINT:PRINT"THE POLICEMAN DOESN'T RECOGNISE Y
OU, ANDBIDS YOU GOOD-DAY.":GOTO3002
4038 PRINT:PRINT"OH NO, THE POLICEMAN SEES YOUR FA
CE AND
4039 PRINT"SAYS 'YOU'RE NICKED, MY SON'. HE MARCHE
SYOU OFF BY THE EAR.":EO=1:GOTO3002
4040 PRINT"YOU ARE TO THE WEST OF YOUR FLAT,
4041 PRINT"WALKING ALONG AN UNTIDY PAVEMENT. IRON
4042 PRINT"RAILINGS TO THE NORTH OBSCURE THE VIEW
4043 PRINT"INTO OTHER FLATS, WHILE TO THE SOUTH
4044 PRINT"THE FLOW OF TRAFFIC IS AS HEAVY AS EVER
.":GOTO3002
4050 PRINT"YOU HAVE REACHED THE END OF THE STREET.
4052 PRINT"SINCE YOUR MUMMY TOLD YOU NEVER TO
4054 PRINT"CROSS THE ROAD BY YOURSELF, YOU CAN GO
NO FURTHER.":GOTO3002
4060 PRINT"YOU ARE TO THE EAST OF YOUR FLAT. FOR
4061 PRINT"SALE SIGNS LITTER THE GARDENS TO THE
4062 PRINT"NORTH, BUT WITH THE POLICEMAN NEARBY
4063 PRINT"THERE WOULDN'T BE MUCH POINT PINCHING
4064 PRINT"ONE.":PRINT:PRINT"IN THE DISTANCE YOU C
AN MAKE OUT THE
4065 PRINT"TOWN HIGH STREET.":GOTO3002

```

```

4070 PRINT"YOU ARE ON THE ROAD LEADING FROM YOUR
4072 PRINT"FLAT TO THE CENTRE OF TOWN. YOUR BANK
4073 PRINT"LIES TO THE EAST.":GOTO3002
4080 PRINT"YOU ARE OUTSIDE YOUR BANK. FOR ONCE
4081 PRINT"THE HOLE-IN-THE-WALL MACHINE IS WORKING
,INVITING YOU TO ACQUIRE MONEY."
4082 PRINT:PRINT"YOU KNOW YOU DARE NOT ENTER THE B
ANK,
4084 PRINT"SO YOU MUST CHOOSE BETWEEN USING THE
4085 PRINT"MACHINE OR WALKING ON.":GOTO3002
4090 PRINT"YOU ARE MID-WAY BETWEEN THE BANK AND
4091 PRINT"THE OLD LIBRARY. ABOVE THE LIBRARY IS
A
4092 PRINT"[CUJ]CLOCK, WHICH TELLS YOU THAT YOU'VE
NOW
4094 PRINT"BEEN PLAYING THIS GAME FOR JUST ABOUT
4095 A$=MID$(TI$,1,2):B$=MID$(TI$,3,2):C$=MID$(TI$
,5,2)
4096 PRINTA$" HOURS, "B$" MINUTES AND "C$" SECONDS
.
4097 PRINT:PRINT"BETTER GET ON WITH IT AND DO SOME
THING.":GOTO3002
4100 PRINT"YOU ARE OUTSIDE THE IMPOSING DOOR OF
4102 PRINT"THE OLD LIBRARY. INSIDE, THROUGH THE
4104 PRINT"CLEAN WINDOWS, YOU CAN SEE THE FROSTY
4106 PRINT"FIGURE OF THE LIBRARIAN, AND ROW UPON
4107 PRINT"ROW OF BOOKS. NOW THEN, WHERE'S YOUR
LIBRARY TICKET?":GOTO3002
4110 PRINT"YOU ARE INSIDE THE LIBRARY, TRYING TO
4111 PRINT"TIPTOE PAST THE LIBRARIAN ON CREAKING
4112 PRINT"FLOORBOARDS.
4113 PRINT:PRINT"ALL AROUND YOU ARE MASSIVE NUMBER
S OF
4114 PRINT"MUSTY OLD BOOKS.":GOTO3002
4120 PRINT"YOU ARE IN THE CENTRE OF THE LIBRARY.
4121 PRINT:PRINT"70 NORTH AND SOUTH LIE OLD STAIRW
ELLS,
4122 PRINT"LEADING TO RARELY VISITED PARTS OF THE
4123 PRINT"LIBRARY. COBWEBS HANG OFF THE CEILING,
4124 PRINT"AND THE DUST SEEMS TO COVER EVERYTHING.
":GOTO3002
4130 PRINT"YOU ARE IN THE NORTHERN PART OF THE
LIBRARY."
4132 PRINT:PRINT"SPIDERS SCURRY AND DART ABOUT AT
YOUR
4134 PRINT"UNEXPECTED INTRUSION INTO THEIR DOMAIN,
4135 PRINT"AND LITTLE RUSTLING NOISES AND THE SOUN
D
4136 PRINT"[CUJ]OF SCAMPERING FEET SEEMS TO BE IN E
VERY CORNER.":GOTO3002
4140 PRINT"YOU ARE IN THE SOUTHERN PART OF THE

```

LIBRARY.

```
4141 PRINT:PRINT"THROUGH PANELLED WINDOWS YOU CAN
SEE
4142 PRINT"AN IMPOSING VIEW TO THE SOUTH AND WEST,
4144 PRINT"ALTHOUGH HOW YOU GET TO IT IS A BIT OF
4146 PRINT"A MYSTERY. MUSTY, DUSTY BOOKS COVER
4147 PRINT"THE WALLS, AND I'M SURE THE LIBRARIAN
4148 PRINT"MUST BE PEERING OVER YOUR SHOULDER BY
NOW!":GOTO3002
4150 PRINT"YOU ARE ON THE TOP OF HAMPSTEAD HEATH."
4151 PRINT:PRINT"OVERHEAD FLUFFY WHITE CLOUDS FLOA
T
4152 PRINT"BENEATH A CLEAR BLUE SKY, WHILE TO THE
4153 PRINT"EAST A DUSTY TRACK DISAPPEARS INTO A
4154 PRINT"HOST OF TREES AND RHODODENDRONS.
4156 PRINT:PRINT"GREEN GRASS LIES LUXURIANT ON EVE
RY
4157 PRINT"SIDE, AND ALL-IN-ALL THE WORLD APPEARS
TO BE A HAPPY PLACE.":GOTO3002
4160 PRINT"YOU ARE ON A DUSTY TRACK HEADING ROUGHL
Y
4161 PRINT"[CUJEAST-WEST. PEOPLE ARE EXERCISING T
HEIR
4162 PRINT"DOGS, KITES ARE FLYING ALL AROUND, AND
4163 PRINT"IN THE DISTANCE YOU CAN HEAR THE CHIMES
4164 PRINT"OF AN ICE-CREAM VAN.":GOTO3002
4170 PRINT"YOU ARE IN A DUST BOWL IN THE MIDDLE OF
THE HEATH.":PRINT
4171 PRINT"KIDS ARE RUNNING ABOUT ALL OVER THE
4172 PRINT"PLACE, AND IN THE CENTRE OF THE BOWL IS
4173 PRINT"AN ICE-CREAM VAN.":PRINT:PRINT"A TRACK
HEADS DOWN TOWARDS THE SOUTH.":GOTO3002
4180 PRINT"YOU ARE ON A TRACK HEADING ROUGHLY
NORTH-SOUTH.":PRINT
4181 PRINT"7TREES AND BUSHES SURROUND YOU ON ALL
4182 PRINT"SIDES, MAKING IT DIFFICULT TO MOVE OFF
4183 PRINT"THE PATH.":GOTO3002
4190 PRINT"YOU ARE TO THE EAST OF A LUXURIANT GLAD
EOF BUSHES AND TREES."
4192 PRINT:PRINT"A DUSTY OLD TRACK HEADS WEST, INT
O THE
4194 PRINT"MIDDLE OF THIS GLADE, BUT IT SOON SPLIT
S
4195 PRINT"[CUJUP INTO A NUMBER OF SEPARATE PATHS.
":GOTO3002
4200 PRINT"YOU ARE IN A DARK, TANGLED GLADE OF
4202 PRINT"BRIARS AND BUSHES. PATHS LEAD OFF IN
ALL DIRECTIONS.":GOTO3002
4210 GOTO4200
4220 GOTO4200
4230 GOTO4200
```

```

4240 PRINT"YOU ARE TO THE WEST OF THE GLADE, IN AN
4242 PRINT"UNEXPECTED OPENING. THE HEATH SEEMS TO
4243 PRINT"BE QUITE CLEAR ONCE AGAIN, AND HORDES
4244 PRINT"OF SCREAMING KIDS RUN AROUND EVERYWHERE
.
4245 PRINT"WHAT IT IS TO BE YOUNG!":IFWB=1THEN3002
4246 PRINT:PRINT"ONE OF THEM APPROACHES YOU, CARRY
ING A WALLET.
4247 PRINT:PRINT"'IS THIS YOURS, MISTAH?' HE SHOUT
S.
4248 PRINT"'WELL, YOU CAN'T HAVE IT', AND HE
4249 PRINT"STANDS JUST OUT OF ARM'S REACH.":GOTO30
02
4999 REM DATA FOR ALL THE LOCATIONS, IN NORTH,SOUT
H,EAST,WEST FORMAT
5000 DATA0,2,0,0,1,3,0,0,2,4,0,0,3,0,7,5,0,0,4,6,0
,0,5,0,0,0,8,4,0,0,9,7
5002 DATA0,10,0,8,9,11,0,0,10,0,0,12,0,0,11,13,14,
15,12,0,0,13,0,0,13,0,0,0
5004 DATA0,0,17,0,0,0,18,16,0,19,0,17,18,20,0,0,19
,0,0,21,21,22,20,24,21,21,21,23
5006 DATA24,21,22,25,21,23,21,21,0,0,23,0
5008 DIMP%(25,3):FORI=1TO25:FORJ=0TO3:READP%(I,J):
NEXTJ,I
5010 DATAQUIT,HELP,SCOR,INVE,LOOK,REDE,GO,JUMP,N,S
,E,W,GET,DROP,OPEN,CLOS
5012 DATACROS,EXAM,FIND,WEAR,DISC,BUY,READ,TAKE,PI
CK,INSE,CHAS,LOAD,SAVE
5014 DATATALK,GIVE,OFFE,SEAR,LIST,ENTE,AWAY
5016 NV=36:DIMVB$(NV):FORI=1TONV:READVB$(I):NEXT
5020 DATAN,S,E,W,KEY,NORT,SOUT,EAST,WEST,RECO,DOOR
,WALL,ROAD,LIBR,BANK,CARD,MONE
5021 DATATICK,ICE-,POLI,CHIL,KID,FLAT,DISG,ROOM,BO
OK,RUBB
5030 NN=27:DIMNO$(NN):FORI=1TONN:READNO$(I):NEXT:D
IMOB$(NN),OB$(NN)
5040 DATA"NORTH","SOUTH","EAST","WEST"
5042 FORI=0TO3:READDD$(I):NEXT
5050 DATAN,S,E,W,A SMALL KEY,N,S,E,W,A RECORD,A SO
LID DOOR,YOUR WALLET!,A ROAD
5052 DATATHE LIBRARY,THE BANK,YOUR CASHCARD,SOME M
ONEY,YOUR LIBRARY TICKET
5054 DATAAN ICE-CREAM,A POLICEMAN,A CHILD,A KID,YO
UR FLAT,A HAIRY DISGUISE
5056 DATAYOUR ROOM,AN OLD BOOK,SOME RUBBISH
5060 FORI=1TONN:READOB$(I):NEXT
5070 DATA0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,24,0,6,0,4,
25,0,0,0,0,0,0
5072 FORI=1TONN:READOB$(I):NEXT
5099 RETURN
10000 POKE53281,7:POKE53280,9:PRINTCHR$(14):PRINC

```

```

HR$(8):PRINT"[CLR,BLK]THE LOST WALLET!
10002 FORI=1TO16:PRINT"[CBMY]";:FORJ=1TO50:NEXTJ,I
10004 PRINT:PRINT"ONE DAY, WHILST STROLLING AROUND
IN
10006 PRINT"LONDON DOING SOME SHOPPING, YOU MUST
10008 PRINT"HAVE LOST YOUR WALLET.
10010 PRINT:PRINT"WHEN YOU GOT HOME, IT WAS NO LONGER
10012 PRINT"THERE. OF COURSE, YOU CAN'T REMEMBER
10014 PRINT"WHERE YOU DROPPED IT, AND SO THE GOAL
10016 PRINT"IN THIS MINI-ADVENTURE IS TO FIND YOUR
WALLET AGAIN."
10018 PRINT:PRINT"PLEASE HANG ON WHILE I READ SOME
DATA IN."
10020 RETURN
20000 FORI=49152TO49162:READA:POKEI,A:NEXT:RETURN
20001 DATA 104,104,32,253,174,32,235,183,76,166,16
8
60000 CM$="":
60002 PRINT"[RVS,BLK]*[OFF,CL]";
60005 GETZ$:IFZ$=""THEN60005
60010 Z=ASC(Z$):IFZ>95THEN60005
60080 ZL=LEN(CM$):IFZL>27THEN60110
60100 IFZ>31THENCM$=CM$+Z$:PRINTZ$;:GOTO60002
60110 IFZ=13ANDZLTHENPRINT" ":RETURN
60120 IFZ=20ANDZLTHENCM$=LEFT$(CM$,ZL-1):PRINTZ$;
60140 GOTO60002

```

READY.

The room descriptions run from lines 3999-4249, and should be fairly obvious. Each new location starts ten lines on from the last one, so room one starts at line 4000, room two at 4010, room three at 4020, and so on. Lines 4090-4097 have you standing between the library and a bank, and tell you how long you've been playing the game. A little real-time feature there.

Lines 4999 to 5008 contain the data for all the locations, and tell you which way you can move from the various locations. The variable P% in line 5008 controls this information, so if, for example, P%(1,0) was equal to zero, then you couldn't go north from location one. If P%(1,1) was equal to two, then you could go south to location two from location one. P%(1,2) and P%(1,3) control whether or not you can go east or west respectively from location one. If P%(16,2) was equal to seventeen, then you could go east to location seventeen from location sixteen, and thus all our movements around the game are controlled.

Lines 5010 to 5030 contain the list of verbs and nouns in four letter format. We'll see the reason for this in the next chapter, when we discuss various parsers. Lines 5040 to 5042 contain the words NORTH, SOUTH, EAST and WEST, and are used when the room descriptions are printed up. Lines 5050 to 5060 contain lengthier descriptions for the nouns. It's no use informing the player that he can see a POLI when we want him to see a POLICEMAN: he might think the game involved parrots.

The last little bit of data, in lines 5070 to 5072, tells us where each object starts in the game. A zero tells us that the object doesn't yet exist, or hasn't been found by the player, and a number (e.g. 24) tells us that that particular object starts in location twenty-four.

Conclusion

All that data will be used again in Chapter 8, so keep it safe somewhere for now: don't save over it whatever you do.

Well, we've started anyway, but there's a lot more needed yet before we can complete our adventure. The next two chapters concern themselves with some of the mechanics of the game, but just before that a few words about working as part of a team.

The great games, like 'Zork', 'The Hobbit', and others, are all created by a team of authors, not just one lone programmer. You might like to try your hand at that. You might know someone who is gifted with words, you might be a good programmer, and if you're going to use graphics you might have an artistic friend. Why not pool your talents? Three heads are better than one.

I've worked with my brother on a couple of adventures now, one we finished very quickly, the other we regrettably didn't (although we made it in the end)! The finished result was much better than anything either of us could have done by ourselves, so why not give it a try? You never know what you might come up with.

6

Parsers

First of all, what precisely is a 'parser'? In every adventure game there must be some part of the program that allows the player to type in something. Maybe a simple movement, such as GO NORTH, or just N, a command to acquire something, such as GET OBJECT, or any one of the million and one things that people are likely to think of as they set about solving an adventure.

The first adventure games limited you to a VERB...NOUN type of input, so that you could only type in (at most) two words. Personally I find that this is usually sufficient, but some people obviously thought otherwise, and we began to see programs like 'Zork', or 'The Hobbit', which allow you to type in extremely lengthy instructions such as:

TAKE THE BLUE HAT AND PUT IT ON THE GREEN SHELF

Or:

GO NORTH AND THEN GO WEST AND OPEN THE CUPBOARD

Now this is all very well, unless you happen to make a spelling mistake in the first couple of words and type TKE THE BLUE HAT or GO NOTRH or something like that, in which case you'll have to type the whole thing in again and might just as well have gone for the old VERB...NOUN input in the first place.

Putting personal preferences aside for a while, though, it is clear that many adventures are being produced that use these sophisticated techniques of input. Level 9 games have used them for a while, although it is interesting to note that some of their later efforts ('Red Moon' is current at the time of writing) have gone back to the VERB...NOUN school of thought, to the extent of only analysing the first three letters of each word. This was the method I used in the 'Exploring Adventures' series: how galling it is to see that

reviewers heartily dislike that particular method of input when used in one person's adventures, and yet those same reviewers solidly approve of Level 9 when they do it. One man's meat is the same man's poison, I suppose.

Basic or machine code?

As we keep on seeing, there are always two sides to the coin. If you write something in Basic it is very easy to alter it, or to extend it, but it doesn't necessarily run very fast. On the other hand, write a routine in machine code and it will run like the wind, until you come to alter it. You've got to decide at the outset which you're going to use, and as in the rest of this book we'll start with a Basic version first.

Basic parser

Part of this will be familiar to those who have read the 'Exploring Adventures' series, but part will not. The second part of the parser listing is an old friend that can easily be adapted.

```
60000 CM$="":
60002 PRINT"[RVB]*[OFF,CL]";
60005 GETZ$: IFZ$="" THEN 60005
60010 Z=ASC(Z$): IFZ>95 THEN 60005
60080 ZL=LEN(CM$): IFZL>27 THEN 60110
60100 IFZ>31 THEN CM$=CM$+Z$: PRINTZ$: GOTO 60002
60110 IFZ=13 AND ZL THEN PRINT " ": RETURN
60120 IFZ=20 AND ZL THEN CM$=LEFT$(CM$,ZL-1): PRINTZ$:
60140 GOTO 60002
```

READY.

Essentially this prints an asterisk on the screen and waits for a key to be pressed. Then, by checking the ASCII value of the key pressed, we can either add to the input string, delete a character, complete the string or ignore the character. Whatever we do, the end input (whether it is GO NORTH or KNOCK AT HOUSE NUMBER THREE) is stored in the string CM\$.

This is where the first part of the listing comes into play.

```

390 PRINT:PRINT"WHAT NOW ? ";IFA1=1THENPRINTA1$;
CM$=A1$;PRINT:LC=LEN(CM$);PRINT:GOTO392
391 GOSUB60000:PRINT:LC=LEN(CM$)
392 IFOB%(19)=-1THENMI=MI+1;IFMI=10THENB50
393 NO$="";VB$="";VB=0:NO=0;IFA1=1THENA1=0:GOTO396
394 A1=0:FORI=1TOLC:IFMID$(CM$,I,5)=" AND "THENA1$
=RIGHT$(CM$,LC-I-4):A1=1:A2=I
395 NEXT:IFA1=1THENCM$=LEFT$(CM$,A2-1)
396 FORI=1TOLC:IFMID$(CM$,I,1)<>" "THENVB$=VB$+MID
$(CM$,I,1):NEXT
397 V1$=VB$:VB$=LEFT$(V1$,4):FORI=1TONV:IFVB$(I)=V
B$THENVB=I:GOTO402
398 NEXT
400 VB=7:NO$=VB$:GOTO406
402 IFLEN(V1$)+1>=LEN(CM$) THENNO=0:RETURN
403 FORI=LCTO1STEP-1:IFMID$(CM$,I,1)<>" "THENNO$=N
O$+MID$(CM$,I,1):NEXT
404 NO$=RIGHT$(NO$,4):FORI=1TO4:A$(I)=MID$(NO$,5-I
,1):NEXT
405 NO$="":FORI=1TO4:NO$=NO$+A$(I):NEXT
406 FORI=1TONN:IFNO$=NO$(I) THEN409
407 NEXTI
408 NO=0:RETURN
409 NO=I:RETURN

```

Look at lines 390 to 392. After printing up the 'What now' prompt, a check is made on the variable A1. The first time we run through this routine it will obviously equal zero, and so we'll come back to this part later. Line 391 goes off to the routine that we've just looked at, and then declares the variable LC to be equal to the length of the input string.

Line 392 belongs to a real adventure game, and in fact comes from 'The Lost Wallet', reproduced in Chapter 8, so again we'll look at that later on in the relevant chapter.

Line 393 declares a few variables, and has another check on the variable A1: more later.

The next two lines allow us to type in and respond to such complex inputs as:

GO NORTH AND OPEN DOOR

GET KEY AND UNLOCK CUPBOARD

TAKE GLASS AND GET DRINK

and so on.

What they can't respond to is inputs like:

GO NORTH SLOWLY

and others of that ilk, which 'The Hobbit' allowed you to use. Actually, I don't think the game ever required you to type anything like that in, which is just as well really because then probably no one would ever have solved it.

Line 394 scans through the input string taking groups of five letters at a time. If it finds the sequence 'space-A-N-D-space' then it sets the variable A1 to be equal to one. This means that there is a repeat command to be acted on, so that the next time this routine is called up the appropriate responses can be made in lines 390 and 393. A1\$ then becomes the right hand side of the input string, and in line 395 if A1 has been set then we have to have another temporary input string, which is taken from the left hand side of the original one. So the temporary string becomes all the characters leading up to the first space in the sequence 'space-A-N-D-space', and the secondary string (A1\$) contains all the characters after the second space in that same sequence.

Whether we're acting on a repeat input string or a temporary one it's still CM\$ that has to be analysed, and so we must proceed to line 396.

This begins a check on CM\$ by looking at every letter in the string and seeing if it's a space or not. If it isn't then we start to build up VB\$ as the VERB part of our input. When a space is found program flow goes to line 397 where we chop VB\$ down to size by taking the first four letters of it.

Then a search begins through a previously set-up table of verbs to see if what has been typed in matches any of the ones in that table. If it does, then we get a value for VB and proceed to line 402, but if it doesn't then set VB to seven (which in the example this was taken from was the verb GO) and declare the NOUN part of our input to be equal to the VERB part and go off to line 406.

Assuming the verb has been recognised, line 402 checks for the length of the input to see whether the player has typed in more than one word. If he has just typed in one word then return from this subroutine, otherwise onward we go.

Line 403 checks through the input string backwards one character at a time, and if that character isn't a string then we build up the NOUN part of the string in the same way that we earlier built up the VERB part of it. When (or indeed if) a space is eventually found the program falls through to line 404.

As an example at this point, if the player had typed in OPEN THE RED DOOR, VB\$ would contain "OPEN" and NO\$ would contain "ROOD", since we checked through the input string backwards to find the noun. So now the noun part of the string has to be turned round, and this is the function of lines 404 and 405.

Finally a check is made to identify the noun using a previously set-up table just as we did for the verb. If the noun is recognised we have a noun number NO, if it isn't then NO is just set to zero, but either way that is the end of this Basic input parser.

Conclusion

This is a reasonably fast routine that allows the program to cope with fairly complex instructions. It is also easy to amend for any other adventure game, but its great virtue is that it enables you to use a virtually limitless vocabulary.

Finally, you can also very easily insert a check somewhere in the program so that if a player types in something rude then a suitable response can be given. Such as:

```
100 IF VB$="rude word" THEN PRINT "WATCH IT!":GOTO  
somewhere
```

However, in common with any routine that is written in Basic, it is not as fast as we would sometimes like such a routine to be, and so we have to proceed (as usual) to machine code.

Machine-code parser

The main part of the routine, as used here, resides in memory from locations \$C000 to \$C383, and is not easily relocatable. You can move it somewhere else if you want, but it's going to take a lot of time and trouble, so I suggest that you leave it where it is.

B*

PC SR AC XR YR SP
.,8FEB 33 00 D3 00 F6

.
C000 4C 41 C2 JMP \$C241
C003 A2 00 LDX ##00
C005 A9 20 LDA ##20
C007 9D 8D CD STA \$CD8D,X
C00A 9D A2 CD STA \$CDA2,X
C00D 9D B7 CD STA \$CDB7,X
C010 EB INX
C011 E0 15 CPX ##15
C013 D0 F2 BNE \$C007
C015 A9 00 LDA ##00
C017 8D CD CD STA \$CDCD
C01A 8D CE CD STA \$CDCE
C01D 8D CF CD STA \$CDCF
C020 8D D1 CD STA \$CDD1
C023 8D D2 CD STA \$CDD2
C026 8D D3 CD STA \$CDD3
C029 8D D5 CD STA \$CDD5
C02C A9 41 LDA ##41
C02E 8D D7 CD STA \$CDD7
C031 A9 41 LDA ##41
C033 8D DA CD STA \$CDDA
C036 A9 41 LDA ##41
C038 8D DB CD STA \$CDD8
C03B A9 41 LDA ##41
C03D 8D D9 CD STA \$CDD9
C040 A9 40 LDA ##40
C042 8D DB CD STA \$CDD8
C045 A0 00 LDY ##00
C047 B9 3C CD LDA \$CD3C,Y
C04A C9 20 CMP ##20
C04C F0 0A BEQ \$C058
C04E 4C 20 CC JMP \$CC20
C051 EE CD CD INC \$CDDC
C054 C8 INY
C055 4C 47 C0 JMP \$C047
C058 AC CD CD LDY \$CDDC
C05B CC CC CD CPY \$CDCC
C05E D0 03 BNE \$C063
C060 4C EF C0 JMP \$C0EF
C063 C8 INY
C064 A2 00 LDX ##00
C066 B9 3C CD LDA \$CD3C,Y
C069 4C 30 CC JMP \$CC30
C06C EB INX
C06D C8 INY
C06E CC CC CD CPY \$CDCC

C071	D0	F3		BNE	\$C066
C073	AD	CC	CD	LDA	\$CDCC
C076	38			SEC	
C077	ED	CD	CD	SBC	\$CDCCD
C07A	A8			TAY	
C07B	88			DEY	
C07C	A9	20		LDA	##20
C07E	4C	40	CC	JMP	\$CC40
C081	C8			INX	
C082	CC	CC	CD	CPY	\$CDCC
C085	D0	F5		BNE	\$C07C
C087	AD	CC	CD	LDA	\$CDCC
C08A	38			SEC	
C08B	ED	CD	CD	SBC	\$CDCCD
C08E	8D	CC	CD	STA	\$CDCC
C091	CE	CC	CD	DEC	\$CDCC
C094	AC	CC	CD	LDY	\$CDCC
C097	88			DEY	
C098	B9	3C	CD	LDA	\$CD3C,Y
C09B	C9	20		CMP	##20
C09D	F0	12		BEQ	\$C0B1
C09F	88			DEY	
C0A0	D0	09		BNE	\$C0AB
C0A2	AD	CC	CD	LDA	\$CDCC
C0A5	8D	CE	CD	STA	\$CDCE
C0AB	4C	B1	C0	JMP	\$C0B1
C0AB	EE	CE	CD	INC	\$CDCE
C0AE	4C	98	C0	JMP	\$C098
C0B1	AD	CC	CD	LDA	\$CDCC
C0B4	38			SEC	
C0B5	ED	CE	CD	SBC	\$CDCE
C0B8	A8			TAY	
C0B9	A2	00		LDX	##00
C0BB	B9	3C	CD	LDA	\$CD3C,Y
C0BE	4C	50	CC	JMP	\$CC50
C0C1	A9	20		LDA	##20
C0C3	4C	10	CC	JMP	\$CC10
C0C6	E8			INX	
C0C7	C8			INX	
C0C8	CC	CC	CD	CPY	\$CDCC
C0CB	D0	EE		BNE	\$C0BB
C0CD	AD	CC	CD	LDA	\$CDCC
C0D0	38			SEC	
C0D1	ED	CE	CD	SBC	\$CDCE
C0D4	F0	19		BEQ	\$C0EF
C0D6	8D	CC	CD	STA	\$CDCC
C0D9	CE	CC	CD	DEC	\$CDCC
C0DC	A0	00		LDY	##00
C0DE	B9	3C	CD	LDA	\$CD3C,Y
C0E1	C9	20		CMP	##20
C0E3	F0	0A		BEQ	\$C0EF

COE5	99	B7	CD	STA	\$CDB7,Y
COE8	4C	00	CC	JMP	\$CC00
COEB	EA			NOP	
COEC	4C	DE	CO	JMP	\$CODE
COEF	20	F5	C2	JSR	\$C2F5
COF2	A9	00		LDA	#\$00
COF4	85	BB		STA	\$BB
COF6	A9	C5		LDA	#\$C5
COF8	85	BC		STA	\$BC
COFA	A9	8D		LDA	#\$8D
COFC	85	FB		STA	\$FB
COFE	A9	CD		LDA	#\$CD
C100	85	FC		STA	\$FC
C102	AD	D7	CD	LDA	\$CDD7
C105	8D	D4	CD	STA	\$CDD4
C108	20	A7	C1	JSR	\$C1A7
C10B	AD	D0	CD	LDA	\$CDD0
C10E	C9	FF		CMP	\$\$FF
C110	D0	06		BNE	\$C118
C112	20	DE	C1	JSR	\$C1DE
C115	4C	99	C1	JMP	\$C199
C118	8D	D1	CD	STA	\$CDD1
C11B	AD	CE	CD	LDA	\$CDCE
C11E	F0	79		BEQ	\$C199
C120	A9	00		LDA	#\$00
C122	85	BB		STA	\$BB
C124	A9	C6		LDA	#\$C6
C126	85	BC		STA	\$BC
C128	A9	A2		LDA	##A2
C12A	85	FB		STA	\$FB
C12C	A9	CD		LDA	#\$CD
C12E	85	FC		STA	\$FC
C130	AD	D8	CD	LDA	\$CDD8
C133	8D	D4	CD	STA	\$CDD4
C136	20	A7	C1	JSR	\$C1A7
C139	AD	D0	CD	LDA	\$CDD0
C13C	C9	FF		CMP	\$\$FF
C13E	D0	25		BNE	\$C165
C140	A9	00		LDA	#\$00
C142	85	BB		STA	\$BB
C144	A9	C7		LDA	#\$C7
C146	85	BC		STA	\$BC
C148	AD	D9	CD	LDA	\$CDD9
C14B	8D	D4	CD	STA	\$CDD4
C14E	20	A7	C1	JSR	\$C1A7
C151	AD	D0	CD	LDA	\$CDD0
C154	C9	FF		CMP	\$\$FF
C156	F0	07		BEQ	\$C15F
C158	18			CLC	
C159	6D	DB	CD	ADC	\$CDD8
C15C	4C	65	C1	JMP	\$C165

C15F	20	DE	C1	JSR	\$C1DE
C162	4C	99	C1	JMP	\$C199
C165	8D	D2	CD	STA	\$CDD2
C168	AD	CF	CD	LDA	\$CDCF
C16B	F0	2C		BEQ	\$C199
C16D	A9	00		LDA	#\$00
C16F	85	BB		STA	\$BB
C171	A9	C8		LDA	#\$C8
C173	85	BC		STA	\$BC
C175	A9	B7		LDA	#\$B7
C177	85	FB		STA	\$FB
C179	A9	CD		LDA	#\$CD
C17B	85	FC		STA	\$FC
C17D	AD	DA	CD	LDA	\$CDDA
C180	8D	D4	CD	STA	\$CDD4
C183	20	A7	C1	JSR	\$C1A7
C186	AD	D0	CD	LDA	\$CDD0
C189	C9	FF		CMP	\$\$FF
C18B	D0	04		BNE	\$C191
C18D	20	DE	C1	JSR	\$C1DE
C190	60			RTS	
C191	8D	D3	CD	STA	\$CDD3
C194	A9	00		LDA	#\$00
C196	8D	B6	02	STA	\$02B6
C199	60			RTS	
C19A	A2	00		LDX	#\$00
C19C	A9	20		LDA	#\$20
C19E	9D	3C	CD	STA	\$CD3C,X
C1A1	E8			INX	
C1A2	E0	51		CPX	\$\$51
C1A4	D0	FB		BNE	\$C19E
C1A6	60			RTS	
C1A7	A9	01		LDA	#\$01
C1A9	8D	D0	CD	STA	\$CDD0
C1AC	A0	00		LDY	#\$00
C1AE	B1	BB		LDA	(\$BB),Y
C1B0	D1	FB		CMP	(\$FB),Y
C1B2	D0	06		BNE	\$C1BA
C1B4	C8			INX	
C1B5	C0	04		CPY	#\$04
C1B7	D0	F5		BNE	\$C1AE
C1B9	60			RTS	
C1BA	AD	D0	CD	LDA	\$CDD0
C1BD	18			CLC	
C1BE	69	01		ADC	#\$01
C1C0	CD	D4	CD	CMP	\$CDD4
C1C3	F0	13		BEQ	\$C1DB
C1C5	8D	D0	CD	STA	\$CDD0
C1C8	18			CLC	
C1C9	A5	BB		LDA	\$BB
C1CB	69	04		ADC	#\$04

C1CD	85	BB		STA	\$BB
C1CF	A5	BC		LDA	\$BC
C1D1	69	00		ADC	##00
C1D3	85	BC		STA	\$BC
C1D5	4C	AC	C1	JMP	*C1AC
C1D8	A9	FF		LDA	##FF
C1DA	8D	D0	CD	STA	\$CDD0
C1DD	60			RTS	
C1DE	A9	00		LDA	##00
C1E0	8D	B6	02	STA	\$02B6
C1E3	A9	0D		LDA	##0D
C1E5	20	D2	FF	JSR	\$FFD2
C1E8	20	D2	FF	JSR	\$FFD2
C1EB	A0	00		LDY	##00
C1ED	B9	12	C2	LDA	\$C212,Y
C1F0	20	D2	FF	JSR	\$FFD2
C1F3	C8			INY	
C1F4	C0	23		CPY	##23
C1F6	D0	F5		BNE	*C1ED
C1F8	A9	00		LDA	##00
C1FA	8D	B6	02	STA	\$02B6
C1FD	A9	20		LDA	##20
C1FF	20	D2	FF	JSR	\$FFD2
C202	20	A9	C3	JSR	\$C3A9
C205	EA			NOP	
C206	EA			NOP	
C207	A9	FF		LDA	##FF
C209	8D	D5	CD	STA	\$CDD5
C20C	A9	00		LDA	##00
C20E	8D	B6	02	STA	\$02B6
C211	60			RTS	
C212	C9	20		CMP	##20
C214	44			???	
C215	4F			???	
C216	4E	27	54	LSR	\$5427
C219	20	55	4E	JSR	\$4E55
C21C	44			???	
C21D	45	52		EOR	\$52
C21F	53			???	
C220	54			???	
C221	41	4E		EOR	(\$4E,X)
C223	44			???	
C224	20	59	4F	JSR	\$4F59
C227	55	2C		EOR	\$2C,X
C229	20	C9	27	JSR	\$27C9
C22C	4D	20	41	EOR	\$4120
C22F	46	52		LSR	\$52
C231	41	49		EOR	(\$49,X)
C233	44			???	
C234	2E	EA	EA	ROL	\$EAEA
C237	EA			NOP	

C238	EA		NOP
C239	EA		NOP
C23A	EA		NOP
C23B	EA		NOP
C23C	EA		NOP
C23D	EA		NOP
C23E	EA		NOP
C23F	EA		NOP
C240	EA		NOP
C241	A6	D6	LDX \$D6
C243	A0	00	LDY #\$00
C245	18		CLC
C246	20	F0 FF	JSR \$FFFF0
C249	A9	05	LDA #\$05
C24B	20	D2 FF	JSR \$FFD2
C24E	A9	8D	LDA #\$8D
C250	20	D2 FF	JSR \$FFD2
C253	A9	11	LDA #\$11
C255	20	D2 FF	JSR \$FFD2
C258	EA		NOP
C259	EA		NOP
C25A	A6	D6	LDX \$D6
C25C	A0	00	LDY #\$00
C25E	8C	D0 CD	STY \$CDD0
C261	18		CLC
C262	20	F0 FF	JSR \$FFFF0
C265	20	00 CB	JSR \$CB00
C268	20	D2 FF	JSR \$FFD2
C26B	20	10 CB	JSR \$CB10
C26E	EA		NOP
C26F	20	9A C1	JSR \$C19A
C272	20	E4 FF	JSR \$FFE4
C275	C9	00	CMP #\$00
C277	F0	F9	BEQ \$C272
C279	C9	20	CMP #\$20
C27B	D0	08	BNE \$C285
C27D	AE	D0 CD	LDX \$CDD0
C280	F0	F0	BEQ \$C272
C282	4C	C3 C2	JMP \$C2C3
C285	C9	0D	CMP #\$0D
C287	D0	08	BNE \$C291
C289	AD	D0 CD	LDA \$CDD0
C28C	F0	E4	BEQ \$C272
C28E	4C	E6 C2	JMP \$C2E6
C291	C9	5F	CMP #\$5F
C293	F0	AC	BEQ \$C241
C295	C9	2A	CMP #\$2A
C297	F0	59	BEQ \$C2F2
C299	C9	14	CMP #\$14
C29B	D0	1E	BNE \$C2BB
C29D	AC	D0 CD	LDY \$CDD0

C2A0	F0	D0		BEQ	\$C272
C2A2	CE	D0	CD	DEC	\$CDD0
C2A5	20	D2	FF	JSR	\$FFD2
C2A8	20	D2	FF	JSR	\$FFD2
C2AB	A9	2A		LDA	#\$2A
C2AD	20	D2	FF	JSR	\$FFD2
C2B0	AC	D0	CD	LDY	\$CDD0
C2B3	A9	20		LDA	#\$20
C2B5	99	3C	CD	STA	\$CD3C,Y
C2B8	4C	72	C2	JMP	\$C272
C2BB	C9	41		CMP	#\$41
C2BD	30	B3		BMI	\$C272
C2BF	C9	5B		CMP	#\$5B
C2C1	10	AF		BPL	\$C272
C2C3	AC	D0	CD	LDY	\$CDD0
C2C6	C0	24		CPY	#\$24
C2C8	F0	A8		BEQ	\$C272
C2CA	99	3C	CD	STA	\$CD3C,Y
C2CD	EE	D0	CD	INC	\$CDD0
C2D0	8D	D6	CD	STA	\$CDD6
C2D3	A9	9D		LDA	#\$9D
C2D5	20	D2	FF	JSR	\$FFD2
C2D8	AD	D6	CD	LDA	\$CDD6
C2DB	20	D2	FF	JSR	\$FFD2
C2DE	A9	2A		LDA	#\$2A
C2E0	20	D2	FF	JSR	\$FFD2
C2E3	4C	72	C2	JMP	\$C272
C2E6	AD	D0	CD	LDA	\$CDD0
C2E9	8D	CC	CD	STA	\$CDCC
C2EC	20	02	C3	JSR	\$C302
C2EF	4C	03	C0	JMP	\$C003
C2F2	4C	EF	C0	JMP	\$C0EF
C2F5	A2	00		LDX	#\$00
C2F7	60			RTS	
C2F8	20	9D	C0	JSR	\$C09D
C2FB	07			???	
C2FC	E8			INX	
C2FD	E0	27		CPX	#\$27
C2FF	D0	F8		BNE	\$C2F9
C301	60			RTS	
C302	A2	00		LDX	#\$00
C304	A9	20		LDA	#\$20
C306	9D	9B	07	STA	\$079B,X
C309	E8			INX	
C30A	E0	27		CPX	#\$27
C30C	D0	F8		BNE	\$C306
C30E	A9	00		LDA	#\$00
C310	8D	86	02	STA	\$0286
C313	A9	12		LDA	#\$12
C315	20	D2	FF	JSR	\$FFD2
C318	A6	D6		LDX	\$D6

C31A	A0	00	LDY	#\$00
C31C	18		CLC	
C31D	20	F0 FF	JSR	\$FFFF0
C320	A0	00	LDY	#\$00
C322	B9	3C CD	LDA	\$CD3C,Y
C325	20	D2 FF	JSR	\$FFD2
C328	C8		INY	
C329	CC	D0 CD	CPY	\$CDD0
C32C	D0	F4	BNE	\$C322
C32E	A9	2E	LDA	#\$2E
C330	20	D2 FF	JSR	\$FFD2
C333	20	A9 C3	JSR	\$C3A9
C336	EA		NOP	
C337	EA		NOP	
C338	A9	00	LDA	#\$00
C33A	8D	86 02	STA	\$0286
C33D	60		RTS	
C33E	EA		NOP	
C33F	A0	00	LDY	#\$00
C341	A9	05	LDA	#\$05
C343	99	00 DB	STA	\$DB00,Y
C346	A9	A0	LDA	\$A0
C348	99	00 04	STA	\$0400,Y
C34B	C8		INY	
C34C	C0	F0	CPY	\$F0
C34E	D0	F1	BNE	\$C341
C350	A0	00	LDY	#\$00
C352	A9	05	LDA	#\$05
C354	99	F0 DB	STA	\$DBF0,Y
C357	A9	A0	LDA	\$A0
C359	99	F0 04	STA	\$04F0,Y
C35C	C8		INY	
C35D	C0	F0	CPY	\$F0
C35F	D0	F1	BNE	\$C352
C361	A0	00	LDY	#\$00
C363	A9	05	LDA	#\$05
C365	99	E0 D9	STA	\$D9E0,Y
C368	A9	A0	LDA	\$A0
C36A	99	E0 05	STA	\$05E0,Y
C36D	C8		INY	
C36E	C0	F0	CPY	\$F0
C370	D0	F1	BNE	\$C363
C372	A0	00	LDY	#\$00
C374	A9	77	LDA	\$77
C376	99	D0 06	STA	\$06D0,Y
C379	A9	00	LDA	#\$00
C37B	99	D0 DA	STA	\$DAD0,Y
C37E	C8		INY	
C37F	C0	2B	CPY	\$2B
C381	D0	F1	BNE	\$C374
C383	60		RTS	

C384	FF		???
C385	FF		???
C386	FF		???
C387	FF		???
C388	FF		???
C389	FF		???
C38A	FF		???
C38B	FF		???
C38C	FF		???
C38D	FF		???
C38E	FF		???
C38F	FF		???
C390	FF		???
C391	FF		???
C392	FF		???
C393	FF		???
C394	FF		???
C395	FF		???
C396	FF		???
C397	FF		???
C398	FF		???
C399	FF		???
C39A	FF		???
C39B	FF		???
C39C	FF		???
C39D	FF		???
C39E	FF		???
C39F	FF		???
C3A0	60		RTS
C3A1	D6	8D	DEC #8D,X
C3A3	AB		TAY
C3A4	C3		???
C3A5	4C	41 C2	JMP #C241
C3A8	07		???
C3A9	60		RTS
C3AA	AB		TAY
C3AB	C3		???
C3AC	85	D6	STA #D6
C3AE	60		RTS
C3AF	07		???
C3B0	C2		???
C3B1	00		BRK
.			
.			

Then comes our word table, which allows up to 64 verbs, 128 nouns, and 64 'link words'. We'll come to those 'link words' later.

```

.:C500 41 54 54 41 42 45 41 43
.:C508 42 4C 49 4E 42 52 49 42
.:C510 43 4C 4F 53 51 55 49 54
.:C518 43 55 54 20 44 49 47 20
.:C520 44 52 49 4E 44 52 4F 50
.:C528 45 20 20 20 45 41 54 20
.:C530 45 4E 54 45 45 58 50 4C
.:C538 45 58 41 4D 46 45 45 44
.:C540 53 54 41 54 46 49 4C 4C
.:C548 47 45 54 20 47 49 56 45
.:C550 47 4F 20 20 47 52 41 42
.:C558 48 45 41 54 48 49 44 45
.:C560 48 4F 4C 44 49 4E 56 45
.:C568 4A 4F 49 4E 4A 55 4D 50
.:C570 4B 49 4C 4C 4C 41 53 48
.:C578 4C 4F 4F 4B 4D 4F 56 45
.:C580 4E 20 20 20 4F 46 46 45
.:C588 4F 50 45 4E 4F 52 44 45
.:C590 50 41 55 53 50 49 43 4B
.:C598 50 4C 41 43 50 4C 45 41
.:C5A0 50 52 41 59 52 41 49 53
.:C5A8 52 45 43 52 52 45 44 45
.:C5B0 48 45 4C 50 52 4F 57 20
.:C5B8 53 41 43 4B 53 45 41 52
.:C5C0 53 45 4E 44 53 45 54 20
.:C5C8 53 48 41 52 53 20 20 20
.:C5D0 53 50 41 52 53 4C 45 45
.:C5D8 53 43 4F 52 53 54 4F 52
.:C5E0 54 41 4B 45 54 41 4C 4B
.:C5E8 54 49 45 20 55 4E 54 49
.:C5F0 57 41 49 54 57 20 20 20
.:C5F8 53 41 56 45 4C 4F 41 44
.:C600 41 4E 43 48 41 4E 54 53
.:C608 42 41 47 20 42 41 52 4C
.:C610 42 41 53 4B 42 45 45 53
.:C618 42 4F 41 54 42 4F 57 20
.:C620 43 41 54 54 43 41 56 45
.:C628 43 48 41 4E 43 49 52 43
.:C630 43 49 52 43 43 49 54 59
.:C638 43 59 43 4C 44 4F 4F 52
.:C640 44 4F 57 4E 44 52 55 47
.:C648 44 55 4E 47 45 41 53 54
.:C650 45 55 52 59 46 41 4D 49
.:C658 46 45 4E 43 46 4F 4F 44
.:C660 47 49 41 4E 47 4F 41 54
.:C668 53 4B 49 4E 48 41 52 42
.:C670 48 49 4C 4C 48 49 56 45
.:C678 48 4F 4C 44 48 4F 4E 45
.:C680 49 53 4C 41 48 45 45 4C
.:C688 4B 4E 49 46 4D 41 53 54
.:C690 4D 45 4E 20 4D 49 4C 4B

```



```

.:C698 4E 4F 52 54 4F 41 52 53
.:C6A0 4F 44 59 53 50 49 47 53
.:C6A8 50 4C 41 4E 50 4F 4C 59
.:C6B0 50 52 49 45 52 4F 43 48
.:C6B8 52 4F 44 20 52 4F 50 45
.:C6C0 52 55 49 4E 53 41 49 4C
.:C6C8 53 43 59 4C 53 48 45 45
.:C6D0 53 49 52 45 53 4C 41 56
.:C6D8 53 4F 55 54 53 50 52 49
.:C6E0 53 54 41 46 53 54 45 57
.:C6E8 53 57 4F 52 54 41 56 45
.:C6F0 54 45 54 48 54 49 52 45
.:C6F8 54 52 41 43 54 52 45 45
.:C700 54 52 45 4E 55 4E 54 45
.:C708 55 50 20 20 57 41 4C 4C
.:C710 57 41 58 20 57 45 53 54
.:C718 57 48 45 59 57 49 4E 45
.:C720 57 4F 4F 44 4F 46 46 20
.:C728 59 4F 55 20 50 52 4F 47
.:C730 53 4C 45 45 45 41 54 45
.:C738 53 45 41 20 53 48 4F 52
.:C740 41 53 48 4F 53 48 49 50
.:C748 57 41 54 45 4D 41 4E 20
.:C750 42 45 45 48 20 20 20 20
.:C758 20 20 20 20 20 20 20 20
.:C760 20 20 20 20 20 20 20 20
.:C768 20 20 20 20 20 20 20 20
.:C770 20 20 20 20 20 20 20 20
.:C778 20 20 20 20 20 20 20 20
.:C780 20 20 20 20 20 20 20 20
.:C788 20 20 20 20 20 20 20 20
.:C790 20 20 20 20 20 20 20 20
.:C798 20 20 20 20 20 20 20 20
.:C7A0 20 20 20 20 20 20 20 20
.:C7A8 20 20 20 20 20 20 20 20
.:C7B0 20 20 20 20 20 20 20 20
.:C7B8 20 20 20 20 20 20 20 20
.:C7C0 20 20 20 20 20 20 20 20
.:C7C8 20 20 20 20 20 20 20 20
.:C7D0 20 20 20 20 20 20 20 20
.:C7D8 20 20 20 20 20 20 20 20
.:C7E0 20 20 20 20 20 20 20 20
.:C7E8 20 20 20 20 20 20 20 20
.:C7F0 20 20 20 20 20 20 20 20
.:C7F8 20 20 20 20 20 20 20 20
.:C800 41 52 4F 55 41 52 52 4F
.:C808 4F 56 45 52 42 55 43 4B
.:C810 46 52 45 53 47 4F 4C 44
.:C818 4C 45 41 54 4C 45 4E 47
.:C820 4C 4F 54 55 42 4F 57 4C
.:C828 50 4F 54 45 52 4F 55 47

```

```

.:C830 53 45 56 45 54 48 45 20
.:C838 54 48 52 4F 54 4F 20 20
.:C840 54 4F 57 41 55 50 20 20
.:C848 44 4F 57 4E 57 49 43 4B
.:C850 57 49 54 48 48 45 52 45
.:C858 57 4F 4F 44 46 4F 52 20
.:C860 41 4D 4F 4E 42 45 54 57
.:C868 41 43 52 4F 42 45 53 49
.:C870 55 4E 44 45 43 41 54 54
.:C878 43 49 52 43 43 49 52 43
.:C880 43 59 43 4C 45 55 52 59
.:C888 46 41 4D 49 47 49 41 4E
.:C890 47 4F 41 54 4D 45 4E 20
.:C898 4F 44 59 53 50 4F 4C 59
.:C8A0 50 52 49 45 53 43 59 4C
.:C8A8 53 48 45 45 53 49 52 45
.:C8B0 53 4C 41 56 54 49 52 45
.:C8B8 45 41 54 45 49 4E 20 20
.:C8C0 41 54 20 20 4F 4E 20 20
.:C8C8 42 41 47 20 20 20 20 20
.:C8D0 20 20 20 20 20 20 20 20
.:C8D8 20 20 20 20 20 20 20 20
.:C8E0 20 20 20 20 20 20 20 20
.:C8E8 20 20 20 20 20 20 20 20
.:C8F0 20 20 20 20 20 20 20 20
.:C8FB 20 20 20 20 20 20 20 20

```

This, as you can see, runs from locations \$C500 to \$C8FF, with the verbs occupying the first 256 locations, the nouns the next 512, and finally another 256 bytes for the link words.

You don't need to type this in for your own programs, but if you want to see the vocabulary list for one of my games, well, be my guest!

The last few parts of this listing are a hotch-potch of little routines that are used occasionally by the main one given earlier.

```

C400 A9 00      LDA #$00
C402 BD 14 03   STA $0314
C405 A9 C9      LDA #$C9
C407 BD 15 03   STA $0315
C40A 60         RTS
.
.
.
.
CC00 EE CF CD   INC $CDCF
CC03 C8         INY
CC04 C0 49      CPY #$49

```

CC06	F0	03		BEQ	\$CC0B
CC08	4C	EC	C0	JMP	\$C0EC
CC0B	4C	EF	C0	JMP	\$C0EF
CC0E	FF			???	
CC0F	FF			???	
CC10	99	3C	CD	STA	\$CD3C,Y
CC13	E0	C3		CPX	#\$C3
CC15	F0	03		BEQ	\$CC1A
CC17	4C	C6	C0	JMP	\$C0C6
CC1A	4C	CD	C0	JMP	\$C0CD
CC1D	FF			???	
CC1E	FF			???	
CC1F	FF			???	
CC20	99	8D	CD	STA	\$CD8D,Y
CC23	C0	7D		CPY	#\$7D
CC25	F0	03		BEQ	\$CC2A
CC27	4C	51	C0	JMP	\$C051
CC2A	4C	5B	C0	JMP	\$C05B
CC2D	00			BRK	
CC2E	00			BRK	
CC2F	00			BRK	
CC30	9D	3C	CD	STA	\$CD3C,X
CC33	E0	C3		CPX	#\$C3
CC35	F0	03		BEQ	\$CC3A
CC37	4C	6C	C0	JMP	\$C06C
CC3A	4C	73	C0	JMP	\$C073
CC3D	00			BRK	
CC3E	00			BRK	
CC3F	00			BRK	
CC40	99	3C	CD	STA	\$CD3C,Y
CC43	C0	C3		CPY	#\$C3
CC45	F0	03		BEQ	\$CC4A
CC47	4C	81	C0	JMP	\$C081
CC4A	4C	87	C0	JMP	\$C087
CC4D	FF			???	
CC4E	FF			???	
CC4F	FF			???	
CC50	9D	A2	CD	STA	\$CDA2,X
CC53	E0	5D		CPX	#\$5D
CC55	F0	03		BEQ	\$CC5A
CC57	4C	C1	C0	JMP	\$C0C1
CC5A	4C	CD	C0	JMP	\$C0CD
CC5D	FF			???	
.					
CB00	A9	01		LDA	#\$01
CB02	85	C7		STA	\$C7
CB04	A9	00		LDA	#\$00
CB06	8D	86	02	STA	\$0286
CB09	A9	2A		LDA	#\$2A
CB0B	60			RTS	

CB0C	FF		???
CB0D	FF		???
CB0E	FF		???
CB0F	FF		???
CB10	A9	00	LDA #\$00
CB12	85	C7	STA #C7
CB14	60		RTS
CB15	FF		???
CB16	FF		???
CB17	FF		???
CB18	FF		???
CB19	FF		???
CB1A	FF		???
CB1B	FF		???
CB1C	FF		???
CB1D	FF		???
CB1E	FF		???
CB1F	FF		???
CB20	A9	93	LDA #\$93
CB22	20	D2 FF	JSR \$FFD2
CB25	A9	00	LDA #\$00
CB27	85	FB	STA \$FB
CB29	A9	A0	LDA #\$A0
CB2B	85	FC	STA \$FC
CB2D	A0	00	LDY #\$00
CB2F	B1	FB	LDA (\$FB),Y
CB31	91	FB	STA (\$FB),Y
CB33	C8		INY
CB34	D0	F9	BNE \$CB2F
CB36	E6	FC	INC \$FC
CB38	A5	FC	LDA \$FC
CB3A	C9	C0	CMP #\$C0
CB3C	D0	EF	BNE \$CB2D
CB3E	A9	36	LDA #\$36
CB40	85	01	STA \$01
CB42	A9	00	LDA #\$00
CB44	85	FB	STA \$FB
CB46	A9	E0	LDA #\$E0
CB48	85	FC	STA \$FC
CB4A	A0	00	LDY #\$00
CB4C	B1	FB	LDA (\$FB),Y
CB4E	91	FB	STA (\$FB),Y
CB50	C8		INY
CB51	D0	F9	BNE \$CB4C
CB53	E6	FC	INC \$FC
CB55	A5	FC	LDA \$FC
CB57	C9	00	CMP #\$00
CB59	D0	EF	BNE \$CB4A
CB5B	A9	35	LDA #\$35

```

CB5D 85 01      STA $01
CB5F A9 12      LDA ##12
CB61 8D 82 E8   STA $E882
CB64 8D D0 E6   STA $E6D0
CB67 8D 8E E9   STA $E98E
CB6A A9 11      LDA ##11
CB6C 8D 04 E9   STA $E904
CB6F 8D 5F E5   STA $E55F
CB72 8D 26 E9   STA $E926
CB75 8D 70 E9   STA $E970
CB78 A9 15      LDA ##15
CB7A 8D 57 E5   STA $E557
CB7D 60         RTS
CB7E 00         BRK
CB7F 00         BRK
CB80 D0 FF      BNE $CB81
CB82 FF        ???
.
.
```

These little programs are mainly used to store variables in various parts of memory, but don't feel tempted to leave them out because if you do the program will come to a grinding halt.

If you want to use any other routines in the block \$C000 to \$CFFF (such as the text compressor in the next chapter), then 'safe' locations are \$C900 to \$CAFF, \$CB7E to \$CBFF and \$CE00 to \$CFFF.

Finally, if you want to type it in in Basic form, here it is:

```

10 SE = 49152 : EE = 53247
20 :
30 FOR I = SE TO EE : READ A : POKE I , A : NEXT
40 :
50000 DATA 076,065,194,162,000,169,032,157,141,205
,157,162,205,157
50001 DATA 183,205,232,224,021,208,242,169,000,141
,205,205,141,206
50002 DATA 205,141,207,205,141,209,205,141,210,205
,141,211,205,141
50003 DATA 213,205,169,065,141,215,205,169,065,141
,218,205,169,065
50004 DATA 141,216,205,169,065,141,217,205,169,064
,141,219,205,160
50005 DATA 000,185,060,205,201,032,240,010,076,032
,204,238,205,205
50006 DATA 200,076,071,192,172,205,205,204,204,205
,208,003,076,239
```

50007 DATA 192,200,162,000,185,060,205,076,048,204
 ,232,200,204,204
 50008 DATA 205,208,243,173,204,205,056,237,205,205
 ,168,136,169,032
 50009 DATA 076,064,204,200,204,204,205,208,245,173
 ,204,205,056,237
 50010 DATA 205,205,141,204,205,206,204,205,172,204
 ,205,136,185,060
 50011 DATA 205,201,032,240,018,136,208,009,173,204
 ,205,141,206,205
 50012 DATA 076,177,192,238,206,205,076,152,192,173
 ,204,205,056,237
 50013 DATA 206,205,168,162,000,185,060,205,076,080
 ,204,169,032,076
 50014 DATA 016,204,232,200,204,204,205,208,238,173
 ,204,205,056,237
 50015 DATA 206,205,240,025,141,204,205,206,204,205
 ,160,000,185,060
 50016 DATA 205,201,032,240,010,153,183,205,076,000
 ,204,234,076,222
 50017 DATA 192,032,245,194,169,000,133,187,169,197
 ,133,188,169,141
 50018 DATA 133,251,169,205,133,252,173,215,205,141
 ,212,205,032,167
 50019 DATA 193,173,208,205,201,255,208,006,032,222
 ,193,076,153,193
 50020 DATA 141,209,205,173,206,205,240,121,169,000
 ,133,187,169,198
 50021 DATA 133,188,169,162,133,251,169,205,133,252
 ,173,216,205,141
 50022 DATA 212,205,032,167,193,173,208,205,201,255
 ,208,037,169,000
 50023 DATA 133,187,169,199,133,188,173,217,205,141
 ,212,205,032,167
 50024 DATA 193,173,208,205,201,255,240,007,024,109
 ,219,205,076,101
 50025 DATA 193,032,222,193,076,153,193,141,210,205
 ,173,207,205,240
 50026 DATA 044,169,000,133,187,169,200,133,188,169
 ,183,133,251,169
 50027 DATA 205,133,252,173,218,205,141,212,205,032
 ,167,193,173,208
 50028 DATA 205,201,255,208,004,032,222,193,096,141
 ,211,205,169,000
 50029 DATA 141,134,002,096,162,000,169,032,157,060
 ,205,232,224,081
 50030 DATA 208,248,096,169,001,141,208,205,160,000
 ,177,187,209,251
 50031 DATA 208,006,200,192,004,208,245,096,173,208
 ,205,024,105,001

50032 DATA 205,212,205,240,019,141,208,205,024,165
 ,187,105,004,133
 50033 DATA 187,165,188,105,000,133,188,076,172,193
 ,169,255,141,208
 50034 DATA 205,096,169,000,141,134,002,169,013,032
 ,210,255,032,210
 50035 DATA 255,160,000,185,018,194,032,210,255,200
 ,192,035,208,245
 50036 DATA 169,000,141,134,002,169,032,032,210,255
 ,032,169,195,234
 50037 DATA 234,169,255,141,213,205,169,000,141,134
 ,002,096,201,032
 50038 DATA 068,079,078,039,084,032,085,078,068,069
 ,082,083,084,065
 50039 DATA 078,068,032,089,079,085,044,032,201,039
 ,077,032,065,070
 50040 DATA 082,065,073,068,046,234,234,234,234,234
 ,234,234,234,234
 50041 DATA 234,234,234,166,214,160,000,024,032,240
 ,255,169,005,032
 50042 DATA 210,255,169,141,032,210,255,169,017,032
 ,210,255,234,234
 50043 DATA 166,214,160,000,140,208,205,024,032,240
 ,255,032,000,203
 50044 DATA 032,210,255,032,016,203,234,032,154,193
 ,032,228,255,201
 50045 DATA 000,240,249,201,032,208,008,174,208,205
 ,240,240,076,195
 50046 DATA 194,201,013,208,008,173,208,205,240,228
 ,076,230,194,201
 50047 DATA 095,240,172,201,042,240,089,201,020,208
 ,030,172,208,205
 50048 DATA 240,208,206,208,205,032,210,255,032,210
 ,255,169,042,032
 50049 DATA 210,255,172,208,205,169,032,153,060,205
 ,076,114,194,201
 50050 DATA 065,048,179,201,091,016,175,172,208,205
 ,192,036,240,168
 50051 DATA 153,060,205,238,208,205,141,214,205,169
 ,157,032,210,255
 50052 DATA 173,214,205,032,210,255,169,042,032,210
 ,255,076,114,194
 50053 DATA 173,208,205,141,204,205,032,002,195,076
 ,003,192,076,239
 50054 DATA 192,162,000,096,032,157,192,007,232,224
 ,039,208,248,096
 50055 DATA 162,000,169,032,157,152,007,232,224,039
 ,208,248,169,000
 50056 DATA 141,134,002,169,018,032,210,255,166,214
 ,160,000,024,032
 50057 DATA 240,255,160,000,185,060,205,032,210,255

,200,204,208,205
50058 DATA 208,244,169,046,032,210,255,032,169,195
,234,234,169,000
50059 DATA 141,134,002,096,234,160,000,169,005,153
,000,216,169,160
50060 DATA 153,000,004,200,192,240,208,241,160,000
,169,005,153,240
50061 DATA 216,169,160,153,240,004,200,192,240,208
,241,160,000,169
50062 DATA 005,153,224,217,169,160,153,224,005,200
,192,240,208,241
50063 DATA 160,000,169,119,153,208,006,169,000,153
,208,218,200,192
50064 DATA 040,208,241,096,255,255,255,255,255,255
,255,255,255,255
50065 DATA 255,255,255,255,255,255,255,255,255,255
,255,255,255,255
50066 DATA 255,255,255,255,096,214,141,168,195,076
,065,194,007,096
50067 DATA 168,195,133,214,096,007,194,000,000,000
,000,000,000,000
50068 DATA 000,000,000,000,000,000,000,000,000,255,255
,255,255,255,255
50069 DATA 255,255,255,255,255,255,255,255,255,255
,255,255,255,255
50070 DATA 255,255,255,255,255,255,255,255,255,255
,255,255,128,000
50071 DATA 000,000,000,000,000,000,000,000,000,000,000
,000,000,000,000
50072 DATA 000,000,000,000,000,000,000,000,000,000,000
,000,000,000,000
50073 DATA 000,000,169,000,141,020,003,169,201,141
,021,003,096,255
50074 DATA 255,255,255,255,255,255,255,255,255,255
,255,255,255,255
50075 DATA 255,255,255,255,255,255,128,000,000,000
,000,000,000,000
50076 DATA 000,000,000,000,000,000,000,000,000,000,000
,000,000,000,000
50077 DATA 000,000,000,000,000,000,000,000,000,000,000
,255,255,255,255
50078 DATA 255,255,255,255,255,255,255,255,255,255
,255,255,255,255
50079 DATA 255,255,255,255,255,255,255,255,255,255
,255,255,255,255
50080 DATA 128,000,000,000,000,000,000,000,000,000,000
,000,000,000,000
50081 DATA 000,000,000,000,000,000,000,000,000,000,000
,000,000,000,000
50082 DATA 000,000,000,000,000,193,255,255,255,255,255
,255,255,255,255

50083 DATA 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 50084 DATA 255,255,255,255,255,255,255,255,128,000,000,000,000,000,000,000
 50085 DATA 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
 50086 DATA 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
 50087 DATA 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 50088 DATA 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 50089 DATA 255,255,128,000,000,000,000,000,000,000,000,000,000,000,000,000
 50090 DATA 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
 50091 DATA 000,000,000,000,000,000,032,065,084,084,065,066,069,065,067
 50092 DATA 066,076,073,078,066,082,073,066,067,076,079,083,081,085
 50093 DATA 073,084,067,085,084,032,068,073,071,032,068,082,073,078
 50094 DATA 068,082,079,080,069,032,032,032,069,065,084,032,069,078
 50095 DATA 084,069,069,088,080,076,069,088,065,077,070,069,069,068
 50096 DATA 083,084,065,084,070,073,076,076,071,069,084,032,071,073
 50097 DATA 086,069,071,079,032,032,071,082,065,066,072,069,065,084
 50098 DATA 072,073,068,069,072,079,076,068,073,078,086,069,074,079
 50099 DATA 073,078,074,085,077,080,075,073,076,076,076,065,083,072
 50100 DATA 076,079,079,075,077,079,086,069,078,032,032,032,079,070
 50101 DATA 070,069,079,080,069,078,079,082,068,069,080,065,085,083
 50102 DATA 080,073,067,075,080,076,065,067,080,076,069,065,080,082
 50103 DATA 065,089,082,065,073,083,082,069,067,082,082,069,068,069
 50104 DATA 072,069,076,080,082,079,087,032,083,065,067,075,083,069
 50105 DATA 065,082,083,069,078,068,083,069,084,032,083,072,065,082
 50106 DATA 083,032,032,032,083,080,065,082,083,076,069,069,083,067
 50107 DATA 079,082,083,084,079,082,084,065,075,069,084,065,076,075
 50108 DATA 084,073,069,032,085,078,084,073,087,065

,073,084,087,032
50109 DATA 032,032,083,065,086,069,076,079,065,068
,065,078,067,072
50110 DATA 065,078,084,083,066,065,071,032,066,065
,082,076,066,065
50111 DATA 083,075,066,069,069,083,066,079,065,084
,066,079,087,032
50112 DATA 067,065,084,084,067,065,086,069,067,072
,065,078,067,073
50113 DATA 082,067,067,073,082,067,067,073,084,089
,067,089,067,076
50114 DATA 068,079,079,082,068,079,087,078,068,082
,085,071,068,085
50115 DATA 078,071,069,065,083,084,069,085,082,089
,070,065,077,073
50116 DATA 070,069,078,067,070,079,079,068,071,073
,065,078,071,079
50117 DATA 065,084,083,075,073,078,072,065,082,066
,072,073,076,076
50118 DATA 072,073,086,069,072,079,076,068,072,079
,078,069,073,083
50119 DATA 076,065,075,069,069,076,075,078,073,070
,077,065,083,084
50120 DATA 077,069,078,032,077,073,076,075,078,079
,082,084,079,065
50121 DATA 082,083,079,068,089,083,080,073,071,083
,080,076,065,078
50122 DATA 080,079,076,089,080,082,073,069,082,079
,067,075,082,079
50123 DATA 068,032,082,079,080,069,082,085,073,078
,083,065,073,076
50124 DATA 083,067,089,076,083,072,069,069,083,073
,082,069,083,076
50125 DATA 065,086,083,079,085,084,083,080,082,073
,083,084,065,070
50126 DATA 083,084,069,087,083,087,079,082,084,065
,086,069,084,069
50127 DATA 084,072,084,073,082,069,084,082,065,067
,084,082,069,069
50128 DATA 084,082,069,078,085,078,084,069,085,080
,032,032,087,065
50129 DATA 076,076,087,065,088,032,087,069,083,084
,087,072,069,089
50130 DATA 087,073,078,069,087,079,079,068,079,070
,070,032,089,079
50131 DATA 085,032,080,082,079,071,083,076,069,069
,069,065,084,069
50132 DATA 083,069,065,032,083,072,079,082,065,083
,072,079,083,072
50133 DATA 073,080,087,065,084,069,077,065,078,032
,066,069,069,072

50134 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50135 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50136 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50137 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50138 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50139 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50140 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50141 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50142 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50143 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50144 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50145 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50146 DATA 032,032,032,032,065,082,079,085,065,082
 ,082,079,079,086
 50147 DATA 069,082,066,085,067,075,070,082,069,083
 ,071,079,076,068
 50148 DATA 076,069,065,084,076,069,078,071,076,079
 ,084,085,066,079
 50149 DATA 087,076,080,079,084,069,082,079,085,071
 ,083,069,086,069
 50150 DATA 084,072,069,032,084,072,082,079,084,079
 ,032,032,084,079
 50151 DATA 087,065,085,080,032,032,068,079,087,078
 ,087,073,067,075
 50152 DATA 087,073,084,072,072,069,082,069,087,079
 ,079,068,070,079
 50153 DATA 082,032,065,077,079,078,066,069,084,087
 ,065,067,082,079
 50154 DATA 066,069,083,073,085,078,068,069,067,065
 ,084,084,067,073
 50155 DATA 082,067,067,073,082,067,067,089,067,076
 ,069,085,082,089
 50156 DATA 070,065,077,073,071,073,065,078,071,079
 ,065,084,077,069
 50157 DATA 078,032,079,068,089,083,080,079,076,089
 ,080,082,073,069
 50158 DATA 083,067,089,076,083,072,069,069,083,073
 ,082,069,083,076
 50159 DATA 065,086,084,073,082,069,069,065,084,069

,073,078,032,032
50160 DATA 065,084,032,032,079,078,032,032,066,065
,071,032,032,032
50161 DATA 032,032,032,032,032,032,032,032,032,032
,032,032,032,032
50162 DATA 032,032,032,032,032,032,032,032,032,032
,032,032,032,032
50163 DATA 032,032,032,032,032,032,032,032,032,032
,032,032,032,032
50164 DATA 032,032,032,032,032,032,032,032,179,173
,000,201,201,133
50165 DATA 048,014,201,193,016,010,233,132,170,189
,026,201,076,088
50166 DATA 202,234,032,210,255,096,003,003,004,003
,003,004,004,003
50167 DATA 000,000,000,000,000,000,000,000,000,005
,004,003,007,004
50168 DATA 006,003,000,005,007,005,003,003,003,003
,003,003,003,004
50169 DATA 003,005,004,003,003,005,004,004,004,004
,004,004,007,003
50170 DATA 003,005,004,004,006,004,004,003,003,003
,000,003,006,010
50171 DATA 013,016,020,024,027,027,027,027,027,027
,027,027,027,027
50172 DATA 032,036,039,046,050,056,059,059,064,071
,076,079,082,085
50173 DATA 088,091,094,097,101,104,109,113,116,119
,124,128,132,136
50174 DATA 140,144,148,155,158,161,166,170,174,180
,184,188,191,194
50175 DATA 197,069,065,068,069,082,083,066,076,079
,087,084,069,082
50176 DATA 065,082,068,070,082,079,077,073,071,072
,084,079,085,084
50177 DATA 082,079,085,078,068,068,079,087,078,079
,078,032,195,072
50178 DATA 065,082,089,066,068,084,072,065,084,211
,067,089,076,076
50179 DATA 065,084,079,032,087,072,073,067,072,067
,072,065,078,078
50180 DATA 069,076,087,072,069,082,069,078,079,087
,212,072,069,084
50181 DATA 072,069,079,070,032,217,079,085,065,076
,076,065,082,069
50182 DATA 083,065,073,076,073,083,032,206,079,082
,084,072,212,082
50183 DATA 079,089,065,078,068,065,086,069,211,079
,085,084,072,087
50184 DATA 073,084,072,083,072,073,080,197,065,083
,084,079,085,071

50185 DATA 072,215,069,083,084,072,073,076,076,072
 ,065,082,066,079
 50186 DATA 085,082,073,078,071,089,079,085,087,072
 ,073,076,069,076
 50187 DATA 065,078,068,069,065,067,072,073,083,076
 ,065,078,068,087
 50188 DATA 073,078,068,083,073,068,069,065,078,084
 ,083,069,065,073
 50189 DATA 078,032,224,024,240,015,188,086,201,170
 ,185,147,201,032
 50190 DATA 210,255,200,202,208,246,096,169,157,032
 ,210,255,096,000
 50191 DATA 000,000,000,000,000,000,000,000,000,000
 ,000,000,000,000
 50192 DATA 201,255,255,255,255,255,255,255,255,255
 ,255,255,255,255
 50193 DATA 255,255,255,255,255,255,255,255,255,255
 ,255,255,255,255
 50194 DATA 255,255,255,255,128,000,000,000,000,000
 ,000,000,000,000
 50195 DATA 000,000,000,000,000,000,000,000,000,000
 ,000,000,000,000
 50196 DATA 000,000,000,000,000,000,000,000,255,255
 ,255,255,255,255
 50197 DATA 255,255,255,255,255,255,255,255,255,255
 ,255,255,255,255
 50198 DATA 255,255,255,255,255,255,255,255,255,255
 ,255,255,128,000
 50199 DATA 000,000,000,000,000,000,000,000,000,000
 ,000,000,000,000
 50200 DATA 000,000,000,000,000,000,000,000,000,000
 ,000,000,000,000
 50201 DATA 000,000,169,001,133,199,169,000,141,134
 ,002,169,042,096
 50202 DATA 255,255,255,255,169,000,133,199,096,255
 ,255,255,255,255
 50203 DATA 255,255,255,255,255,255,169,147,032,210
 ,255,169,000,133
 50204 DATA 251,169,160,133,252,160,000,177,251,145
 ,251,200,208,249
 50205 DATA 230,252,165,252,201,192,208,239,169,054
 ,133,001,169,000
 50206 DATA 133,251,169,224,133,252,160,000,177,251
 ,145,251,200,208
 50207 DATA 249,230,252,165,252,201,000,208,239,169
 ,053,133,001,169
 50208 DATA 018,141,130,232,141,208,230,141,142,233
 ,169,017,141,004
 50209 DATA 233,141,095,229,141,038,233,141,112,233
 ,169,021,141,087
 50210 DATA 229,096,000,000,208,255,255,255,255,255

,255,255,255,255
50211 DATA 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
50212 DATA 255,255,255,255,255,255,255,255,255,255,128,000,000,000,000,000
50213 DATA 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
50214 DATA 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
50215 DATA 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
50216 DATA 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
50217 DATA 255,255,128,000,000,000,000,000,000,000,000,000,000,000,000,000
50218 DATA 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
50219 DATA 000,000,000,000,000,000,000,238,207,205,200,192,073,240,003
50220 DATA 076,236,192,076,239,192,255,255,153,060,205,224,195,240
50221 DATA 003,076,198,192,076,205,192,255,255,255,153,141,205,192
50222 DATA 125,240,003,076,081,192,076,088,192,000,000,000,157,060
50223 DATA 205,224,195,240,003,076,108,192,076,115,192,000,000,000
50224 DATA 153,060,205,192,195,240,003,076,129,192,076,135,192,255
50225 DATA 255,255,157,162,205,224,093,240,003,076,193,192,076,205
50226 DATA 192,255,255,255,192,195,016,006,153,060,012,076,184,194
50227 DATA 136,076,184,194,000,000,192,195,016,006,153,060,205,076
50228 DATA 205,194,076,187,194,000,000,000,128,255,255,255,255,255
50229 DATA 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
50230 DATA 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
50231 DATA 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
50232 DATA 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
50233 DATA 000,000,255,255,255,255,255,255,255,255,255,255,255,255,255
50234 DATA 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
50235 DATA 255,255,255,255,255,255,255,255,128,000,000,000,000,000,000,000

50236 DATA 000,000,000,000,000,000,000,000,000,000
 ,000,000,000,000
 50237 DATA 000,000,000,000,000,000,000,000,000,000
 ,169,001,141,025
 50238 DATA 208,162,194,160,000,173,018,208,016,004
 ,162,001,160,002
 50239 DATA 142,018,208,140,033,208,032,048,205,173
 ,013,220,041,001
 50240 DATA 240,003,076,049,234,076,188,254,200,105
 ,000,020,255,002
 50241 DATA 234,234,165,214,201,017,208,008,032,064
 ,205,169,145,032
 50242 DATA 072,069,076,080,032,032,032,032,032,032
 ,032,032,032,032
 50243 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50244 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50245 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50246 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50247 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,072,069,076
 50248 DATA 080,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50249 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50250 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50251 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50252 DATA 032,032,032,032,004,004,000,000,045,045
 ,000,000,065,000
 50253 DATA 080,065,065,065,065,064,032,032,032,032
 ,032,032,032,032
 50254 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,084,072,073
 50255 DATA 083,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50256 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,032
 50257 DATA 032,032,032,032,032,032,032,032,032,032
 ,032,032,032,255
 50258 DATA 250,250,251,253,040,006,140,188,255,072
 ,136,175,238,104
 50259 DATA 000,020,250,002,248,172,129,036,000,144
 ,045,152,106,016
 50260 DATA 130,007,000,032,248,014,127,090,255,255
 ,255,219,255,255
 50261 DATA 255,255,127,255,255,255,255,247,255,124

,255,127,251,255
50262 DATA 255,255,251,188,251,122,255,255,255,255
,250,255,000,230
50263 DATA 141,128,032,047,136,160,040,008,144,062
,205,000,127,014
50264 DATA 128,016,000,024,155,004,013,008,030,002
,024,000,000,000
50265 DATA 000,062,066,223,219,251,184,255,242,059
,218,178,255,251
50266 DATA 255,255,191,251,255,247,247,223,255,191
,255,159,255,251
50267 DATA 255,255,250,242,250,053,032,004,012,040
,211,008,008,131
50268 DATA 162,040,000,020,186,010,240,138,129,000
,008,016,045,008
50269 DATA 042,016,002,007,000,000,144,006,063,082
,223,255,255,219
50270 DATA 255,191,255,255,243,255,255,255,255,243
,255,053,255,127
50271 DATA 243,253,255,255,251,184,255,122,255,255
,254,255,250,119
50272 DATA 000,098,135,000,000,043,128,032,008,008
,144,022,076,000
50273 DATA 059,014,000,016,000,024,026,004,001,008
,030,002,008,000
50274 DATA 000,000,000,062,173,240,207,201,001,240
,003,076,049,234
50275 DATA 032,040,201,076,049,234,005,032,114,195
,234,234,234,234
50276 DATA 234,234,234,234,234,234,234,234,234,234
,234,234,234,234
50277 DATA 234,234,234,234,234,234,234,234,234,234
,173,240,207,201
50278 DATA 001,240,003,076,049,234,032,040,201,076
,049,234,005,036
50279 DATA 000,144,045,136,106,016,134,007,000,000
,216,014,127,090
50280 DATA 223,255,255,219,255,255,255,255,127,255
,255,255,255,247
50281 DATA 255,125,255,102,013,000,032,047,008,032
,012,008,144,062
50282 DATA 077,000,127,015,128,016,000,024,155,004
,005,008,030,002
50283 DATA 008,000,000,000,000,062,082,223,223,251
,188,255,242,187
50284 DATA 218,178,255,251,255,255,191,255,255,247
,247,223,255,191
50285 DATA 255,143,255,249,255,255,250,242,250,245
,032,004,140,040
50286 DATA 215,072,136,135,162,040,000,020,250,010
,240,136,129,004


```

50287 DATA 008,144,045,008,042,000,002,007,000,000
,144,006,063,082
50288 DATA 223,255,255,219,255,191,255,255,243,255
,255,255,255,243
50289 DATA 255,117,255,127,243,253,255,255,251,184
,255,114,255,255
50290 DATA 254,255,250,119,000,226,133,128,032,175
,128,032,008,008
50291 DATA 144,022,076,000,063,014,000,016,000,024
,026,004,005,008
50292 DATA 030,128,008,000,000,000,000,000,000

```

READY.

Daunting, isn't it?

How it works

The basis for this routine lies in a program developed by Allen Webb, although it's been modified quite a bit. To use it you'll need a short Basic subroutine of the form:

```

10 PRINT:PRINT"WHAT NOW ?":PRINT:GOSUB1000
1000 SYS 49152:IF PEEK(52693)=255 THEN 1000
1002 VE=PEEK(52689):NO=PEEK(52690):LI=PEEK(52691)
1004 RETURN

```

Here VE and NO are analogous to the verb and noun numbers contained in the earlier Basic parser, and LI belongs to the mysterious link words.

The routine works by taking the first, second and last words that have been typed in, shortening (if the word was, say, UNLOCK) or lengthening (if the word was, say, GO) them to contain four characters, and then comparing those words against a table held in memory. Pressing the back-arrow key will delete a command, and pressing the asterisk key will repeat a command if you want to, for example, go north five times without typing it in five times.

If the routine comes across something that it doesn't recognise, then it prints up the message:

"I don't understand you, I'm afraid."

and repeats the prompt symbol (a reverse field asterisk).

If you want to change that message, the ASCII codes for the letters and characters printed lie in memory from \$C212 to \$C235: 35 of them in all. If you lengthen or shorten the message, be careful to change that part of the program which checks to see that all 35 characters have been printed out. This is at location \$C1F4, the CPY # \$23. If your message only had 20 characters, then change that to a CPY # \$14 (hexadecimal mode now, remember?).

If you're a competent machine-code programmer, you'll be able to work out how the program achieves its aims, but if you're not it will probably take the rest of this book to explain everything, so instead we'll concentrate on how you'd go about using the routine in your own programs.

Amending the routine

As you might imagine, what you'd have to alter to use this routine yourself is the word table from location \$C500 to location \$CBFF.

First of all, you'll need to draw up a list of verbs, nouns and link words. Link words? Well, normally we'd just check for a VERB...NOUN input of the form:

GO NORTH
OPEN DOOR

or whatever. However, by judicious use of these link words we can check for more complex things. For example:

OPEN RED DOOR
GIVE KEY TO SMITH
TALK TO ROBOT
UNLOCK BLUE FILING CABINET

and so on. In other words, if we have an alternative list of words that the routine can understand, as well as all the verbs and nouns, we can build up quite complicated sentences and thus enhance our adventure games.

Let's set about starting a small game. Our verbs will be GET, DROP, OPEN, CLOSE, the nouns will be KEY, DOOR, CABINET, CUPBOARD, and the link words will be RED, BLUE, THE.

Looking at locations \$C500 to \$C50F, we'll need to alter these to

cope with the verbs GET, DROP, OPEN, CLOSE. All you have to do is type in the ASCII values of the first four letters of each word (or the number of letters and a suitable number of spaces) into memory. The result would be:

```
.:C500 47 45 54 20 44 52 4F 50  
.:C508 4F 50 45 4E 43 4C 4F 53
```

So if you ran the little Basic routine given earlier and typed in GET, location 52689 would contain a 1, because it's the first verb on the list.

Now to type the nouns in. They start at \$C600, so we'll have to alter the first 16 bytes (four per noun) to read:

```
.:C600 4B 45 59 20 44 4F 4F 52  
.:C608 43 41 42 49 43 55 50 42
```

Now if you run that Basic routine and type in OPEN DOOR, location 52689 would contain a 3 (for OPEN) and location 52690 would contain a 2 (for DOOR).

Finally, the link words. These start in memory at location \$C800, and we'd need to alter the first 12 bytes of this lot to read:

```
.:C800 52 45 44 20 42 4C 55 45  
.:C808 54 48 45 20 20 20 20 20
```

So, typing in OPEN RED DOOR would put the same values in locations 52689 and 52690 as before, but also a 1 into location 52691, and your program would now know that the player had typed in the third verb, the first link word and the second noun.

By judicious use of link words you can get some reasonably complicated inputs, and your adventures will grow accordingly.

Conclusion

Understanding what the player has typed in is one of the many keys to producing a great adventure. If your room description contains the sentence:

THERE IS A BLUE DOOR TO THE NORTH, AND A SOLID WALL TO THE EAST.

and a player types in:

SMASH THE WALL

it's not going to look very good if the program responds with:

I DON'T KNOW WHAT A WALL IS.

or even worse:

THERE ISN'T A WALL HERE.

Just because you expect the player to type in **OPEN BLUE DOOR** doesn't mean that he will. So be careful, and with a little bit of time and effort while writing the game you'll be on your way to rivalling Level Nine.

However, one of Level Nine's big secrets is cramming large room descriptions into small portions of memory, so we'll take a look at how to do that in Chapter 7.

7

Text Compression

Back in the days of innocence and *Exploring Adventures on the 64* we never talked about text compression at all. Room descriptions were kept short, usually less than two screen lines in length, and as a result of this those of you who are masters of producing glorious prose were a little bit hampered.

In those early programs we used strings because, quite frankly, it is a lot easier for a beginner. If you have an adventure with, say, 50 locations in it, then dimension a string P\$(50), you could have P\$(1) containing the description for location one, P\$(2) for room two, and so on. However, this led to room descriptions of the 'YOU ARE IN A DARK ALLEYWAY' type, which doesn't really tell the player much about what is going on.

I did see a couple of games based on the 'Exploring Adventures' series that had used ordinary PRINT statements to print out the room descriptions (see 'The Lost Wallet' program in Chapter 8 for an example of this). The only trouble with this approach is that it eats up memory, always a precious commodity in adventure games. It also requires a nest of ON...GOTO loops to direct program flow to the correct description. You could use a computed GOTO as described in Chapter 4, but you would still sacrifice vast amounts of memory.

The solution is obviously to compress the text in the program, and then expand it again when it comes to printing out the location descriptions (or messages) on the screen. The method I'm about to describe is not the most memory efficient in the world, although it will save you up to around 30 per cent: not to be sneezed at.

It has the merit of being pretty easy to understand, and as long as you have a pencil and paper ready when you type in your location descriptions, it is also fairly easy to use.

Tokenising

As you go through your room descriptions, it soon becomes apparent that a large number of words are used over and over again. Words like AND and THE crop up quite frequently, and not only by themselves eiTHER! Words like raTHER, anoTHER, and so on, all have the word THE contained in them. Similarly, rANdOm, abANDon and others all have AND in them. You might find that you're using the word DOOR a lot, or PATH, or any one of a group of words.

Rather than typing out the same word over and over again, wouldn't it be better to replace it with a symbol (or token) which, when encountered in the routine that prints out location descriptions, could be expanded back to the original word again.

Let's take a simple example.

Z\$ = "There is a [cbmd] to the north."

where [cbmd] indicates 'press the CBM logo key and the d key at the same time'.

Here I've assumed that the adventure is being written in upper/lower case, which leaves us with about sixty symbols, or tokens, that we can conveniently use.

What we need to do now is to write a short routine that looks at the string Z\$ and checks every character in it. Perhaps something like:

```
1000 FOR I=1 TO LEN(Z$)
1002 A$ = MID$(Z$,I,1):A = ASC(A$)
1004 IF A = 172 THEN PRINT "DOOR";:GOTO1008
1006 PRINTA$;
1008 NEXT I
1010 RETURN
```

Not very satisfactory, but we've got to start somewhere.

In this short example we're checking the ASCII value of each character contained within the string Z\$. If that ASCII value is equal to 172, then we print out the word DOOR, otherwise we print out

that part of the string which has just been checked.

Obviously you'd have more than one token within your programs, and so the routine would need to be expanded a little:

```
1000 FOR I=1 TO LEN(Z$)
1002 A$=MID$(Z$,I,1):A=ASC(A$)
1004 IF A=172 THEN PRINT "DOOR";:GOTO1020
1006 IF A=174 THEN PRINT "SOUTH";:GOTO1020
1008 IF A=170 THEN PRINT "NORTH";:GOTO1020
1010 .....
1012 .....
1014 etc.
1018 PRINT A$;
1020 NEXT I
1022 RETURN
```

This is all very well when there are only three or four tokens to be checked, but if you're using sixty or more life could get a little hairy, and the amount of memory needed to check every symbol becomes a bit ridiculous.

This particular Basic approach could be shortened, by (for example) assigning a string C\$ to be equal to all the tokenised characters. Thus you might have something like:

```
C$="[cbma,cbmb,cbmc,cbmd ... etc.]"
```

and another array B\$(60) to contain all the words associated with C\$. For instance, if [cbma] was meant to represent the word 'corridor', the first entry in the array B\$ would be:

```
B$(1)="corridor"
```

Then the program routine to check for all the tokens might look like this:

```
1000 FOR I=1 TO LEN(Z$)
1002 A$=MID$(Z$,I,1)
1004 FOR J=1 TO 60
1006 IF C$(J)=A$ THEN PRINT B$(J);:GOTO1012
1008 NEXT J
1010 PRINT A$;
1012 NEXT I
1014 RETURN
```

Not a very satisfactory program, in that we're possibly jumping out of the inner J loop, but it does work. The major drawback to this approach is that it takes a fair amount of time to print out any particular location description (or message), and even when it does print it out it looks very jumpy on the screen: some words (the tokenised ones) are printed out very quickly, while others are printed out one character at a time.

So, as ever, the solution has to lie in machine code, and the following collection of routines does the job for us.

Into machine code

For the purpose of this routine, we're assuming that the special characters used have ASCII values in the range 133 to 192. Only one character within this range might cause us any problems, and that is ASCII character 157: the ASCII value of the cursor left key. We'll see why in a moment.

The first thing to do is to find a spare portion of memory. Here I've used locations \$C900 to \$CA66, but you can relocate the code to be just about anywhere you like.

Then you must decide what each token will represent. The listings given here come from a real-life adventure, they weren't just made up for this book. We start with token number 133 (the first function key 'F1'): I decided to let this indicate the letters 'EAD': a common occurrence in various words.

'F3' indicates the letters 'ERS', 'F5' indicates the word 'BLOW' (the game had a nautical flavour, and winds kept blowing from various directions), and so on throughout the rest of the ASCII characters up to and including ASCII character 192 (which incidentally represents 'IN ': note the space).

Now all the room descriptions can be typed in. Say, for example, room one has the description:

'YOU ARE IN A LOW CHANNEL.'

Obviously you'll want to use longer descriptions than that, but just for the sake of an example we'll stick to something short and sweet. The word 'YOU' is, in this list of tokens, represented by ASCII character 183, produced by typing in [cbm and y]. The word 'ARE'

is represented (don't ask me why!) by [cbm and m], 'IN' is [shift and *], 'A LOW' is typed in as normal, and 'CHANNEL' is [ctrl and 4].

So, instead of having:

Z\$="YOU ARE IN A LOW CHANNEL."

We now have:

Z\$="[cbmy,cbmm,shift*]A LOW [ctrl4]."

Typed in using tokens instead of words, this gives us a Z\$ containing 11 characters, instead of 25. In this case, then, a saving of over 50 per cent in terms of memory.

To go through the first 8 characters used in this example, we come up with the following table:

ASCII value	Key combination	Word
-----	-----	-----
133	F1	EAD
134	F3	ERS
135	F5	BLOW
136	F7	TER
137	F2	ARD
138	F4	FROM
139	F6	IGHT
140	F8	OUT

Now for the important part, getting the machine-code listing working!

```
B*
      PC  SR  AC  XR  YR  SP
.:5766 33 00 4E 00 F6
.
.:C91A 03 03 04 03 03 04 04 03
.:C922 00 00 00 00 00 00 00 00
.:C92A 00 05 04 03 07 04 06 03
.:C932 00 05 07 05 03 03 03 03
.:C93A 03 03 03 04 03 05 04 03
.:C942 03 05 04 04 04 04 04 04
.:C94A 07 03 03 05 04 04 06 04
.:C952 04 03 03 03 00 03 06 0A
.:C95A 0D 10 14 1B 1B 1B 1B 1B
.:C962 1B 1B 1B 1B 1B 1B 20 24
```

```

.:C96A 27 2E 32 38 3B 3B 40 47
.:C972 4C 4F 52 55 58 5B 5E 61
.:C97A 65 68 6D 71 74 77 7C 80
.:C982 84 88 8C 90 94 9B 9E A1
.:C98A A6 AA AE B4 B8 BC BF C2
.:C992 C5 45 41 44 45 52 53 42
.:C99A 4C 4F 57 54 45 52 41 52
.:C9A2 44 46 52 4F 4D 49 47 48
.:C9AA 54 4F 55 54 52 4F 55 4E
.:C9B2 44 44 4F 57 4E 4F 4E 20
.:C9BA C3 48 41 52 59 42 44 54
.:C9C2 48 41 54 D3 43 59 4C 4C
.:C9CA 41 54 4F 20 57 48 49 43
.:C9D2 48 43 48 41 4E 4E 45 4C
.:C9DA 57 48 45 52 45 4E 4F 57
.:C9E2 D4 48 45 54 48 45 4F 46
.:C9EA 20 D9 4F 55 41 4C 4C 41
.:C9F2 52 45 53 41 49 4C 49 53
.:C9FA 20 CE 4F 52 54 48 D4 52
.:CA02 4F 59 41 4E 44 41 56 45
.:CA0A D3 4F 55 54 48 57 49 54
.:CA12 48 53 48 49 50 C5 41 53
.:CA1A 54 4F 55 47 48 D7 45 53
.:CA22 54 48 49 4C 4C 48 41 52
.:CA2A 42 4F 55 52 49 4E 47 59
.:CA32 4F 55 57 48 49 4C 45 4C
.:CA3A 41 4E 44 45 41 43 48 49
.:CA42 53 4C 41 4E 44 57 49 4E
.:CA4A 44 53 49 44 45 41 4E 54
.:CA52 53 45 41 49 4E 20 BC 56
.

```

If you look at memory locations \$C91A to \$C955 you'll see a collection of numbers from 0 to 7. These indicate the length of each word. From the table above you can see that ASCII key 133 has a length of 3 characters, and so the first entry in this table is a 3. ASCII key 135 has 4 characters in it, and so the third entry has a 4 in it. Type in all the numbers depending on which words you choose to represent as symbols.

Now look at memory locations \$C956 to \$C992. These are used to tell the machine-code program the location of the word represented by each token. The words are stored in ASCII format in locations \$C993 to \$CA56.

The first word, 'EAD', has 3 letters in it (memory location \$C91A). The letters 'EAD' are stored in memory in locations \$C993, \$C994 and \$C995. Thus they are offset from the start of the word table by nothing (they're right at the start of the table), and so the entry

at location \$C956 is a 0: look right at the start of the listing for the first word.

The second word, 'ERS', also has 3 letters in it (memory location \$C91B). The letters 'ERS' are stored in locations \$C996, \$C997 and \$C998. Thus they are offset from the start of the word table by 3 characters, and so the entry at location \$C957 is a 3: look 3 characters into the word table for the second word.

The third word, 'IGHT', has 4 letters in it (memory location \$C91C). The letters 'IGHT' are stored in locations \$C999, \$C99A and \$C99B. Thus they are offset from the start of the word table by 6 characters (the previous two words both had 3 characters in them), and so the entry at location \$C958 is a six: look 6 characters into the word table for the third word.

The fourth word, 'TER', again has 3 letters in it (memory location \$C91D). The letters 'TER' are stored in locations \$C99C, \$C99D and \$C99E. Thus they are offset from the start of the word table by 10 characters (3 + 3 + 4 for the previous 3 words), and so the entry at location \$C959 is a 10 (\$0A): look 10 characters into the word table for the fourth word.

And so it goes on for all 60 words. Some of them weren't used (hence the zeros every now and again), and we eventually end up with three separate tables for the complete collection of words. One for how long each word is, one showing where it starts in memory offset from the start of the word table, and then the word table itself.

Obviously this will change from adventure to adventure, and you won't always choose words that are the same length as the ones I used in this particular adventure. No problem, simply alter the entry for the length of any word that has been changed, alter the location of that word in the word table, and finally alter the word table itself.

The location of the words in the table might seem a daunting thing to type in, but it isn't really. A simple FOR ... NEXT loop and a PEEK and a POKE do the job for us. Like this:

```
B=0:FOR I=0 TO 59:A=PEEK(51482+I):B=B+A:POKE51579+I,B:NEXT
```

which just looks at all the word lengths and collectively adds them together.

All we need now is the machine-code routines to activate these tables, and the Basic routine to call them up.

```

B*
      PC  SR  AC  XR  YR  SP
.:77B2 33 00 9A 00 F6
.
C901 AD 00 C9      LDA $C900
C904 C9 85        CMP #$85
C906 30 0E        BMI $C916
C908 C9 C1        CMP #$C1
C90A 10 0A        BPL $C916
C90C E9 84        SBC #$84
C90E AA          TAX
C90F BD 1A C9     LDA $C91A,X
C912 4C 5B CA     JMP $CA5B
C915 EA          NOP
C916 20 D2 FF     JSR $FFD2
C919 60          RTS
.
.

```

```

B*
      PC  SR  AC  XR  YR  SP
.:67BC 33 00 74 00 F6
.
CA5B BC 56 C9     LDY $C956,X
CA5B AA          TAX
CA5C B9 93 C9     LDA $C993,Y
CA5F 20 D2 FF     JSR $FFD2
CA62 CB          INY
CA63 CA          DEX
CA64 D0 F6       BNE $CA5C
CA66 60          RTS
.
.

```

Having got our enormous string Z\$ to contain the room description, the Basic activator might look something like this:

```

10000 FOR I=1 TO LEN(Z$)
10002 A=ASC(MID$(Z$,I,1)):POKE51456,A:SYS51457:NEXT
10004 RETURN

```

The RETURN assumes that you're calling this as part of a subroutine.

The first part of the machine-code listing looks at the ASCII value of the character stored in memory location 51456. If it's less than 133 (\$85) or greater than 192 (\$C1) then just jump to the internal

ROM routine at \$FFD2 and print the character out.

Otherwise, subtract 132 from it (\$84), load the accumulator with the length of the word from the first of our three tables, and jump to the next part of the routine, starting at memory location \$CA58.

This then finds where the word starts in the word table, and carries on reading and printing characters from that table until it comes to the end of the word: in other words, it's reached the end of the word length.

If you prefer all this in Basic loader form:

```
10 SE = 51456 : EE = 51814
20 :
30 FOR I = SE TO EE : READ A : POKE I , A : NEXT
40 :
50000 DATA 179,173,000,201,201,133,048,014,201,193
,016,010,233,132
50001 DATA 170,189,026,201,076,088,202,234,032,210
,255,096,003,003
50002 DATA 004,003,003,004,004,003,000,000,000,000
,000,000,000,000
50003 DATA 000,005,004,003,007,004,006,003,000,005
,007,005,003,003
50004 DATA 003,003,003,003,003,004,003,005,004,003
,003,005,004,004
50005 DATA 004,004,004,004,007,003,003,005,004,004
,006,004,004,003
50006 DATA 003,003,000,003,006,010,013,016,020,024
,027,027,027,027
50007 DATA 027,027,027,027,027,027,032,036,039,046
,050,056,059,059
50008 DATA 064,071,076,079,082,085,088,091,094,097
,101,104,109,113
50009 DATA 116,119,124,128,132,136,140,144,148,155
,158,161,166,170
50010 DATA 174,180,184,188,191,194,197,069,065,068
,069,082,083,066
50011 DATA 076,079,087,084,069,082,065,082,068,070
,082,079,077,073
50012 DATA 071,072,084,079,085,084,082,079,085,078
,068,068,079,087
50013 DATA 078,079,078,032,195,072,065,082,089,066
,068,084,072,065
50014 DATA 084,211,067,089,076,076,065,084,079,032
,087,072,073,067
50015 DATA 072,067,072,065,078,078,069,076,087,072
,069,082,069,078
```

```

50016 DATA 079,087,212,072,069,084,072,069,079,070
,032,217,079,085
50017 DATA 065,076,076,065,082,069,083,065,073,076
,073,083,032,206
50018 DATA 079,082,084,072,212,082,079,089,065,078
,068,065,086,069
50019 DATA 211,079,085,084,072,087,073,084,072,083
,072,073,080,197
50020 DATA 065,083,084,079,085,071,072,215,069,083
,084,072,073,076
50021 DATA 076,072,065,082,066,079,085,082,073,078
,071,089,079,085
50022 DATA 087,072,073,076,069,076,065,078,068,069
,065,067,072,073
50023 DATA 083,076,065,078,068,087,073,078,068,083
,073,068,069,065
50024 DATA 078,084,083,069,065,073,078,032,224,024
,240,015,188,086
50025 DATA 201,170,185,147,201,032,210,255,200

```

READY.

You may remember what we said earlier on about ASCII value 157, the cursor left key. You might sometimes want to use this in a room description, and the routine as it stands will go haywire if it encounters ASCII 157, since the table indicates that it hasn't got any characters in it.

You'll have to insert a check at location \$CA58 to look for this character:

```

CMP #$18
BEQ $CA6B
.
.
.
LDA #$9D
JSR $FFD2
RTS

```

if you happen to want to use it.

Conclusion

The resulting output is satisfactorily fast (although not so fast that someone who can read very quickly couldn't keep up with it).

It looks professional, and allows your imagination scope to roam as you come up with the sort of lengthy room descriptions that might put Level Nine or Infocom to shame. Since it sits in the spare block of memory from locations \$C000 to \$CFFF it doesn't take up any of the available Basic memory: another bonus.

Now that we've looked at all sorts of routines, parsers, text compression techniques, drawn maps and done just about everything else, let's write our first adventure using all the things that we've learnt so far.

One of the criticisms of my earlier book, *Exploring Adventures on the 64*, was that it presented a lot of large adventures but no little ones: in other words the would-be adventure writer was plunged right in at the deep end. I've taken this opportunity to amend that, and in the next chapter we're going to write a small but advanced adventure called 'The Lost Wallet'.

8

A Tape-based Adventure

Putting together your first adventure can be a daunting experience, so this (relatively) short program, 'The Lost Wallet', has been concocted so that you can get some idea of how the parts come together to form the whole.

In Chapters 6 and 7 we studied various techniques for producing parsers and compressing text. In this program we won't go into text compression at all, for the simple reason that my printer refuses to print out much more than the standard ASCII character set, and the methods discussed in Chapter 7 rely on all sorts of strange characters being used.

However, you are certainly welcome to try to compress the text if you wish. The parser used in the program is the more advanced of the two Basic ones presented in Chapter 6. This allows the player to type in relatively complex instructions, while still letting the programmer alter the adventure quite easily. Also, trying to absorb your first adventure is a lot easier if the entire program is written in Basic.

Almost all of 'The Lost Wallet' is in Basic, but there is one short machine-code routine used that allows the Commodore 64 to have a computed GOTO statement. In other words, instead of having a whole sequence of ON VB-10 GOTO 1000,1010,1020 or whatever, we can now say `SYS 49152,800 + VB*10,0`: a great memory saving, and a great increase in the speed and efficiency of the program.

To begin with, though, every adventure should have an introduction.

Introducing the game

A simple, straightforward exercise in using the PRINT statement, which looks like this:


```

10000 POKE53281,7:POKE53280,9:PRINTCHR$(14):PRINTC
HR$(8):PRINT"[CLR,BLK]THE LOST WALLET!
10002 FORI=1TO16:PRINT"[CBM]";:FORJ=1TO50:NEXTJ,I
10004 PRINT:PRINT"ONE DAY, WHILST STROLLING AROUND
IN
10006 PRINT"LONDON DOING SOME SHOPPING, YOU MUST
10008 PRINT"HAVE LOST YOUR WALLET.
10010 PRINT:PRINT"WHEN YOU GOT HOME, IT WAS NO LON
GER
10012 PRINT"THERE.  OF COURSE, YOU CAN'T REMEMBER
10014 PRINT"WHERE YOU DROPPED IT, AND SO THE GOAL
10016 PRINT"IN THIS MINI-ADVENTURE IS TO FIND YOUR
WALLET AGAIN."
10018 PRINT:PRINT"PLEASE HANG ON WHILE I READ SOME
DATA  IN."
10020 RETURN

```

This sets the background and border colours, goes into upper/lower case mode, prevents people using the CBM Logo and shift keys together to revert to upper case/graphics mode (that's the CHR\$(8) statement), before clearing the screen and setting the print colour to black.

Once a few instructions have been printed up the player is invited to wait while some data is read in. We encountered all the data in Chapter 5, so if you're setting about the marathon task of typing this whole listing in, load that data from wherever it was saved and begin by typing in the introduction.

The first few lines of the program control a number of things, so here they are:

```

0 TI$="000000"
1 GOSUB10000:CP=1
2 GOSUB5000
3 PRINT:PRINT"PRESS SPACE TO START."
4 GETSP$:IFSP$<>" "THEN4
5 SQ$="YOU FIND SOMETHING OF INTEREST."
10 GOSUB20000
15 GOSUB3000
20 GOSUB390

```

Line 0 simply sets the internal clock to zero, so that the room description contained in lines 4090-4097 can tell you how long you've been playing the game. The GOSUB to line 10000 prints up the instructions already encountered, while the variable CP is declared to be equal to one. In other words, your Current Position is in location one, the start of the game.

Line 2 is a GOSUB to read in all the data which was encountered in Chapter 5. Lines 3 and 4 are a simple message and wait for a key to be pressed before the game can begin, and line 5 declares a string variable. The GOSUB to line 20000 in line 10 reads some data in, and we'll look at this in a moment. Line 15 GOSUBs to line 3000, which is part of the routine that tells you where you are, what you can see, and where the obvious exits are, while line 20 GOSUBs to line 390 and really starts the game going: this is the input parser.

Computed GOTOs

```
20000 FORI=49152TO49162:READA:POKEI,A:NEXT:RETURN
20001 DATA 104,104,32,253,174,32,235,183,76,166,16
8
```

This short machine-code routine is the machine-code computed GOTO we talked about earlier. A mere eleven bytes long, it takes up no space in the 38K or so that is available for use, since it resides in the spare 4K block of memory starting at location \$C000. It also saves a lot of space by replacing six or more lines of ON VB GOTO with a single SYS call.

Where are we?

```
3000 PRINT"[CLR,BLK]";:SYS49152,3990+CP*10,0
3002 IFEO=1THENPRINT:PRINT"AH, THAT LOOKS LIKE THE
END OF THE GAME TO ME.":GOTO92
3004 PRINT:PRINT"[RVS,BRN]YOU CAN SEE:[BLK]
3006 A=0:FORI=1TONN:IFOB%(I)=CPTHENPRINTOB$(I):A=A
+1
3008 NEXT:IFA=0THENPRINT"NOTHING MUCH."
3010 PRINT:PRINT"[RVS,ORNG]OBVIOUS EXITS ARE:[BLK]
3012 A=0:FORI=0TO3:IFP%(CP,I)<>0THENPRINTDD$(I):"
";:A=A+1
3014 NEXT:IFA=0THENPRINT"NOWHERE."
3015 PRINT
3099 RETURN
```

Line by line then:

Line 3000: clear the screen, set the text colour to black, and jump to line number $3990 + CP * 10$. So if you're in location one, program flow goes to line 4000 ($CP = 1$), location two, then off to line 4010, and so on.

Line 3002: if the variable EO has been set, then the game is over, so print a suitable message and go to line 92, of which more later.

Line 3004: print, in reverse field brown, the message 'YOU CAN SEE:'.

Line 3006: set the variable A to zero, go through a loop of all the objects to see if any of them are in the current location, and if they are then print the long object description and increment the variable A.

Line 3008: next step on the loop, and if at the end of it the variable A is still equal to zero then inform the player that there's not very much to look at in this location.

Line 3010: print in reverse field orange the message 'OBVIOUS EXITS ARE:'.

Line 3012: again set the variable A to equal zero and check for all the values of P%(CP,I) to determine whether or not an exit exists in a particular direction. If it does then print the appropriate part of the variable DD\$ and increment the variable A.

Line 3014: continue on round the loop, and if at the end of it the variable A is still equal to zero then tell the player that there is nowhere obvious to go.

Line 3015: a simple PRINT statement to tidy up the screen output.

Line 3099: return from this routine.

As you can see, there are a lot of spare line numbers in the region 3015-3099. If you wish to 'beef up' this program, or use this set of routines and subsequently amend them in your own programs, then there's plenty of spare space in there.

Input parser

```
390 PRINT:PRINT"[RVS,PUR]WHAT NOW ?[OFF] ";:IFA1=1  
THENPRINTA1$;:CM$=A1$;PRINT:LC=LEN(CM$);PRINT:GOTO  
392  
391 GOSUB60000:PRINT:LC=LEN(CM$)  
392 IFOB%(19)=-1THENMI=MI+1:IFMI=10THENB50  
393 NO$="":VB$="":VB=0:NO=0:IFA1=1THENA1=0:GOTO396
```

```

394 A1=0:FORI=1TOLC:IFMID$(CM$,I,5)=" AND "THENA1$
=RIGHT$(CM$,LC-I-4):A1=1:A2=I
395 NEXT:IFA1=1THENCN$=LEFT$(CM$,A2-1)
396 FORI=1TOLC:IFMID$(CM$,I,1)<>" "THENVB$=VB$+MID
$(CM$,I,1):NEXT
397 V1$=VB$:VB$=LEFT$(V1$,4):FORI=1TONV:IFVB$(I)=V
B$THENVB=I:GOTO402
398 NEXT
400 VB=7:NO$=VB$:GOTO406
402 IFLEN(V1$)+1>=LEN(CN$)THENNO=0:RETURN
403 FORI=LCTO1STEP-1:IFMID$(CN$,I,1)<>" "THENNO$=N
O$+MID$(CN$,I,1):NEXT
404 NO$=RIGHT$(NO$,4):FORI=1TO4:A$(I)=MID$(NO$,5-I
,1):NEXT
405 NO$="":FORI=1TO4:NO$=NO$+A$(I):NEXT
406 FORI=1TONN:IFNO$=NO$(I)THEN409
407 NEXTI
408 NO=0:RETURN
409 NO=I:RETURN

```

```

60000 CM$="":
60002 PRINT"[RVS,BLK]*[OFF,CL]";
60005 GETZ$:IFZ$=""THEN60005
60010 Z=ASC(Z$):IFZ>95THEN60005
60080 ZL=LEN(CM$):IFZL>27THEN60110
60100 IFZ>31THENCN$=CM$+Z$:PRINTZ$:GOTO60002
60110 IFZ=13ANDZLTHENPRINT" ":RETURN
60120 IFZ=20ANDZLTHENCN$=LEFT$(CM$,ZL-1):PRINTZ$:
60140 GOTO60002

```

This is the input routine discussed in Chapter 6. The only part not really covered there is line 392. This checks to see whether you're carrying an ice-cream, and if you are then the counter MI is incremented. Since this adventure all takes place on a hot day, if you carry the ice-cream about for more than ten moves it melts, and a suitable message to that effect is printed up in line 850. We'll see that a little later on.

After coming back from the input routine, a few simple checks are needed on what was typed in, and this forms the next part of the program.

Input checking

```

21 IFNO<0THENPRINT"PARDON?":GOTO20
22 A=0:FORI=1TOLEN(VB$):A=A+ASC(MID$(VB$,I,1)):NEX
T

```

```

23 IF (A=319ORA=297ORA=301)ANDVB$<>"HELP"THENPRINT "
WATCH IT!":GOTO20
24 IFVB$="WHY"THENPRINT"BECAUSE I SAID SO!":GOTO20
25 IFVB=7ANDVB$<>"GO"THENPRINT"I'M SORRY, BUT I DO
N'T UNDERSTAND THAT.":GOTO20
26 IFNO=0ANDNO$<>" "THENPRINT"[RVS]#HAT[OFF] DO YOU
WANT TO "V1$"?":GOTO20
27 IF (VB>12ANDVB<>28ANDVB<>29ANDVB<>36) ANDNO=0THEN
PRINT"DO WHAT?":GOTO20
30 SYS49152,50+VB*30,0

```

Line 21: if the value NO is less than zero this means that an unintelligible word was typed in, so print out a message and go back for more.

Line 22: declare our useful variable A to be equal to zero, and for each letter in the verb part of the input string (VB\$) take the ASCII value of that character and add it to the variable A.

Line 23: if the verb string isn't the word HELP, but does end up with one of the three values quoted in this line, then someone has typed in a rude word, so tell the player to 'watch it' and carry on with the game. This is the method mentioned earlier of checking for rude words without actually including them in the listing. If you want to make your adventures even more cryptic than this listing you could check for all verbs and nouns by taking the ASCII values, but remember that, as checked for here, different words can give the same result. What rude word is the same as typing in HELP, you may well wonder!

Line 24: if the computer has responded to a certain request by saying NO, or something like that, the player may well be tempted to type in WHY NOT. If he does, this line takes care of it.

Line 25: the input parser returns a value of seven as a verb number for any verb that it doesn't understand, so if that happens, and the verb typed in isn't the seventh one in our data list (the verb GO) then tell the player that you don't understand what he's typed in.

Line 26: if there is an unrecognised noun, but a noun has definitely been typed in (in other words the noun typed in is not part of our data list) then tell the player that it is an unknown one by printing out a suitable message. I realise now that this could lead to some strange things being printed out, but you only get out of an adventure what you type into it!

Line 27: if the verb is one of the group quoted, and no noun has

been typed in, then tell the player that he's trying to do something ridiculous. For instance, if the player just typed in GET, instead of GET SOMETHING, this is the response that would be used.

Line 30: our machine-code computed GOTO command. In other words, take the verb number VB, multiply it by 30, and go to the resulting line number *plus* 50. 30 lines of code per verb should be ample for most purposes.

The verbs occupy the space from line 80 to line 1132. Rather than give it as just a great big list of verbs, presumably unintelligible to most people, we'll split up each part of the code, and put the code on one page and the description for the code on the facing page. This is following the method used in the 'Exploring Adventures' series and should enable both newcomers and old hands to see what is going on.

The complete list of verbs

As already noted there are some 36 verbs in all, and starting with the first one here we go with a blow by blow description of what they all do.

QUIT

This is the verb used whenever a player wants to quit from the game currently in progress. As should always be the case (and if any reviewer is reading this, yes, I know I break my own rules occasionally!) there is a check to make sure that the player really wants to do this.

```
80 REM QUIT
82 PRINT:PRINT"ARE YOU SURE (Y OR N)?"
84 GETAS$: IFAS$="Y"THEN92
86 IFAS$="N"THEN20
88 GOTO84
92 PRINT:PRINT"/PLAY AGAIN (Y OR N)?"
94 GETAG$: IFAG$="Y"THENRUN
96 IFAG$="N"THENPRINT"[CLR]BYE ...":END
98 GOTO94
```

Explanation

Line 80: an explanatory REM statement.

Line 82: print the 'ARE YOU SURE' message.

Line 84: wait for a key to be pressed, and if the player types 'y' then GOTO line 92.

Line 86: if the player types 'n' then return to line 20 for more input.

Line 88: if the player types neither then sit and wait until he does.

Line 92: see if the player wants to have another go at the game.

Line 94: if he does then simply RUN the program.

Line 96: if he doesn't, then clear the screen, print a final message, and end the program.

Line 98: the player must press 'y' or 'n' to get out of this loop, so wait until he does.

HELP

Every adventure player wants some help at some time or another, and so this little piece of code takes care of that.

```
110 PRINT"I'M AFRAID THAT I CAN'T GIVE YOU ANY  
HELP AT ALL.":GOTO20
```

Explanation

Just a simple PRINT statement to inform the player that he's going to get no help at all in this game.

SCORE

Something we all want to do at some time or another, in adventure games you're usually trying to find out how well, or otherwise, you have done in the game.

```
140 PRINT"IF YOU HAVEN'T FOUND YOUR WALLET THEN  
YOU HAVEN'T SCORED ANYTHING.":GOTO20
```

Explanation

No explanation required really, just another PRINT statement to tell the player that, until he finds his wallet, he hasn't scored anything.

INVENTORY, LOOK and REDESCRIBE

Three verbs in one now. INVENTORY lets you see all the things that you're carrying, while LOOK and REDESCRIBE give you another look at the room description in case you forget what has been printed up on the screen.

```
170 ZZ=0:PRINT"[RV$ ,BRN]YOU ARE CARRYING:[BLK]  
172 FORI=1TONN:IFOB$(I)=-1THENPRINTOB$(I):ZZ=ZZ+1  
174 NEXT:IFZZ=0THENPRINT"NOTHING MUCH."  
176 GOTO20  
200 GOTO15  
230 GOTO15
```

Explanation

Line 170: set the variable ZZ, which determines how many objects you're carrying, to zero, and print in reverse field brown the message 'YOU ARE CARRYING:'.

Line 172: set up a loop to go through the entire list of objects to see if their current value is equal to -1. If it is, this means that it's being carried, so print out the longer object description and increment the variable ZZ.

Line 174: next step through the loop, and if at the end of it the variable ZZ is equal to zero, then tell the player that he's carrying nothing.

Line 176: back to the input routine again.

Line 200: go to line 15 to start of the room description sequence.

Line 230: same as line 200.

GO and JUMP

Players try to GO to all sorts of places, but in this game we only allow them to go through the cardinal compass points of NORTH, SOUTH, EAST and WEST. If anything else is tried, then the program is not impressed.

Similarly with JUMP. People will try to jump all over the place, but in this game it doesn't fill any useful purpose whatsoever.

```
260 IFNO=5ORNO>9THENPRINT"I CAN'T DO THAT.":GOTO20
261 IFNO>5THENNO=NO-5
262 NO=NO-1:GOTO322
290 PRINT"YOU ARE NOT A KANGAROO, JUST GET ON WITH
IT.":GOTO20
```

Explanation

Line 260: if the player has entered the fifth noun (the word KEY), or a noun whose noun number is greater than nine, then tell the player that he's trying to do something stupid before going back to line 20.

Line 261: if the noun number is greater than five, subtract five from it. In other words, if the player types GO NORTH, he really should have typed in GO N.

Line 262: subtract one from the noun number before going to the routine starting at line 322 which deals with all player movement. This enables us to use the P% variable discussed earlier.

Line 290: a simple message to tell the player that there's no point in jumping anywhere.

N, S, E and W

If a player can't be bothered to enter the word GO, and just wants to move North, South, East or West, then all he has to do is enter the single letter N, S, E or W. The following routines deal with this, along with a couple of special cases where individual locations have to be checked.

```
320 IFCP=4ANDDO=0THENPRINT"THE DOOR IS LOCKED.":GO
    TO20
321 NO=VB-9
322 IFP%(CP,NO)<>0THENCN=P%(CP,NO):GOTO15
324 PRINT"YOU CAN'T GO THAT WAY.":GOTO20
350 IFCP=3ANDDO=0THENPRINT"THE DOOR IS LOCKED.":GO
    TO20
354 GOTO321
380 IFCP=11ANDCL=0THENPRINT"NOT WITHOUT A LIBRARY
    TICKET.":GOTO20
382 GOTO321

410 GOTO321
```

Explanation

Line 320: trying to move north in location four if the door is locked ($DO = 0$) is unproductive, so tell the player so and go back to line 20.

Line 321: make the noun number equal to the verb number minus nine, in order that the program can use the P% variable to determine whether or not a player can move in a particular direction.

Line 322: if he can, i.e. the value of P%(room number,noun number) doesn't equal zero, then put the player in the new location and go to line 15 to tell the player where he now is.

Line 324: the only other alternative, namely that the player can't go in that particular direction.

Line 350: trying to move south from location three if the door is locked is not very profitable, so tell the player so and go back to line 20.

Line 354: no other checks for moving south, so go to line 321 and carry on from there.

Line 380: you can't move east out of the library at location eleven if you've left your library ticket behind, so tell the player so and go back to line 20.

Line 382: no other checks for moving east, so go off to line 321 and carry on from there.

Line 410: no checks at all for moving west (freedom!), so just go to line 321 and sort it out from there.

GET

In adventure games players try to GET all sorts of things, they may even tell the adventure itself to GET 'somethinged', so a routine has to be written that attempts to cope with most eventualities.

There will be many things that the player cannot get hold of, for a variety of reasons. Walls, rooms etc. are not easily carried, and the player may try to get something that hasn't been discovered yet. In some games the player may well try to get something that is out of reach, and so a little piece of code would need to be inserted to tell the player that he isn't tall enough, or something like that.

Happily there are no such complications in this simple game, and so the code is really quite straightforward.

```
440 IF OB%(NO)=-1 THEN PRINT "YOU ALREADY HAVE.": GOTO 200
442 IF ZZ>4 THEN PRINT "YOU'RE CARRYING TOO MUCH.": GOTO 200
444 IF (NO<10 AND NO<>5) OR (NO>18 AND NO<>24) THEN PRINT "I WON'T LET YOU.": GOTO 200
446 IF NO=11 OR NO=13 OR NO=14 OR NO=15 THEN PRINT "DON'T BE SILLY.": GOTO 200
458 IF OB%(NO)<>CP THEN PRINT "I CAN'T SEEM TO SEE THAT.": GOTO 200
459 IF NO=18 THEN CL=1
460 PRINT "OK.": ZZ=ZZ+1: OB%(NO)=-1: GOTO 200
```

Explanation

Line 440: if the object is in the player's possession then tell him so, and go back to line 20.

Line 442: if the number of objects being carried is greater than four, then the player cannot carry any more.

Line 444: various objects that can't be GOT.

Line 446: and ditto for a few more objects.

Line 458: if the object isn't in the room then tell the player so before going back to line 20.

Line 459: if he's getting the library ticket then set the variable CL to show that he's carrying it.

Line 460: everything's okay, so increment the object counter, put the object in the player's possession, tell him it's 'OK', and go back to line 20.

DROP

If you're going to be able to get things, you should also be able to drop them, and this is the purpose of this piece of code.

Nothing too complicated here. Some adventures would have a much longer routine to handle this particular verb, as dropping an object can occasionally cause something to happen. Dropping a bottle, for instance, might well break it, or, as in the original 'Adventure', if you dropped a valuable Ming vase without there being a velvet pillow in the room to catch it safely, it shattered into a thousand pieces.

```
470 IF OB%(NO)<>-1 THEN PRINT "YOU HAVEN'T GOT IT." : GO  
TO 20  
472 IF NO=24 THEN WD=0  
474 IF NO=18 THEN CL=0  
480 PRINT "OK." : ZZ=ZZ-1 : OB%(NO)=CP : GO TO 20
```

Explanation

Line 470: if the player isn't carrying the object then tell him so, before going back to line 20.

Line 472 : if the player is literally dropping his disguise, then set the variable WD to zero.

Line 474: if the player is dropping the library ticket, then set the variable CL to zero.

Line 480: nothing else to check, so print up 'OK', decrement the object counter, put the object in the room, and then off to line 20 as usual.

Incidentally, the WD variable in line 472 is used in one of the room descriptions to check to see whether a player can get past the policeman or not. If you're wearing the disguise you're okay, but if not, then you're recognised and carted off to the station. This is location four, and the relevant lines of code are 4036-4039.

OPEN

Traditionally there are many things that an adventure player might try to open, but most games limit your choices somewhat to mundane things such as doors and, in this game, your wallet - if ever you find it, that is.

However, this doesn't stop people trying to open everything they come across, and so you have to check and ensure that no ridiculous follies remain in the program.

```
500 IFNO<>11ANDNO<>12THENPRINT"YOU'RE RUNNING OUT  
OF IDEAS, AREN'T YOU?":GOTO20  
501 IF(CP=3ORCP=4)ANDOB%(5)<>-1THENPRINT"YOU HAVE  
NO KEY.":GOTO20  
502 IF(CP=3ORCP=4)ANDDO=1THENPRINT"IT ALREADY IS O  
PEN!":GOTO20  
504 IF(CP=3ORCP=4)THENPRINT"YOU OPEN THE DOOR.":DO  
=1:GOTO20  
506 IFNO=12ANDOB%(12)=-1THENPRINT"WELL ... ALL YOU  
R MONEY'S HERE!":GOTO509  
508 PRINT"I DON'T THINK SO, SOMEHOW.":GOTO20  
509 IFCP=1THENE0=1:GOTO3002  
510 PRINT:PRINT"NOW ALL YOU'VE GOT TO DO IS GET BA  
CK TO YOUR FLAT.":GOTO20
```


Explanation

Line 500: if the player is not trying to open the door (no = 11) or the wallet (no = 12) then inform him that he's beginning to get a little bit desperate in his manoeuvres, and back to line 20 again.

Line 501: if the player's in location three or location four, and he's not carrying the key, then he can't open the door, so tell him so and then go to line 20.

Line 502: assuming that the player has got the key, and he's still in locations three or four, and the door is already open, tell him so and go to line 20.

Line 504: still in locations three and four, a successful door-opening operation, set the variable DO to be equal to one, and off to 20.

Line 506: if he's opening the wallet, and is actually carrying it, print a suitable message and go to line 509.

Line 508: standard response to all other opening suggestions.

Line 509: if the player's in room one, he's opened the wallet successfully, then that's the end of the game, so set the variable EO to be equal to one, and go to line 3002.

Line 510: having opened the wallet in a room other than room one, tell the player that he's got to get back to the flat to finish the game. Actually, the bedroom of the flat.

Thinking about it, lines 501 and 502 should be renumbered to be 502 and 503 respectively, line 506 should be renumbered to be 501, and the original line 506 removed. Why? Well, think about it. What if the player is in location three or four and tries to open the wallet?

Something to watch out for in your own adventures.

CLOSE

Naturally enough, if you can open things you should also be able to close them.

But, as usual, intrepid adventurers always try to close things that you don't want them to try to close, so we have to insert various checks into the program to stop strange things happening.

```
530 IFNO<>11THEN500
532 IF (CP=3ORCP=4)ANDDO=0THENPRINT"IT ALREADY IS C
LOSED!";GOTO20
534 IF (CP=3ORCP=4)THENPRINT"YOU CLOSE THE DOOR.";D
O=0;GOTO20
536 PRINT"I DON'T THINK SO, SOMEHOW.";GOTO20
```

Explanation

Line 530: if the player isn't trying to close the door (no = 11), then go to line 500 and print out a suitably sarcastic message.

Line 532: if the player is in locations three or four, and the variable DO is equal to zero, then this means that the door is already shut, so print out a suitable message before going to line 20.

Line 534: still in locations three or four, assuming now that the variable DO is equal to one, the player can close the door. So, print out a message, set the variable DO to equal zero, and off to line 20.

Line 536: standard response to all other closing suggestions.

CROSS

In a lot of adventures, people try to cross ravines or gorges, but in this one the only thing they can attempt to cross is a road outside their flat at the first few locations in the game.

This will not, of course, stop people trying to cross other things. Perhaps some players want to start up a strange sort of breeding colony?

```
560 IF (CP<4ORCP>10) THEN PRINT "THERE'S NOTHING TO CR  
OSS HERE.": GOTO 20  
562 IF NO<>13 THEN PRINT "BIZARRE IDEA!": GOTO 20  
564 EO=0: PRINT "YOU TRY TO CROSS THE ROAD. BUT GET  
CRUSHED UNDER A CAR!": GOTO 3002
```

Explanation

Line 560: if the player's current location (the variable CP) is less than location four or greater than location ten, then tell the player that there's nothing to cross before going back to line 20.

Line 562: if the player is in the correct range of location numbers, but attempts to cross anything other than the road (no = 13) then tell the player that he has a warped mind before going to line 20.

Line 564: all the room descriptions that mention the road say that it's a very busy one, so if someone will attempt to cross it the inevitable happens and the player dies. So, set the variable EO and go to line 3002.

EXAMINE

Every adventure game should have an EXAMINE verb in it. This is one of the player's greatest weapons, because a lot of games hide things away, and they can only be found by examining other objects.

Even if you examine something and get a negative response such as 'I see nothing of interest' this is in itself helpful, since it means that the object examined has no magical purpose, is not a treasure, or anything like that.

```
590 IFCP=1ANDNO=23ANDOB%(5)=0THENPRINTSO$:OB%(5)=C
P:GOTO20
591 IFCP=2ANDNO=25ANDOB%(24)=0THENOB%(24)=CP:PRINT
SO$:GOTO20
592 IFOB%(NO)<>-1THENPRINT"YOU FIND NOTHING OF INT
EREST.":GOTO20
594 IFNO=10THENPRINT"IT'S A STEVE HILLAGE ALBUM.":
GOTO20
596 IFNO=12THENPRINT"#UH ... IMITATION LEATHER.":G
OTO20
598 IFNO=16THENPRINT"A BIT BENT AND WORN. BUT STIL
L USABLE.":GOTO20
600 PRINT"YOU DON'T FIND ANYTHING OF INTEREST.":GO
TO20
```

Explanation

Line 590: if the player is in location one, is examining the flat, and hasn't yet found the key, then print the 'you discover something of interest' message and put the key in the room before going to line 20.

Line 591: if the player is in location two, is examining the room, and hasn't yet come across the disguise which will get him past the policeman, then print our friendly message and put the disguise in the room before going to line 20.

Line 592: if the player isn't carrying an object then he can't examine it, so print out a suitable message before going to line 20.

Line 594: if the player is examining the record, tell him who it's by.

Line 596: examining the wallet?

Line 598: and now examining the cashcard.

Line 600: examining anything else produces nothing of interest, so say so before going to line 20.

FIND and WEAR

Grouped together, because neither of these are particularly long routines.

FIND doesn't, in fact, do very much at all, but people will type in 'FIND thing' if they're getting a bit lost and fed up in a game, so just a few lines of encouraging code to cheer them up are needed.

WEAR is another short piece of code, because (in this game at any rate) the only thing that the player can wear is his disguise. For the sake of decency we must assume that our intrepid adventurer is already clothed before the game started.

```
620 IF OB%(ND)=-1 THEN PRINT "YOU'VE ALREADY GOT IT!":  
GOTO 20  
622 IF OB%(ND)=CP THEN PRINT "IT'S IN FRONT OF YOU.":G  
OTO 20  
624 PRINT "FIND IT YOURSELF.":GOTO 20  
650 IF NO<>24 THEN PRINT "DON'T BE RIDICULOUS.":GOTO 20  
652 IF OB%(24)<>-1 THEN PRINT "YOU HAVEN'T GOT IT.":GO  
TO 20  
654 IF WD=1 THEN PRINT "YOU ALREADY ARE.":GOTO 20  
656 PRINT "OKAY THEN.":WD=1:GOTO 20
```

Explanation

Line 620: if the player is carrying a particular object then we can safely assume that he's already found it, so print out a message and go to line 20.

Line 622: if the object being sought after is in the room, then inform the player of his short-sightedness before going to line 20.

Line 624: curt response if the object isn't being carried and isn't in the room.

Line 650: on to WEAR now, and if the player is trying to wear anything other than his disguise tell him that he's being silly, before going to line 20.

Line 652: if the disguise isn't being carried then it can't be worn, so print a suitable message and off to 20 as usual.

Line 654: if the variable WD is set to one then it means that the disguise is already being worn, so inform the player and go to line 20.

Line 656: you can wear the disguise, so print a message, set the variable WD and go to line 20.

DISCARD and BUY

Another two short routines grouped together, as DISCARD is coped with by another verb, and there is only one thing that we'll allow the player to buy in this particular game.

But, as usual, this doesn't stop people trying to buy other things, and so we have to put the appropriate checks in to make sure that strange things are not allowed to happen.

```
680 GOTO470
710 IF OB%(17)<>-1 THEN PRINT "WHAT WITH?": GOTO20
712 IF CP<>18 THEN PRINT "THERE'S NOTHING TO BUY HERE."
"
714 IF NO<>19 THEN PRINT "WHAT ARE YOU THINKING OF?": G
OTO20
716 IF ZZ>4 THEN PRINT "YOU'RE CARRYING TOO MUCH AND W
ON'T BE ABLE TO HOLD IT.": GOTO20
718 PRINT "OK.": ZZ=ZZ+1: OB%(NO)=-1: GOTO20
```


Explanation

Line 680: go to line 470 and let the verb DROP take care of everything.

Line 710: if the player hasn't got any money (ob%(17) represents the money), tell him that he can't buy anything and go to line 20.

Line 712: if the player isn't in location eighteen then there's nothing to buy, so tell him so and go to line 20 as usual.

Line 714: if the player isn't trying to buy an ice-cream (no = 19) then there are strange things afoot, so tell the player that you're not impressed by his thinking abilities before going to line 20.

Line 716: if you're carrying more than four objects then you won't be able to hold on to the ice-cream anyway, so a suitable message before retreating to line 20.

Line 718: increase the object counter, put the ice-cream in the player's possession, and go to line 20.

READ, TAKE and PICK

Three verbs at a time now! Actually, only the first one serves any sort of useful purpose, since the other two are only in there as synonyms for the verb GET, and the code for that word is used to handle everything. Still, some people prefer using TAKE to GET, so the appropriate jump has to be inserted.

```
740 IFCP<11ORCP>15THENPRINT"THERE'S NOTHING TO REA  
D HERE.":GOTO20  
742 IFNO<>26THENPRINT"YOU'RE IN A LIBRARY ... WHAT  
DO PEOPLE READ IN LIBRARIES?":GOTO20  
744 IFCP<>15THENPRINT"#MMM: SOMETHING ABOUT WRITIN  
G ADVENTURE GAMES.":GOTO20  
745 IFBD=1THENPRINT"YOU CAN'T, THE LIBRARIAN DOESN  
'T TRUST YOU.":GOTO20  
746 IFRB=1THENPRINT"THE BOOK DISINTEGRATES IN YOUR  
HANDS.":BD=1:GOTO20  
747 RB=1:PRINT"ON THE FLY LEAF IT SAYS 'MAGIC WORD  
IS AWAY'.":GOTO20  
770 GOTO440  
800 GOTO440
```

Explanation

Line 740: if the player's current location is less than eleven or greater than fifteen, i.e. he is outside the library, then tell him that there's nothing to read before going to line 20.

Line 742: the player is in the library but is attempting to read something other than a book, so print out a suitable message before going to line 20.

Line 744: reading in any location other than location fifteen produces a plug for the publishers, before going to line 20.

Line 745: if the variable BD has been set then you can't read anything because you can't be trusted.

Line 746: if the variable RB has been set then it means that you've already read a book, so just to be nasty we make it disintegrate and set the variable BD (see line 745).

Line 747: set the variable RB (see line 746), tell the player what the magic word is, and go to line 20.

Line 770: handle TAKE by using the routine for GET starting at line 440.

Line 800: ditto for PICK.

INSERT

Now here's a potentially dangerous verb, and rigorous checks have to be carried out to ensure that the player types in the correct thing, and only the correct thing.

It is obviously meant to be used in conjunction with the cashcard and the bank in an attempt to get money out. But, as usual, some players have warped minds ...

```
830 IFCP<>9THENPRINT"DON'T BE SILLY. JUST GET ON  
WITH IT.":GOTO20  
831 IFNO<>16THENPRINT"NO, NO, NO, NO, NO!":GOTO20  
832 IFAG=1THENPRINT"YOU CAN ONLY USE YOUR CARD ONCE  
PER GAME.":GOTO20  
834 IFOB%(16)<>-1THENPRINT"WHERE'S YOUR CARD, THEN  
?":GOTO20  
835 PRINT"OKAY, WHAT'S YOUR CODE NUMBER (FOUR  
DIGITS)?":PRINT:GOSUB60000  
836 PRINT:A=0:FORI=1TOLEN(CM%):A=A+ASC(MID$(CM%,I,  
1)):NEXT:IFA=215THEN840  
837 NG=NG+1:IFNG=3THEN842  
838 PRINT"WRONG, TRY AGAIN."  
839 PRINT:GOSUB60000:GOTO836  
840 AG=1:PRINT"LO AND BEHOLD, SOME MONEY POPS OUT."  
:OB%(17)=-1:ZZ=ZZ+1:GOTO20  
842 PRINT"RIGHT, THAT'S IT, YOU'VE HAD THREE GOES,  
SO I'M TAKING YOUR CARD AWAY."  
843 OB%(16)=0:ZZ=ZZ-1:GOTO20
```

Explanation

Line 830: if the player isn't in location nine then the program will have nothing to do with whatever he's typed in, so print out a message and go to line 20.

Line 831: even if he is in location nine, if the noun isn't the sixteenth one, namely the card, then we tell the player off and retreat to line 20.

Line 832: if the variable AG has been set this means that the card has been used, and as we all know you can only use cards once a day.

Line 834: if the player isn't carrying the card, suggest that he goes and gets it, before the program goes to line 20.

Line 835: now for the tricky part, remembering your code number.

Line 836: take the four digit number typed in, take the ASCII value of each number typed in in turn, and add all four of them up. If the total arrived at is 215 then go to line 840.

Line 837: increment the variable NG by one, and if it equals three go to line 842.

Line 838: simple message.

Line 839: back we go for another try.

Line 840: success, so set the variable AG, print out a message, give the player some money, increment the object counter and go to line 20.

Line 842: if the player takes three goes to guess his number and *still* gets it wrong, then the machine swallows his card. How sadly true to life.

Line 843: remove the card from the player, decrement the object counter and go to line 20 as if nothing's happened.

CHASE

A forlorn hope really, but some people do think that they can go chasing about after things.

```
850 PRINT"AARGH. YOUR ICE-CREAM'S MELTED!":OB%(19)  
=0:ZZ=ZZ-1  
852 FORI=1TO2000:NEXT:GOTO393  
860 PRINT"YOU'RE TOO OLD TO CHASE ANYTHING.":GOTO2  
0
```

Explanation

Actually lines 850-852 having nothing to do with this verb at all, but I needed some space in the program somewhere, and this seemed to fit the bill.

Line 850: this is what happens if the player carries his ice-cream about for too long, so print a message, remove the ice-cream and decrement the object counter.

Line 852: wait for a while before going back to where we came from, namely line 393.

Line 860: aha! The chase routine, which simply prints out a message before going back to line 20.

LOAD and SAVE

Essential to any adventure game, these two verbs allow you to record, and later recall, your progress at any time.

```
890 PRINT:PRINT"INSERT TAPE AND PRESS SPACE WHEN R
EADY."
891 FORI=1TO10:GETSP$:NEXT
892 GETSP$:IFSP$<>" "THEN892
894 OPEN1,1,0,"DATA"
896 FORI=1TONN:INPUT#1,OB%(I):NEXT
898 INPUT#1,WD,EO,WB,CL,DO,AG,NG,MI,RB,BD,CP
900 CLOSE1:GOTO15
920 PRINT:PRINT"INSERT TAPE AND PRESS SPACE WHEN R
EADY."
921 FORI=1TO10:GETSP$:NEXT
922 GETSP$:IFSP$<>" "THEN922
923 Z$=CHR$(13)
924 OPEN1,1,1,"DATA"
926 FORI=1TONN:PRINT#1,OB%(I);Z$;:NEXT
928 PRINT#1,WD;Z$;EO;Z$;WB;Z$;CL;Z$;DO;Z$;AG;Z$;NG
;Z$;MI;Z$;RB;Z$;BD;Z$;CP;Z$;
930 CLOSE1:GOTO20
```

Explanation

Lines 890-892: print a message and wait for the space bar to be pressed.

Line 894: open a file called DATA.

Line 896: load the position of all the objects.

Line 898: load the all-important list of variables.

Line 900: close the file and go to line 15.

Lines 920-922: print a message and wait for the space bar to be pressed.

Line 923: set the variable Z\$ to equal a carriage return.

Line 924: open a file called DATA.

Line 926: save the position of all the objects.

Line 928: save the all-important list of variables.

Line 930: close the file and go to line 20.

TALK

Why people should try to talk I've no idea, but they do, they do, and so a few lines of code are needed to cope with this particular suggestion from the player.

Actually, in our second game ('Enormous Cave') talking plays a very useful role, and without talking to someone in particular you'll never be able to finish the game. However, in this one it fulfils no useful purpose whatsoever.

```
950 IFCP=4ANDNO=20THENPRINT"STOP WASTING POLICE TI  
ME. ":GOTO20  
952 IF (CP>10ANDCP<16)ANDNO=14THENPRINT"YOU GET A S  
TERN LOOK. ":GOTO20  
954 IFCP=25AND(NO=21ORNO=22)THEN958  
956 PRINT"ARE YOU MAD? JUST GET ON WITH THE  
GAME. ":GOTO20  
958 PRINT"HE SAYS 'MY MOTHER TOLD ME NEVER TO TALK  
TO STRANGERS', AND RUSHES OFF."  
960 GOTO20
```

Explanation

Line 950: talking to the policeman in location four earns the player a rebuke before going to line 20.

Line 952: talking to the librarian in the library gives the player a similar fate.

Line 954: talking the child, or kid, in location 25 takes the player to line 958.

Line 956: sarcastic message before going to line 20.

Line 958: words of wisdom from one so young.

Line 960: off to line 20 as usual.

GIVE, OFFER and SEARCH

Since GIVE and OFFER are essentially the same thing (well, as far as this game is concerned anyway), the same piece of code is used to handle both verbs.

Again for the purpose of this game, SEARCH is taken to be the same as EXAMINE, and so we just need a program jump in case some people prefer using one word to the other.

```
980 IF OB%(NO)<>-1 THEN PRINT "YOU HAVEN'T GOT IT.": GOTO 20
982 IF CP<>25 THEN PRINT "NO-ONE WANTS IT.": GOTO 20
984 IF NO<>19 THEN PRINT "NO-ONE'S INTERESTED.": GOTO 20
986 PRINT "THE KID ACCEPTS THE ICE-CREAM AND THROWS YOUR WALLET AT YOU."
988 OB%(19)=0: OB%(12)=-1: GOTO 20
1010 GOTO 980
1040 GOTO 590
```

Explanation

Line 980: a player can't offer something he hasn't got.

Line 982: offering anything anywhere other than location 25 is fruitless, so print out message and off to line 20.

Line 984: if the object being offered isn't an ice-cream then no one's interested, so again print out a message and off to line 20.

Line 986: acknowledging the bribe, the kid returns the player's wallet.

Line 988: so we remove the ice-cream from the player's hands and replace it with the wallet before going to line 20.

Line 1010: OFFER is taken to mean GIVE, so go to line 980.

Line 1040: SEARCH is taken to mean EXAMINE, so go to line 590.

LISTEN, ENTER and AWAY

These are last three verbs. The first one serves no purpose other than to satisfy those people who want to type it in. The second is genuinely useful, but can lead you astray if you're not careful, while the third verb is a magic word, which, under certain conditions, causes wondrous things to happen.

```
1070 IFCP=2ANDNO=10THENPRINT"SOUNDS LIKE STEVE HIL  
LAGE TO ME.":GOTO20  
1072 PRINT"YOU HEAR THE USUAL NOISES OF THE DAY.":  
GOTO20  
1100 IFCP=11ANDCL=0ANDNO=14THENPRINT"WITHOUT A LIB  
RARY CARD?":GOTO20  
1101 IFCP=11ANDCL=1ANDNO=14THENC=12:GOTO3000  
1102 IFCP=9ANDNO=15THENE=1:PRINT"OH NO, THOSE WAN  
TED POSTERS ...":GOTO3002  
1104 GOTO508  
1130 IF (C=15ORC=16)ANDBD<>1THENC=31-C:GOTO15  
1132 PRINT"NOTHING HAPPENS.":GOTO20
```

Explanation

Line 1070: listening to the record in location two tells the player who it's by.

Line 1072: listening anywhere else produces nothing very interesting.

Line 1100: if the player is in location eleven and trying to enter the library without a library card, throw him out on his ear and go to line 20.

Line 1102: entering the bank, as we all know, can sometimes be fatal, so set the end of game variable EO before printing out a message and going to line 3002.

Line 1104: go to line 508 and print out a message.

Line 1130: if the player is in location fifteen or sixteen and hasn't yet destroyed a book, then magically transport him.

Line 1132: otherwise just print out a message.

Conclusion

We have now completed 'The Lost Wallet'. Short and sweet it may be, but it should give you a good idea of how various Basic techniques can be implemented.

Now the next game is a different matter altogether, being much longer, much more complex, and requiring a disk drive. It also has a number of interesting characters who roam about the caves in which the game is set, as well as over ten times as many locations as the game in this chapter.

However, it is going to be presented with very little explanation, so if you begin floundering somewhere you'll come to love this chapter. You should be able to sort things out just by looking at this listing (and the rest of it in Chapter 5 of course) and reading the appropriate pages.

9

A Disk-based Adventure

Introduction

In the previous chapter we concerned ourselves with a small-scale adventure, going through the listing more or less line by line, and seeing how the whole thing was put together. That was a tape-only game, so in this chapter we'll graduate to using disks and present the program 'Enormous Cave'.

The introduction, as well as having a title-page display, tells you all about the program.

```
0 POKE53281,7:POKE53280,9:PRINT"[CLR,BLK]":PRINTCHR$(8):PRINTCHR$(142)
1 P$="ADVENTURE SOFTWARE  ADVENTURE SOFTWARE  "
2 PRINT"[HOME,SCD,CR]"LEFT$(P$,38)
3 PRINT"[12CD,CR]"MID$(P$,14);LEFT$(P$,12)
4 PRINT"[HOME]":FORA=1TO23:PRINT"[6CR]"MID$(P$,A,1):NEXT
5 PRINT"[HOME]":FORA=1TO23:PRINT"[26CR]";
6 PRINTMID$(P$,A,1):NEXT:T$=LEFT$(P$,38):P$=RIGHT$(P$,1):P$=P$+T$
7 PRINT"[HOME,10CD]"TAB(11)"[RVS]PRESS ANY  ":PRINTTAB(11)"[RVS]KEY FOR MORE
8 GETK$:IFK$=""THEN2
10 PRINTCHR$(8):POKE53281,7:POKE53280,9:PRINTCHR$(14)
12 PRINT"[CLR,BLK]WELCOME TO ENORMOUS CAVE!"
20 PRINT"[2CD]"THIS STARTED OFF LIFE AS AN EXERCISE
   IN
22 PRINT"USING THE COMMODORE 1541 DISK DRIVE.
24 PRINT"[CD]"I DECIDED TO WRITE AN ADVENTURE GAME,
26 PRINT"USING PLENTY OF DISK FILES, AND THOUGHT
28 PRINT"THAT A VERSION OF THE ORIGINAL COLOSSAL
30 PRINT"CAVE WOULD BE NICE.
32 PRINT"[CD]"HOWEVER, EVERYONE (?) KNOWS THAT GAME
.
```

```

34 PRINT"AND SO THIS 'SON OF COLOSSAL CAVE' WAS B
ORN.
36 PRINT"[CD]70 ANYONE WHO HAS PLAYED THE ORIGINAL
,
38 PRINT"IT SHOULD BE FAMILIAR ENOUGH IN PLACES.
40 PRINT"HOWEVER, MY INTEREST IN ADVENTURE GAMES
42 PRINT"WAS SUCH THAT THE STORYLINE ENDED UP
44 PRINT"BEING A LONG WAY FROM THE ORIGINAL.
46 GOSUB2000
50 PRINT"[CLR]PLENTY OF OLD FRIENDS ARE THERE.
52 PRINT"INCLUDING THE DWARF, THE BIRD, THE TROLL
54 PRINT"[CUJ]AND THE BRIDGE, AND SO ON. THERE ARE
56 PRINT"ALSO PLENTY OF NEW FRIENDS TO BE MADE,
58 PRINT"AND BEWARE, HARDLY ANYTHING YOU'RE USED
60 PRINT"TO DOING WILL WORK ANYMORE.
62 PRINT"[CD]YOU CAN USE THE FAMILIAR VERB/NOUN TY
PE
64 PRINT"OF ENTRY IF YOU WANT, ALTHOUGH YOU CAN
66 PRINT"GO A LITTLE BIT FURTHER THAN THAT IF N
ECESSARY."
68 PRINT"[CD]YOU'LL SOON FIND A COMPANION IN THE
70 PRINT"CAVES, AND IF YOU WANT TO TALK TO HIM,
72 PRINT"ENTER THE COMMAND 'TALK TO EDDY'.
74 PRINT"[CD]THEN, JUST TYPE IN YOUR REQUEST.
76 GOSUB2000
80 PRINT"[CLR]ONLY REMOVE THE MASTER DISK FROM THE
82 PRINT"DRIVE WHEN YOU WANT TO SAVE YOUR
84 PRINT"PROGRESS TO DISK. INSERT YOUR OWN,
86 PRINT"FORMATTED DISK INSTEAD, AND THEN PUT THE
88 PRINT"[CUJ]ORIGINAL BACK IN THE DRIVE.
96 PRINT"[CD]AS IN THE ORIGINAL, YOUR GOAL IS TO
98 PRINT"COLLECT THE TREASURES AND GET OUT ALIVE.
100 PRINT"[CUJ]WHETHER YOU SUCCEED IS, OF COURSE, U
P TO
102 PRINT"[CUJ]YOU (AND ANYONE ELSE WHO MIGHT BE LI
VING
104 PRINT"[CUJ]IN THE CAVES). "
106 GOSUB2000
108 PRINT"[CLR]DON'T EXPECT AN INFOCOM-TYPE GAME.
ONE
110 PRINT"PROGRAMMER WORKING ALONE FOR THREE
112 PRINT"WEEKS WILL NOT CREATE WHAT A TEAM OF
114 PRINT"PROGRAMMERS CAN DO OVER MANY MONTHS.
116 PRINT"BESIDES, THE CATS KEPT WANTING FEEDING
118 PRINT"AND THE CRICKET WAS ON TELE.
120 PRINT"[CD]STILL. I HOPE YOU ENJOY IT. THERE A
RE
122 PRINT"ABOUT 250 LOCATIONS TO FIND, AND THE
124 PRINT"PROGRAM UNDERSTANDS AROUND 200 DIFFERENT
126 PRINT"[CUJ]VERBS AND NOUNS.
127 PRINT"[CD]FINALLY, LONG GIVES YOU LONG ROOM

```

```

128 PRINT"DESCRIPTIONS, AND SHORT GIVES YOU, WELL,
SHORT ONES.
129 PRINT"[CD]REDESCRIBE LETS YOU LOOK AT A LOCATI
ON DESCRIPTION AGAIN.
130 PRINT"[CD]SO, PRESS 'L' TO LOAD THE MAIN GAME.
OR 'R' TO RE-READ THIS LOT.
131 GETA$: IFA$="" THEN131
132 IFA$="R" THENRUN
134 IFA$<"L" THEN131
136 PRINT"[CLR,YEL,2CD]NEW":PRINT"[2CD]LOAD"CHR$(3
4)"ADVENTURE"CHR$(34)",8[HOME]";
138 POKE198,6:POKE631,13:POKE632,13:POKE633,82:POK
E634,85:POKE635,78:POKE636,13:END
2000 PRINT"[CD,RVS,BLK]PRESS SPACE TO CONTINUE.
2002 FORI=1TO10:GETSP$:NEXT
2004 GETSP$: IFSP$<>" " THEN2004
2006 RETURN

```

READY.

If, like me, your fingers waver at the sight of a listing that has to be typed in, you'll be relieved to know that you don't *have* to type in this one. It's purpose is just to tell you about the game and then load the main program, so if you can't be bothered to do some typing then just keep these pages handy and read through them before starting to play the adventure proper.

The main listing

This, as you can see, covers a number of pages, but has very few messages sprinkled about among the code. This is quite all right because, being a disk game, the vast majority of messages and *all* the room descriptions are stored on disk and fetched in and displayed as and when they are required.

As this program follows much the same format as the previous one, but on a much larger scale, there is no need to go through the listing line by line. However, a few pertinent sections of the code ought to be examined.

```

0 B=14:C=17:A=0:GOSUB50:POKE53281.0:POKE53280.0
4 DATA162.6,32.198.255.32.228.255.201.42.208.249.3
2.228.255.205.155.3,208
5 DATA241.76,204,255,0,0.162.6,32.198,255,32.228.2
55.76,204,255
6 FORI=900TO935:READA:POKEI.A:NEXT
7 DATA104.104,32.253,174,32.235.183.76.166.168

```

```

8 PRINTCHR$(14):CHR$(8):PRINT"[CLR.YEL]WELCOME ";
9 OPEN2,8,2,"O:MAP1,S.R":DIMP%(253.6):A=38:D=126:G
OSUB50
10 FORI=1TO125:FORJ=0TO5
11 INPUT#2,P%(I,J)
12 NEXTJ,I:CLOSE2:PRINT"TO ":A=43:D=52:GOSUB50
13 OPEN2,8,2,"O:MAP2,S.R"
14 FORI=126TO253:FORJ=0TO5
15 INPUT#2,P%(I,J)
16 NEXTJ,I:CLOSE2:PRINT"ENORMOUS ":A=34:D=75:GOSU
B50
17 DO$="[RVS.BRN]YOU NEED A DIRECT OBJECT."
18 P%(63.3)=64
19 FORI=49152TO49162:READA:POKEI,A:NEXT
20 DIMP$(253)
21 OPEN2,8,2,"O:SHORT,S.R"
22 FORI=1TO253:INPUT#2,P$(I)
23 NEXT
24 CLOSE2:PRINT"CAVE":A=17:D=37:GOSUB50
26 REM
28 NN=107:NV=84
29 DIMOB$(NN),OB%(NN,1),VB$(NV),NO$(NN),D$(5)
30 OPEN2,8,2,"O:STUFF,S.R"
31 P%(114,0)=114:P%(114,1)=114:P%(17,4)=16
32 FORI=0TO5:INPUT#2,D$(I):NEXT
34 FORI=1TONV:INPUT#2,VB$(I):NEXT
36 FORI=1TONN:INPUT#2,NO$(I):NEXT
38 FORI=1TONN:INPUT#2,OB$(I):NEXT
40 FORI=1TONN:INPUT#2,OB%(I,0):NEXT:OB%(6,0)=0
42 FORI=1TONN:INPUT#2,OB%(I,1):NEXT:CLOSE2
43 OB%(68,0)=115:P%(248,0)=249:P%(232,0)=0:P%(232,
3)=233
44 PRINT"!":A=25:D=177:GOSUB50:FORI=1TO3000:NEXT:P
%(120,0)=123
45 P$(36)="YOU ARE IN LOW MISTY TUNNEL.":P$(43)="Y
OU ARE NORTH OF GAPING CRACK.
46 P%(76,1)=77:P%(76,2)=75:P%(144,3)=0:P%(126,1)=1
27
47 P$(76)="YOU ARE IN CORNER OF DARK ROOM."
48 PRINT"[CLR]":PRINTCHR$(142):GOSUB60
49 P$(126)="YOU ARE SOUTH OF OLD DOOR.":P$(125)="Y
OU ARE NEXT TO OLD DOOR.":GOTO100
50 V=54272:POKEV+24,15:POKEV+5,B:POKEV+6,0:POKEV+4
,0:POKEV+4,C
51 POKEV+1,A:POKEV,D
52 RETURN
60 OB$(69)="A DRINK"
83 P$(199)="YOU ARE IN COLD ROOM.":P$(204)="YOU AR
E ON LEDGE NEAR LAKE."
84 POKE53280,0:POKE53281,0:RETURN
100 PRINT"[CLR]":PRINTCHR$(14):
101 B$="COL3":F2=0:OPEN6,8,6,"O:"+B$+".S.R":CP=1:G

```

```

0SUB4000
200 REM
201 GOTO300
202 PRINT"[RVS.BLU]YOU CAN SEE:[ORNG]
203 A=0:FORI=1TONN:IFOB%(I,0)=CPTHENPRINTOB$(I):A=
A+1
204 NEXT:IFA=0THENPRINT"NOTHING MUCH."
206 PRINT"[CD.RVS.RED]OBVIOUS EXITS ARE:[ORNG]
208 A=0:FORI=0TO5:IFP%(CP,I)<>0THENPRINTD$(I)" "":
A=A+1
209 NEXT:IFA=0THENPRINT"NOT VERY OBVIOUS."
300 IFWB=20THENWB=0:R=INT(RND(.5)*4)+253:GOSUB4000
301 GOSUB500:IFVB=83ORVB=84ORVB=49THEN305
302 IFNO$<>""ANDVBANDNO=0THENPRINT"[BRN]I DON'T KN
OW WHAT YOU'RE SAYING.":WB=WB+1:GOTO300
303 IFNO=0AND(VB<10ORVB>20)ANDVB<>32ANDVB<>33ANDVB<
>34ANDVB<>35ANDVB<>82THENPRINTDO$:GOTO300
304 IFVB=0THENPRINT"[BRN]SORRY. YOU'LL HAVE TO RE-
PHRASE THAT.":GOTO300
305 GOSUB3150:IFVB>7ANDVB<14THENVB=VB-6
306 QA=1:GOSUB1080:QA=0:IFA>260ANDCC=0ANDCP>43THEN
CC=1:GOTO3215
308 IFFL>250ANDBA=1ANDBO%(39,0)=-1THENBA=0:R=350:G
OSUB401:OB%(39,0)=0:FL=-2000000
309 IFFL=250THENR=280:GOSUB4000:PRINT:PRINT
310 IFFL=500THENR=281:GOSUB4000:LO=0:OB%(38,0)=0:O
B%(37,0)=-1
311 GOTO3180
312 IFDF=1ANDVB<>29THENR=376:GOSUB4000:GOTO1453
313 GOTO6300
400 GOSUB4000:GOTO300
401 PRINT:PRINT:GOSUB4000:RETURN
402 R=CP-1:PRINT:PRINT:GOTO400
500 PRINT:PRINT"[CD.RVS.PUR]WHAT NOW?[OFF]" "":GOSU
B60000:PRINT:IFLI>0THENLI=LI-1:IFLI=1THEN3600
501 IFLEFT$(CM$,11)="WHO IS EDDY"ANDED=1THENR=257:
GOSUB4000:GOTO500
502 IFLEFT$(CM$,3)="WHY"THENPRINT"[BRN]BECAUSE I S
AY SO!":GOTO500
503 IF(CP>18ANDCP<246)ANDBO%(38,0)<>-1THENLO=0
504 IFCP=221ANDLEFT$(CM$,8)="PAY SUBS"THENPRINT:PR
INT"[BRN]IT'S NOT MONEY HE WANTS.":GOTO500
505 IFLEFT$(CM$,4)="PLUG"THEN3120
506 IFLEFT$(CM$,4)="AUST"THEN3210
507 IFLEFT$(CM$,3)="FEE"ANDLEFT$(CM$,4)<>"FEED"THE
N1530
508 IFLEFT$(CM$,3)="SAY"THEN3330
509 PRINT"[RED]":N1$="":V1$="":NO=0:VB=0:NO$="":V
B$=""
510 CM=LEN(CM$):FORI=1TOCM:IFMID$(CM$,I,1)<>" "THE
NV1$=V1$+MID$(CM$,I,1):NEXTI
511 VB$=LEFT$(V1$,4):FORI=1TONV:IFVB$(I)=VB$THENVB

```



```

=I:GOTO514
512 NEXT I
513 VB=+1:NO$=V1$:GOTO522
514 IFLEN(V1$)+1>LEN(CM$) THENNO=0:RETURN
516 FORI=CMT01STEP-1:IFMID$(CM$.I.1)<>" " THENNO$=N
O$+MID$(CM$.I.1):NEXT
517 Z=4:IFLEN(NO$)=3THENZ=3
518 NO$=RIGHT$(NO$.Z):FORI=1TOZ:A$(I)=MID$(NO$.Z+1
-I.1):NEXT
520 NO$="":FORI=1TOZ:NO$=NO$+A$(I):NEXT
522 FORI=1TONN:IFLEFT$(NO$(I).Z)=NO$ THENNO=I:GOTO5
25
524 NEXT I:RETURN
525 IFNO=34ANDOB%(35.0)=-1 THENNO=35:RETURN
526 IFNO=36ANDOB%(35.0)=-1 THENNO=35:RETURN
527 RETURN
600 IFNO<16ORNO>27 THENPRINT"[RVS.GREY]TRY RE-PHRAS
ING THAT!":GOTO300
602 VB=NO-14:IFVB>7 THENVB=VB-6:GOTO310
630 IFLO=0AND(CP>18ANDCP<246) THENR=258:GOTO400
631 VB=VB-2:IFP%(CP.VB)<>0ANDP%(CP.VB)<>-1 THENCP=P
%(CP.VB):R=CP-1:GOSUB4000:GOTO200
632 IFP%(CP.VB)=-1 THENPRINT"[GREY]THAT'LL TEACH YO
U TO GO THAT WAY ... YOU'RE DEAD!
633 IFP%(CP.VB)=-1 THENFORI=1TO3000:NEXT:PRINT"[CLR
]":PRINT"[RVS.BRN]HO HO HO!":GOTO5000
634 PRINT"[RVS.GREY]YOU CAN'T GO THAT WAY.":GOTO30
0
660 IFCP=101ANDTB=1 THENPRINT:PRINT"[BRN]THE TROLL
WON'T LET YOU.":PRINT:GOTO300
661 GOTO630
690 IFCP=74 THENQA=1:GOSUB1171:IFZZ>0 THENR=354:QA=0
:GOTO400
691 GOTO630
720 IFCP=73 THENQA=1:GOSUB1171:IFZZ>0 THENR=354:QA=0
:GOTO400
725 GOTO630
750 IFCP=17ANDGF=0 THENPRINT:PRINT"[BRN]THE GRATE I
S LOCKED.":GOTO300
751 GOTO630
780 GOTO630
990 GOSUB1016
992 OPEN2.8.2."O:ENORDATA.S,R"
994 FORI=1TONN:INPUT#2,OB%(I.0):NEXT
996 INPUT#2,V,A,B,C,D,CP,R,SS,NV,NN
998 FORI=1TO253:INPUT#2,P%(I.6):NEXT
1000 INPUT#2,P%(16.5),P%(17.4),P%(28.2),P%(28.3),P
%(52.1)
1002 INPUT#2,P%(56.2),P%(101.1),P%(125.1),P%(168.1
),P%(174.1)
1004 INPUT#2,P%(176.2),P%(176.3),P%(176.1),P%(193.
2),P%(221.3)

```

```

1005 INPUT#2,P%(233,3),P%(249,0),P%(249,3)
1006 INPUT#2,PD,CP,WB,ED,EF,S$,QY,GF,PC,BD,CO
1007 INPUT#2,DO,OD,MD,LO,FL,GP,DK,SK,WK,BB,WP
1008 INPUT#2,TB,DW,FC,GC,SB,BD,BA,DD,WR,RD,DF,LC
1009 INPUT#2,SA,CC,L1,BC,PV,MC:CLOSE2:OPEN6,8,6,"O
: "+A$+",S,R"
1010 PRINT:IFCP>18THENPOKE53281,0:POKE53280,0
1011 R=CP-1:PRINT"[BRN]":GOSUB4002:PRINT:PRINT:GO
T0202
1016 PRINT:PRINT"[RVS. DRNG]PRESS 'SPACE' WHEN READ
Y TO CONTINUE."
1017 FORI=1TO10:GETSP$:NEXT
1018 GETSP$:IFSP$<>" "THEN1018
1019 CLOSE6:PRINT"[CD.BRN]OK ... HANG ON ...":RETU
RN
1020 GOSUB1016
1021 OPEN1,8,15,"SO:ENORDATA":CLOSE1
1022 OPEN2,8,2,"O:ENORDATA,S,W":Z$=CHR$(13)
1024 FORI=1TONN:PRINT#2,OB%(I,0):Z$::NEXT
1026 PRINT#2,V:Z$:A:Z$:B:Z$:C:Z$:D:Z$:CP:Z$:R:Z$:S
S:Z$:NV:Z$:NN:Z$:
1028 FORI=1TO253:PRINT#2,P%(I,6):Z$::NEXT
1030 PRINT#2,P%(16,5):Z$:P%(17,4):Z$:P%(28,2):Z$:P
%(28,3):Z$:P%(52,1):Z$:
1032 PRINT#2,P%(56,2):Z$:P%(101,1):Z$:P%(125,1):Z$
:P%(168,1):Z$:P%(174,1):Z$:
1034 PRINT#2,P%(176,2):Z$:P%(176,3):Z$:P%(176,1):Z
$:P%(193,2):Z$:P%(221,3):Z$:
1035 PRINT#2,P%(233,3):Z$:P%(249,0):Z$:P%(249,3):Z
$:
1036 PRINT#2,PD:Z$:CP:Z$:WB:Z$:ED:Z$:EF:Z$:S$:Z$:Q
Y:Z$:GF:Z$:PC:Z$:BD:Z$:CO:Z$:
1037 PRINT#2,DO:Z$:OD:Z$:MD:Z$:LO:Z$:FL:Z$:GP:Z$:D
K:Z$:SK:Z$:WK:Z$:BB:Z$:WP:Z$:
1038 PRINT#2,TB:Z$:DW:Z$:FC:Z$:GC:Z$:SB:Z$:BD:Z$:B
A:Z$:DD:Z$:WR:Z$:RD:Z$:DF:Z$:
1039 PRINT#2,SA:Z$:CC:Z$:L1:Z$:BC:Z$:PV:Z$:MC:Z$:L
C:Z$:
1040 CLOSE2:OPEN6,8,6,"O: "+A$+",S,R"
1048 CLOSE2:IFQY=1THEN5000
1049 GOTO300
1050 R=334:GOTO400
1080 A=0:FORI=1TONN:IFOB%(I,0)=2THENA=A+OB%(I,1)
1081 NEXT:IFQA=1THENRETURN
1082 PRINT"[RVS.BRN]YOU HAVE SCORED":A:"[CL] OUT O
F 300."
1083 PRINT:S$="USELESS ADVENTURER.":IFA>30THENS$="
BEGINNER."
1084 IFA>60THENS$="PERSON WHO'S GETTING BETTER."
1085 IFA>90THENS$="SEASONED ADVENTURER."
1086 IFA>120THENS$="PERSISTENT ADVENTURER."
1087 IFA>150THENS$="PERSON WHO'S MORE THAN HALFWAY

```

```

THERE."
1088 IFA>180THENS$="WISE OLD SOUL."
1089 IFA>210THENS$="VERY WISE OLD SOUL."
1090 IFA>240THENS$="MEAN WIZARD!"
1091 IFA>270THENS$="MASTER ADVENTURER."
1092 PRINT"IRVS.BRNJTHIS GIVES YOU A PERSONAL RATI
NG AS A":PRINT"IRVS]"S$
1093 IFQY=1THENRETURN
1095 GOTO300
1110 IFLO=0AND(CP>19ANDCP<246) THENR=259:GOSUB4000:
GOTO300
1111 R=0:GOTO202
1140 R=CP-1:P%(CP,6)=0:GOSUB4000:GOTO300
1170 PRINT"IRVS.BRNJYOU ARE CARRYING:
1171 ZZ=0:FORI=1TONN:IFOB%(I,0)=-1THENZZ=ZZ+1:IFQA
=0THENPRINTOB$(I)
1172 NEXT:IFQA=1THENRETURN
1173 IFZZ=0THENPRINT"NOTHING."
1174 GOTO300
1200 IFNO=48ORNO=50ORNO=57ORNO=91ORNO=83THENR=345:
GOTO400
1202 R=346:GOTO400
1230 PRINT"IRVS.GREYJFIND A TOILET ...":GOTO300
1260 PRINT"IRVS.GREYJWATCH IT ...":GOTO300
1290 PRINT"IRVS.GREYJ I BEG YOUR PARDON ?!":GOTO300
1320 PRINT"IRVS.GREYJ TUT TUT TUT ...":GOTO300
1350 IFOB%(NO,0)<>-1ANDOB%(NO,0)<>CPTHENPRINT:PRIN
T"IRVS.GREYJWHERE IS IT?":GOTO300
1351 IFNO=33ANDOB%(NO,0)=-1THENR=335:OB%(33,0)=0:Z
Z=ZZ-1:GOTO400
1352 IFNO=42THENR=336:GOTO400
1353 IFNO=63THENR=337:GOTO400
1354 IFNO=29THENR=338:GOTO400
1355 R=339:GOTO400
1380 IFNO<>36ANDNO<>69THENR=340:GOTO400
1381 IFOB%(35,0)=-1THENR=341:OB%(35,0)=0:OB%(34,0)
=-1:GOTO400
1382 IFOB%(69,0)=-1THENR=347:OB%(69,0)=0:ZZ=ZZ-1:G
OTO400
1383 PRINT:PRINT"IRVS.GREYJYOU CAN'T.":GOTO300
1410 IFNO=29ANDEF=1THENR=342:GOTO400
1411 IFOB%(NO,0)<>-1OROB%(NO,0)<>CPTHENPRINT:PRINT
"IRVS.GREYJWHERE IS IT?":GOTO300
1412 IFNO=42THENR=343:GOTO400
1414 R=344:GOTO400
1440 IFOB%(NO,0)<>-1THENPRINT:PRINT"IRVS.GREYJYOU
HAVEN'T GOT THAT.":GOTO300
1441 IFDFTHEN1450
1442 IFCP=233ANDNO=55THENR=371:OB%(NO,0)=CP:ZZ=ZZ-
1:GOTO400
1443 IFCP=233ANDNO=75THENR=370:OB%(NO,0)=0:ZZ=ZZ-1
:P%(233,3)=234:TB=0:GOTO400

```

```

1444 IFCP=233THENR=374:OB%(NO,0)=CP:ZZ=ZZ-1:GOTO40
0
1445 IFCP=101ANDNO=55THENR=372:OB%(NO,0)=136:P%(10
1,1)=102:TB=0:ZZ=ZZ-1:GOTO400
1446 IFCP=101ANDOB%(NO,1)>0THENR=373:OB%(NO,0)=0:Z
Z=ZZ-1:P%(101,1)=102:GOTO400
1447 IFCP=101THENR=374:OB%(NO,0)=CP:ZZ=ZZ-1:GOTO40
0
1448 GOTO1860
1450 IFNO<>55THENR=375:OB%(NO,0)=CP:GOSUB4000:R=37
6:GOSUB4000:GOTO1453
1451 IFINT(RND(.5)*10)>2THENR=377:OB%(55,0)=CP:OB%
(61,0)=0:DF=0:GOTO400
1452 PRINT"[CD,ORNG]#ISSUED!":PRINT:R=376:GOSUB4000
1453 IFINT(RND(.5)*10)>9THENR=379:GOSUB4000:GOTO50
00
1454 PRINT:PRINT"[CD,RVS,ORNG]IT MISSED!":PRINT:GO
TO313
1470 IFEF<>1THENR=397:GOTO400
1471 IFNO<>29THENR=398:GOTO400
1472 R=399:GOSUB4000:PRINT:PRINT
1473 GOSUB60000:PRINT
1474 IFCM$="ATTACK DRAGON"ANDCP=168ANDP%(168,1)=0T
HENR=400:DK=1:P%(168,1)=169:GOTO400
1475 IFLEFT$(CM$,5)="ATTAC"THENR=401:GOTO400
1476 IFCP=52ANDP%(52,1)=0ANDCM$="PUSH ROCK"THENP%(
52,1)=53:R=402:BB=1:GOTO400
1477 IFCP=52ANDP%(52,1)=0ANDCM$="MOVE ROCK"THENP%(
52,1)=53:R=402:BB=1:GOTO400
1478 IFCP=243ANDSA=0ANDLEFT$(CM$,4)="SWIM"THENR=40
3:SA=1:EF=0:GOTO400
1479 IFCM$="GET LOST"THENEF=0:R=404:OB%(29,0)=CP+1
:GOTO400
1480 IFLEFT$(CM$,5)="CARRY"THENPRINT:PRINT"[BRN]'L
IKE, I'M NOT A DONKEY. MAN'.":GOTO300
1481 GOSUB508:IF(VB>21ANDVB<26)THENPRINT:PRINT"DON
'T GET ALL UN-COOL AND HEAVY. MAN.":GOTO300
1490 R=405:GOTO400
1500 GOTO1470
1530 IFCP=159ANDOB%(2,0)=0THENR=349:OB%(2,0)=CP:GO
TO400
1531 IFCP<52THENPRINT:PRINT"[RVS,GREY]NOTHING HAPP
ENS.":GOTO300
1532 R=351:GOSUB401:GOTO5000
1560 IFCP=199ANDCC=0THENC=2:GOTO402
1562 IFCP=2ANDCC=0THENC=199:GOTO402
1563 GOTO1592
1590 IFCP=150THENR=352:GOSUB401:CP=151:GOTO402
1591 IFCP=166THENR=352:GOSUB401:CP=144:GOTO402
1592 PRINT:PRINT"[BRN]NOTHING HAPPENS.":GOTO300
1620 R=353:GOTO400
1648 IFCP=176ANDNO=67THEN1667

```

```

1650 IFCP=176ANDNO=67THEN1667
1651 IFNO<>4ANDNO<>31ANDNO<>43ANDNO<>47ANDNO<>56AN
DNO<>64ANDNO<>101THENR=260:GOTO400
1652 IFCP=16ANDNO=31THEN1670
1653 IFNO=4ANDPC=0AND(OB%(4,0)=CPOROB%(4,0)=-1)THE
NR=261:PC=1:GOTO3500
1654 IFNO=4ANDPC=1AND(OB%(4,0)=CPOROB%(4,0)=-1)THE
NR=262:GOTO400
1655 IFNO=43AND(OB%(43,0)=-1OROB%(43,0)=CP)THENR=2
63:GOTO400
1656 IFNO=101AND(OB%(101,0)=-1OROB%(101,0)=CP)THEN
R=264:GOTO400
1657 IFNO=47ANDCP=33THEN1673
1658 IFNO=56ORNO=67THEN3300
1659 IFNO=64THEN1662
1661 R=265:GOTO400
1662 IFOB%(64,0)<>CPANDOB%(64,0)<>-1THENPRINT"[CD.
RVS.GREY]IT ISN'T HERE.":GOTO300
1663 BO=BO+1:R=266
1664 IFBO=5THENR=267:OB%(5,0)=CP:GOTO400
1665 IFBO>5THENPRINT"[CD]THE BOX IS NOW ATOMIC SIZ
E. AND YOU CAN'T OPEN IT.":GOTO300
1666 GOTO400
1667 IFMD=1THENR=274:GOTO400
1668 IFNO<>67THENR=275:GOTO400
1669 MD=1:R=276:P%(176,1)=183:P%(176,2)=177:P%(176
,3)=179:GOTO400
1670 IFGF=1THENPRINT"[CD,RVS.GREY]THE GRATE IS OPE
N.":GOTO300
1671 IFOB%(32,0)<>-1THENPRINT"[CD,RVS.GREY]YOU HAV
E NO KEY.":GOTO300
1672 GF=1:PRINT:PRINT"[RVS.GREY]THE GRATE SWINGS O
PEN.":P%(16,5)=17:GOTO2998
1673 IFCO=1THENR=268:GOTO400
1674 IFOB%(1,0)<>-1THENR=269:GOTO400
1675 R=270:OB%(6,0)=35:CO=1:GOTO400
1676 IFDO=1THENR=271:GOTO400
1677 IFRD=0THENR=272:GOTO400
1678 FORI=10TO17:A=I:B=25:C=129:GOSUB50:NEXT:GOTO4
00
1680 PRINT:PRINT"[RVS.ORG]YOU'LL HAVE TO TRY RE-P
HRASING THAT.":GOTO300
1710 IFNO<>37THEN1290
1711 NO=40:GOTO1740
1740 IFNO<>40ANDNO<>41THENR=279:GOTO400
1741 IFOB%(37,0)<>-1ANDOB%(38,0)<>-1THENPRINT"[RVS
.ORG]YOU'VE NOT GOT THE LAMP.":GOTO300
1742 IF(NO=40ANDLO=1)OR(NO=41ANDLO=0)THENPRINT"[CD
.BRN]IT IS ALREADY!":GOTO300
1743 IFNO=40THENLO=1:OB%(37,0)=0:OB%(38,0)=-1:GOTO
1745
1744 LO=0:OB%(37,0)=-1:OB%(38,0)=0

```

```

1745 PRINT"[CD,BRN].":GOTO300
1770 GOSUB5300
1771 IF OB%(NO,0)=-1 THEN PRINT"[CD,BRN]YOU'VE ALREAD
Y GOT IT.":GOTO300
1772 IF ZZ>7 THEN PRINT"[CD,BRN]YOU'RE CARRYING TOO M
UCH.":GOTO300
1773 IF ED=1 AND EF AND NO=29 THEN R=284:EF=1:GOTO400
1774 IF (CP=101 OR CP=233) AND NO=54 THEN R=286:GOTO400
1775 IF (CP=150 OR CP=166) AND NO=94 THEN R=287:GOTO400
1776 IF CP=243 AND NO=77 THEN R=289:GOTO400
1777 IF NO=60 AND GP=1 THEN R=288:GOTO400
1778 IF (NO>15 AND NO<29) THEN R=279:GOTO400
1779 IF NO=30 THEN I=050
1780 IF NO=31 OR NO=40 OR NO=41 OR (NO>46 AND NO<52) OR (NO>5
5 AND NO<60) THEN R=290:GOTO400
1781 IF NO=67 OR NO=70 OR NO=73 OR NO=74 OR NO=76 OR NO=78 OR (
NO>79 AND NO<100) THEN R=291:GOTO400
1782 IF NO=61 THEN R=292:GOTO400
1783 IF CP=28 AND NO=45 THEN R=294:GOTO400
1784 IF NO=65 OR NO=66 THEN R=293:GOTO400
1785 IF OB%(NO,0)<>CP THEN R=298:GOTO400
1786 IF NO=42 AND OB%(44,0)=-1 THEN R=295:GOTO400
1787 IF NO=42 AND OB%(43,0)<>-1 THEN R=296:GOTO400
1788 IF NO=2 AND CP=66 THEN R=297:GOTO400
1789 IF NO=3 AND CP=168 AND DK=0 THEN PRINT:PRINT"#WITH A
20 TON DRAGON LYING ON IT?":GOTO300
1790 IF CP=159 AND NO=2 AND GC=0 THEN PRINT:PRINT"[BRN]IT
'S STUCK TO THE GROUND.":GOTO300
1791 IF (NO=106 OR NO=36) AND OB%(34,0)<>-1 THEN R=368:GO
TO400
1792 IF NO=106 AND OB%(107,0)=-1 THEN NO=107:GOTO1771
1795 IF NO=106 THEN NO=107:ZZ=ZZ-1:OB%(106,0)=0:OB%(3
4,0)=0
1796 IF NO=36 THEN NO=35:ZZ=ZZ-1:OB%(36,0)=0:OB%(34,0
)=0
1797 IF NO=53 THEN PRINT:PRINT"[BRN]YOU MUST BE KIDDI
NG.":GOTO300
1799 PRINT"[CD,RVS,ORNG]OK.":ZZ=ZZ+1:OB%(NO,0)=-1:
GOTO300
1800 GOTO1680
1830 GOTO1770
1860 GOSUB5300
1861 IF OB%(NO,0)<>-1 THEN R=282:GOTO400
1862 IF NO=10 THEN R=283:GOSUB4000:GOTO1864
1863 IF NO=44 AND OB%(42,0)=CP THEN R=285:OB%(42,0)=0:G
OSUB4000:OB%(44,0)=CP:GOTO300
1864 IF CP=28 AND SK=0 AND NO=42 THEN R=299:OB%(42,0)=0:G
OTO400
1865 IF CP=168 AND DK=0 AND NO=42 THEN R=300:OB%(42,0)=0:
GOTO400
1866 IF CP=174 AND WK=0 AND NO=42 THEN WK=1:R=301:P%(174,
1)=175:OB%(42,0)=CP:ZZ=ZZ-1:GOTO400

```

```

1867 IFCP=66ANDBA=0ANDNO=2ANDB%(2,0)=-1THENR=348:
OB%(2,0)=0:OB%(39,0)=CP:BA=1:GOTO400
1868 IFNO=38THENOB%(38,0)=0:OB%(37,0)=CP:ZZ=ZZ-1:L
O=0:PRINT:PRINT"[RVS.GREY]OK.":GOTO300
1869 IF(NO=107ORNO=106)ANDB%(107,0)=-1THENR=369:O
B%(NO,0)=0:OB%(34,0)=CP:GOTO1875
1870 IF(NO=36ORNO=35)ANDB%(35,0)=-1THENR=369:OB%(
NO,0)=0:OB%(34,0)=CP:GOTO1875
1874 OB%(NO,0)=CP:PRINT"[CD,RVS.GREY]OK.":ZZ=ZZ-1:
GOTO300
1875 ZZ=ZZ-1:GOTO400
1890 IFEF=1ANDNO=29THENR=411:GOTO400
1891 IFCP=168ANDNO=63ANDP%(168,1)=0THENR=410:GOTO4
00
1892 IFTB=1ANDNO=54THENR=409:GOTO400
1893 IFNO=65ANDCP=174THENR=408:GOTO400
1894 IFNO<>45THENR=413:GOTO400
1895 IFP%(28,2)<>0THENR=414:GOTO400
1896 R=412:GOSUB4000:PRINT:PRINT:GOSUB60000
1897 IFLEFT$(CM$,3)<>"YES"THENR=415:GOSUB4000:GOTO
5000
1898 R=416:SK=1:P%(28,2)=29:P%(28,3)=30:OB%(45,0)=
0:GOTO400
1920 GOTO1890
1950 GOTO1890
1980 R=407:GOTO400
2010 GOTO1890
2040 IFCP=129ANDCM$<>"JUMP EVEN HIGHER"THENR=364:G
OTO400
2041 IFCP=129THENR=365:GOSUB4000:CP=130:GOTO402
2042 IFCP=130THENCN=129:GOTO402
2043 IF(CP=43ORCP=44)ANDWR=0THENR=361:GOSUB4000:CP
=87-CN:GOTO402
2044 IF(CP=43ORCP=44)THENR=363:GOSUB4000:GOTO5000
2045 A=0:FORI=0TO5:IFP%(CP,I)=-1THENA=A+1
2046 NEXT
2047 IFA>0THENR=360:GOSUB4000:GOTO5000
2048 R=362:GOTO400
2070 IFNO<>44THENR=359:GOTO400
2071 IFOB%(44,0)<>-1THENPRINT:PRINT"[RVS.GREY]YOU
AIN'T GOT IT!":GOTO300
2072 IFWR=1THENPRINT:PRINT"[RVS.GREY]AGAIN?":GOTO3
00
2073 IFCP<>43THEN1592
2074 R=358:WR=1:GOTO400
2100 GOTO2130
2130 IFCP=52ANDP%(52,1)=0ANDNO=50THENR=382:GOTO400
2131 R=383:GOTO400
2160 GOTO2130
2190 IFCP=1ANDNO=81THENCN=2:GOTO402
2191 IFNO=65ORNO=63ORNO=61ORNO=54ORNO=45ORNO=42ORN
O=29THENR=357:GOTO400

```

```

2192 PRINT:PRINT"[RVS.GREY]SORRY. DON'T UNDERSTAND
    THAT ONE. ":GOTO300
2220 IF OB%(35.0)<>-1 THEN R=384:GOTO400
2221 IF NO<>52 OR CP<>55 THEN R=369:GOTO2230
2222 IF WP=0 THEN WP=1:R=385:GOTO2230
2223 IF WP=1 THEN WP=2:R=386:GOTO2230
2224 IF WP=2 THEN WP=3:R=387:GOTO2230
2225 IF WP=3 THEN WP=4:R=388:GOTO2230
2226 IF WP>3 THEN R=389:GOTO2230
2230 OB%(35.0)=0:OB%(34.0)=-1:GOTO400
2250 IF NO<>4 AND NO<>31 AND NO<>43 AND NO<>47 AND NO<>56 AND
    NO<>64 AND NO<>101 THEN R=277:GOTO400
2251 IF CP=16 AND NO=31 THEN R=278:GOTO400
2252 IF NO=56 AND (CP=176 OR CP=125) THEN PRINT"[CD,RVS,0
    RNG]IT WON'T BUDGE. ":GOTO300
2253 IF NO=47 AND CP=33 THEN PRINT"[CD,RVS,ORNG]IT SNAP
    PED SHUT AFTER OPENING ANYWAY. ":GOTO300
2254 IF NO=64 AND (OB%(64.0)=CP OR OB%(64.0)=-1) THEN PRI
    NT"[CD,RVS,ORNG]YOU CAN'T. ":GOTO300
2255 IF NO=4 AND (OB%(4.0)=CP OR OB%(4.0)=-1) THEN PRINT"
    [CD,RVS,ORNG]YOU CAN'T. ":GOTO300
2256 GOTO1661
2280 IF OB%(107.0)<>-1 THEN PRINT:PRINT"[GREY]WHAT WI
    TH?":GOTO300
2281 IF CP<>125 THEN R=366:OB%(107.0)=0:OB%(34.0)=-1:
    GOTO400
2282 IF RD=1 THEN PRINT:PRINT"[RVS.GREY]AGAIN?":GOTO3
    00
2283 RD=1:R=367:P%(125.1)=126:OB%(107.0)=0:DO=1:OB
    %(34.0)=-1:GOTO1678
2310 R=390:GOTO400
2340 IF CP<>159 AND NO<>2 THEN R=391:GOTO400
2341 IF OB%(62.0)<>-1 THEN R=392:GOTO400
2342 IF GC=1 THEN R=393:GOTO400
2344 GC=1:R=394:OB%(2.0)=CP:GOTO400
2370 IF CP<>159 AND NO<>2 THEN 1770
2371 GOTO2340
2400 IF CP=249 THEN 2404
2401 IF CP<>55 AND NO<>52 THEN R=383:GOTO400
2402 IF OB%(55.0)<>-1 THEN R=395:GOTO400
2403 R=396:GOTO400
2404 IF NO<>79 THEN R=383:GOTO400
2405 IF OB%(55.0)<>-1 THEN 2402
2406 IF BC=1 THEN PRINT:PRINT"[RVS.GREY]AGAIN?":GOTO3
    00
2407 BC=1:R=424:GOTO400
2430 GOTO2400
2460 IF (CP=55 OR CP=56) AND WP<>3 AND NO=52 THEN R=460:GOT
    O400
2461 IF (CP=55 OR CP=56) AND WP=3 AND NO=52 THEN R=461:GOSU
    B4000:CP=111-CP:GOTO402
2462 IF OB%(68.0)<>CP AND OB%(68.0)<>-1 THEN R=459:GOTO
    206

```



```

400
2463 IFCP<>190ANDCP<>191ANDCP<>216ANDCP<>217THENR=
462:GOTO400
2464 IFNO<>68THENPRINT:PRINT"[BRN]YOU CAN'T DO THA
T.":GOTO300
2465 IFCP=190ORCP=191THENR=463:GOSUB4000:CP=381-CP
:OB%(68,0)=CP:GOTO2468
2466 IFCP=216ORCP=217THENR=463:GOSUB4000:CP=433-CP
:OB%(68,0)=CP:GOTO2468
2467 R=383:GOTO400
2468 IFEF=1THENPRINT:PRINT"[BRN]EDDY FOLLOWS YOU I
N AN UNUSUAL BURST OF SPEED.":GOTO300
2469 PRINT:PRINT"[RVS.GREY]AND HOW'S POOR OLD EDDY
GOING TO FOLLOW YOU NOW?":GOTO402
2490 IFCP<>193ORNO<>69THENR=383:GOTO400
2491 IFOB%(2,0)<>-1THENR=444:GOSUB4000:CP=192:GOTO
402
2492 IFEF=1THENR=446:GOSUB4000:CP=192:GOTO402
2493 OB%(2,0)=0:OB%(69,0)=-1:R=448:GOSUB4000:CP=19
4:GOTO402
2520 GOTO1680
2550 IFCP<>193THENR=265:GOTO400
2551 IFNO<>70THENR=265:GOTO400
2552 IFOB%(2,0)<>-1THENR=445:GOTO400
2553 R=447:OB%(2,0)=0:GOSUB4000:CP=192:GOTO402
2580 P%(206,2)=207:IFCP>216ORCP<206THENR=440:GOTO4
00
2581 IFOB%(72,0)<>-1THENR=441:GOTO400
2582 IFOB%(71,0)<>CPTHENR=443:GOTO400
2583 IFNO<16ORNO>27THENPRINT:PRINT"[RVS.GREY]GIVE
ME A DIRECTION TOO.":GOTO300
2584 IFNO>21THENNO=NO-6
2585 NO=NO-16:IFCP=206AND(NO=0ORNO=3)THEN2588
2586 IFCP=216ANDNO=1THEN2588
2587 R=442:GOSUB4000:IFP%(CP,NO)<>0THENCN=P%(CP,NO
):OB%(71,0)=CP:GOTO402
2588 PRINT:PRINT"[BRN]YOU APPEAR TO BE TRYING TO R
OW ON DRY LAND.":GOTO300
2610 IFCP<>221ANDNO<>69THENPRINT:PRINT"[RVS.GRY]NO
THING DOING.":GOTO300
2611 IFOB%(69,0)<>-1THENPRINT:PRINT"[RVS.GREY]NO-O
NE'S INTERESTED.":GOTO300
2612 R=449:GOSUB4000:OB%(69,0)=0:ZZ=ZZ-1:P%(221,3)
=222:GOTO300
2640 GOTO2610
2670 IFNO<>74THENR=331:GOTO400
2671 IFCP<>226THENR=332:GOTO400
2672 R=333:GOSUB4000:CP=229:GOTO402
2700 IFCP=206ORCP=216THENR=434:GOSUB4000:GOTO5000
2701 IFCP>206ANDCP<216THENR=436:GOSUB4000:GOTO5000
2702 IFCP=243ANDSA=0THENR=435:GOSUB4000:GOTO5000
2703 IFCP=243THENR=438:GOSUB4000:CP=244:GOTO402

```

```

2704 R=437:GOTO400
2730 GOTO2310
2760 IFCP<252THENR=417:GOTO400
2761 IFNO<>80THENR=418:GOTO400
2762 IFCP<>253THENR=419:GOTO2765
2763 IF0B%(44,0)<>252THENR=420:GOTO2765
2764 R=421
2765 GOSUB4000:GOTO5000
2790 R=422:GOTO400
2820 IF0B%(NO,0)=CPTHENPRINT:PRINT"[RVS.GREY]IN FR
ONT OF YOU.":GOTO300
2821 IF0B%(NO,0)=-1THENPRINT:PRINT"[RVS.GREY]YOU'R
E CARRYING IT. STUPID!":GOTO300
2822 R=423:GOTO400
2850 GOTO2460
2880 IFNO=101AND0B%(101,0)=-1THENR=450:GOTO400
2881 IFNO=101THENPRINT:PRINT"[RVS.GREY]WHAT MAGAZI
NE?":GOTO300
2882 IFCP=19ANDNO=87THENR=451:GOTO400
2883 IFCP=74ANDNO=90THENR=452:GOTO400
2884 IFCP=80THENR=453:GOTO400
2885 IFCP=81ANDNO=87THENR=454:GOTO400
2886 IFCP=252ORCP=253THENR=455:GOTO400
2887 IFCP=137AND(NO=91ORNO=92)THENR=456:GOTO400
2888 IFCP=199AND(NO=91ORNO=92)THENR=457:GOTO400
2889 IFCP=176ANDNO=72THENPRINT:PRINT"[BRN]PECULIAR
BRAND OF DWARVISH : CAN'T READ IT.":GOTO300
2890 R=458:GOTO400
2910 R=439:GOTO400
2940 GOTO2910
2970 IFCP=189ANDNO=73THENR=432:GOTO400
2971 R=433:GOTO400
2998 P%(17.4)=16:GOTO300
3000 GOTO1680
3030 QY=0:PRINT"[CD.RVS.BRN]ARE YOU SURE (Y OR N)?
3031 GETSP$:IFSP$="N"THEN300
3032 IFSP$<>"Y"THEN3031
3033 PRINT"[CD.RVS.BRN]SAVE PROGRESS TO DISK (Y OR
N)?
3034 GETSP$:IFSP$=""THEN3034
3035 IFSP$="N"THEN5000
3036 IFSP$<>"Y"THEN3034
3037 QY=1:GOTO1020
3060 FORI=1TO253:P%(I,6)=1:NEXT:PRINT"[CD.RVS.GREY
10K.":GOTO300
3090 FORI=1TO253:P%(I,6)=0:NEXT:PRINT"[CD.RVS.GREY
10K.":GOTO300
3120 REM PLUGH RESPONSE
3121 IFCP=38THENCP=74:R=CP-1:GOSUB4000:GOTO300
3123 IFCP<>74THENPRINT:PRINT"[BRN]NOTHING HAPPENS.
":GOTO300
3124 FORI=1TONN:IF0B%(I,0)=-1THEN0B%(I,0)=CP
208

```

```

3125 NEXT:OB%(38.0)=-1:CP=38:R=CP-1:GOSUB4000
3126 IFOB%(8.0)=74THENOB%(8.0)=-1
3127 IFOB%(107.0)=74THENOB%(107.0)=-1
3128 GOTO300
3150 IFCP<74ORCP>80THEN3160
3151 DD=DD+1:IFDD=12THENR=355:GOSUB4000:GOTO5000
3159 RETURN
3160 IFCP<248THENRETURN
3161 MC=MC+1:IFMC=12THENR=355:GOSUB4000:GOTO5000
3162 RETURN
3180 IFCP=20RCP=70RCP=80RCP=17BOR(CP>205ANDCP<217)
ORCP=179THENOB%(36.0)=CP
3181 IFCP<74ORCP>80THEN312
3182 IFLO=0THENR=356:GOSUB4000:GOTO300
3183 GOTO312
3210 IFCP=250THENC=226:GOSUB4000:GOTO3213
3211 IFCP=226THENC=250:GOTO402
3212 GOTO1592
3213 QA=1:GOSUB1080:IFA>260ANDCC=0ANDCP>43THENC=1
:LC=1:QA=0:GOTO3215
3214 QA=0:GOTO300
3215 LI=50:PRINT:PRINT"[GREY]A VOICE BOOMS OUT 'CA
VES CLOSING. EXIT
3216 PRINT"BY MAIN ENTRANCE ONLY'."
3217 FORI=1TONN:IFOB%(I.0)=-1THENOB%(I.0)=0
3218 NEXT:OB%(44.0)=253:OB%(38.0)=-1:OB%(55.0)=-1:
BF=0:GOTO300
3300 IFCP=125THEN1676
3301 IFCP=176THEN1667
3302 IFCP<>249THEN1661
3303 IFBC=0THENR=425:GOTO400
3304 IFOB%(32.0)<>-1THENR=430:GOTO400
3305 R=431:P%(249.0)=251:P%(249.3)=250:GOTO400
3330 PRINT:PRINT"[BRN]OKAY THEN:":PRINT:PRINTCHR$(
34):MID$(CM$,5):CHR$(34):PRINT:PRINT"SATISFIED?":G
OTO300
3500 OB%(1,0)=59:OB%(2,0)=159:OB%(3,0)=168:OB%(4,0
)=123:OB%(5,0)=0
3501 OB%(6,0)=0:OB%(7,0)=187:OB%(8,0)=78:OB%(9,0)=
105:OB%(10,0)=142
3502 OB%(11,0)=163:OB%(12,0)=199:OB%(13,0)=90:OB%(
14,0)=250:OB%(15,0)=251
3509 GOTO400
3600 PRINT:PRINT"[RVS,GREY]'CAVES CLOSED'." :CP=252
:GOTO402
4000 PRINT"[BRN]":IFR>252THENR=R+7:GOTO4002:REM M
ESSAGE
4001 R=CP-1:IFSS=10RP%(CP,6)=1THENPRINTP$(CP):GOTO
4020
4002 P%(CP,6)=1:A$=STR$(R):A$=MID$(A$,2):F3=F2
4003 IFLEN(A$)=1THENF$=A$:A$="COL0":GOTO4006
4004 IFLEN(A$)=2THENF$=MID$(A$,2):A$="COL"+LEFT$(A

```

```

$,1):GOTO4006
4005 F$=RIGHT$(A$,1):A$="COL"+LEFT$(A$,2)
4006 F2=VAL(F$):POKE923,F2+48
4007 IFF2<=F3THENCLOSE6:OPEN6,8,6,"0:"+A$+" ,S,R":B
$=A$:GOTO4009
4008 IFA$<>B$THENCLOSE6:OPEN6,8,6,"0:"+A$+" ,S,R":B
$=A$
4009 SYS900
4010 GET#6,Z$
4011 IFZ$="J"THENFORI=1TO2000:NEXT:PRINT:GOTO4010
4012 IFZ$="#"THEN4014
4013 PRINTZ$:GOTO4010
4014 IFR>252THENPRINT:RETURN
4015 PRINT
4020 GOSUB6000:RETURN
5000 REM PLAY AGAIN OPTION
5002 PRINT:PRINT:PRINT"[CD,RVS,BRN]PLAY AGAIN (Y O
R N)?
5003 GETSP$:IFSP$=""THEN5003
5004 IFSP$="N"THEN5010
5006 IFSP$<>"Y"THEN5003
5008 RUN
5010 QY=1:PRINT:GOSUB1080:CLOSE6:END:REMSYS64738
5300 IFNO=37ANDB%(38,0)=-1THENNO=38
5301 IFNO=34ANDB%(35,0)=-1THENNO=35
5302 IFNO=34ANDB%(107,0)=-1THENNO=107
5303 IFNO=36ANDB%(35,0)=-1THENNO=35
5304 IFNO=106ANDB%(107,0)=-1THENNO=107
5309 RETURN
6000 IF(CP=16ORCP=17)ANDGF=0THENPRINT:PRINT"[RVS,G
REY]THE GRATE IS LOCKED.":RETURN
6001 IFCP=16ORCP=17THENPRINT:PRINT"[RVS,GREY]THE G
RATE IS OPEN.":RETURN
6002 IFCP>202ANDDF=1THENDF=0:RETURN
6004 IFCP=28ANDSK=0THENR=303:PRINT:PRINT:GOSUB4000
:A=200:B=29:C=129:GOSUB50:RETURN
6006 IFCP=28THENR=304:GOTO401
6008 IFCP=33THENR=305:GOTO401
6010 IFCP=52ANDBB=0THENR=306:GOTO401
6013 IF(CP>43ANDCP<206)ANDPV=0ANDB%(4,0)<>2THEN62
00
6014 IFCP=55ANDWP=0THENR=308:GOTO401
6016 IFCP=55ANDWP=1THENR=309:GOTO401
6018 IFCP=55ANDWP=2THENR=310:GOTO401
6020 IFCP=55ANDWP=3THENR=311:GOTO401
6022 IFCP=55ANDWP>3THENR=312:GOTO401
6024 IFCP=66THENR=313:GOTO401
6026 IFCP=26ANDED=0THENR=314:ED=1:EF=1:GOTO401
6027 IFCP=103THENTB=0:RETURN
6028 IFCP=101ANDTB=0ANDVB<>0THENR=315:TB=1:GOTO401
6029 IFCP=102THENPRINT:PRINT"[BRN]HERE IS NO SIGN
OF THE TROLL.":RETURN

```

```

6030 IFCP=123ANDFC=0THENFC=1:R=320:GOTO401
6032 IFCP=125ANDDO=0THENR=321:GOTO401
6034 IFCP=159ANDGC=0THENR=322:GOTO401
6036 IFCP=168ANDDK=0THENR=323:GOSUB4000:A=5:B=27:C
=129:GOSUB50:RETURN
6038 IFCP=174ANDWK=0THENR=324:GOTO401
6040 IFCP=176ANDMD=0THENR=325:GOTO401
6042 IFCP=193ANDP%(193.2)=0THENR=326:GOTO401
6044 IFCP=233ANDTB=0THENTB=1:R=327:GOTO401
6045 IFCP=234THENPRINT:PRINT"[BRN]THERE IS NO SIGN
OF THE TROLL."
6046 IFCP=248ANDSB=0THENS=1:R=329:OB%(37.0)=250:O
B%(38.0)=0:ZZ=ZZ-1:LO=0:GOTO400
6050 IFEF=1ANDCP<207THENPRINT:PRINT"[BRN]EDDY LUMB
ERS ON BEHIND YOU.":OB%(29.0)=CP
6051 IFEF=1AND(CP>206ANDCP<216)THENPRINT:PRINT"[BR
N]EDDY SWIMS ALONGSIDE YOU.":OB%(29.0)=CP
6052 IFEF=1ANDCP>215THENPRINT:PRINT"[BRN]EDDY LOPE
S ALONG BESIDE YOU.":OB%(29.0)=CP
6053 IFCP=248THENOB%(38.0)=0:OB%(37.0)=246:ZZ=ZZ-1
6100 IFCP>19ANDINT(RND(.5)*100)>97THENR=317:GOTO40
1
6102 IFCP>19ANDDW=0ANDINT(RND(.5)*100)>75THENDW=1:
OB%(55.0)=CP:R=318:GOTO401
6104 IFDF=1THENOB%(61.0)=CP:R=380:GOTO401
6105 IF(CP>19ANDCP<167)ANDRND(.5)*100>97ANDDW=1THE
NR=319:DF=1:OB%(61.0)=CP:GOTO401
6107 IFOB%(29.0)=CPANDEF=0THENEF=1:R=406:GOTO401
6199 RETURN
6200 IFINT(RND(.5)*100)<70THEN6014
6202 A=0:FORI=1TO15:IFOB%(I.0)=-1ANDOB%(I.1)>0THEN
A=A+1
6203 NEXT:IFA<1THEN6014
6204 PV=1:R=464:PRINT:GOSUB4000:PRINT:FORI=1TO15:I
FOB%(I.0)=-1ANDOB%(I.1)>0THENOB%(I.0)=123
6205 NEXT:GOTO6014
6300 SYS49152,570+VB*30.0
60000 A=TI:CM$=""
60002 PRINT"[RVS,GREY]*[OFF,CL]";
60005 GETZ$
60006 IF(TI-A)>3600THENPRINT" ":GOTO60150
60007 IFZ$=""THEN60005
60010 Z=ASC(Z$):IFZ>95THEN60005
60080 ZL=LEN(CM$):IFZL>27THEN60110
60100 IFZ>31THENCM$=CM$+Z$:PRINTZ$:GOTO60002
60110 IFZ=13ANDZLTHENPRINT" ":RETURN
60120 IFZ=20ANDZLTHENCM$=LEFT$(CM$,ZL-1):PRINTZ$:
60140 GOTO60002
60150 PRINT"[CD,RVS,GREY]GET ON WITH IT!":PRINT"[C
D,RVS,PUR]WHAT NOW ?[OFF] ":GOTO60000

```

If you look at lines 8-44 first of all, these read in a vast amount of data from disk that is necessary to keep the program up and running. In order, the program reads in the map data, the short room descriptions, the direction words, the verbs, the nouns, the object descriptions, the position of the objects at the start of the game, and finally the points value of all the objects. As in the original 'Colossal Cave' the idea of the game is to get all the treasures back to the building outside which the game begins, and then solve the mysteries of the end game. Consequently, various objects that you encounter are awarded points as treasures.

Line 101 opens up a file to the disk so that, when the disk is accessed for the first message or room description there is a file open. Otherwise we might accidentally get a FILE NOT OPEN? error and the program will come to an abrupt halt.

Now look at lines 4000-4020, which access the various messages and room descriptions from disk and display them on the screen. You've seen something similar to this routine in Chapter 3. The only difference now is that sometimes we'll want a message, and sometimes a room description, and the variable R determines which it is to be. Values of R greater than 252 mean that we want a message, and various steps have to be taken to sort that out, while values of R equal to 252 or less (0 to 252 makes 253 values, and there are 253 room descriptions involved in this game) mean that we're reading a room description. The rest of the code is then pretty much as it was in the Chapter 3 listing.

The only difference comes at the end, in line 4020. Here, if a message was displayed program execution returns from this subroutine, but if a room description was printed up (either long or short, as determined by line 4001) then we have to go to the subroutine starting at line 6000 before going back to the main program. This is mainly a straightforward 'if you're in this room and this has opened then print this message' type of routine.

Back to the main program

Back in the main part of the program, everything continues very much as normal, including the input routine starting at line 390 (although there are a few added checks for various words in lines 505-508, including the verb SAY, which we'll come to in a minute), until we arrive at line 990.

Lines 990-1049 are the load and save parts of the program, and these are used to record your progress on disk and restore a previously played game. The prompt in both cases is for the player to remove the master disk and insert a new, formatted disk in the drive: progress is saved to or loaded from this disk. When the save or load is finished you must remove this disk and re-insert the master disk, otherwise all sorts of chaos might ensue.

Lines 1080-1095 are the scoring part of the game, and the variable QY as used in line 1093 (and 1048 incidentally) is there to check whether you're quitting the game or not. If you are, then a rating is given before going back to the QUIT part of the program, otherwise program execution carries on as normal.

Lines 1470-1500 are the next interesting part of the program, and this is used to deal with your companion in the caves, Eddy the underground hippy. It is this section of code that handles the verb TALK, and in this game you're only allowed to talk to Eddy, and for that he must be in the same location as yourself, so the first two lines of this routine check for those two facts.

After that the program just checks for various things that you might want your intrepid underground friend to do, and responds and prints up a suitable message accordingly. In order to solve the game completely, it is necessary to enlist the help of Eddy on a number of occasions.

The rest of the code is pretty straightforward. After the last chapter you should be able to work out what most of it is doing and how the game as a whole was put together. References to various verbs, nouns and objects won't mean very much, since we haven't come to the routines to implement all of those yet, but everything will be revealed in the next few pages.

Now for the plug. If you really can't face typing all of this in, copies of it are available on disk direct from the publishers. The disk you'll get contains the introduction, the game itself, and all the necessary data files.

A word of warning, however. The program makes extensive use of string handling, and as you are probably aware this can occasionally cause problems on the 64 in the form of garbage collection. So if your copy of the program appears to 'freeze' for a few seconds, and nothing seems to be happening, don't be tempted to reach for the run/stop

key, just sit back and wait for a while until the computer sorts itself out.

There are two ways around this. One is continually to invoke a `PRINT FRE(0)` statement every time the `WHAT NOW ?` prompt appears, but this slows the whole game down.

The second way is to use a compiler, for which you'll find a listing in Chapter 10. With boundless generosity this is included on the disk as well (as indeed is *The Lost Wallet* listing), so that when you've finished examining the source code you can compile it and have a really fast game. But be sure that you always compile any program onto a disk containing *only* that program, as the compiler makes a lot of files while going about its business and these take up some considerable room on the disk.

There is another program from Duckworth which uses many of the techniques explored in this book, and that is 'The Odyssey', parts of which have cropped up in this book from time to time. That program invokes most of the machine-code routines we've discussed, and has the added bonus of residing in memory at all times: no diving off to the disk every five seconds. Mind you, we did have to split it up into two separate parts.

Enough of the promotion, back to verbs, nouns, objects and, first of all, short room descriptions.

Room descriptions

One of the features of 'Zork', and other great adventure games, is that they present you with a number of different sets of room descriptions. A long one, for the first time that you get there, a shorter one, for a second visit, and an extremely brief one for any subsequent visits.

That technique has been used, after a fashion, in 'Enormous Cave', although I've stuck to just two types of description, namely long and short. The long one is used when you first visit a location, and the short one for any further visits. However, if you want all room descriptions to be long or short, you just have to enter the verbs `LONG` or `SHORT` respectively.

Thus, if you've visited a location once and then come back to it to find a very short description but want to read the whole thing again, entering `LONG` and then `LOOK` will let you read the original descrip-

tion. This is there in case you missed a cryptic clue first time around.

```
20 DIMP$(253)
22 FORI=1TO253:READP$(I):NEXT
24 OPEN2,8,2,"0:SHORT,S,W"
26 FORI=1TO253:PRINT#2,P$(I):CHR$(13):NEXT
28 CLOSE2
99 END
100 DATA"YOU ARE AT END OF ROAD."
101 DATA"YOU ARE INSIDE BUILDING."
102 DATA"YOU ARE AT END OF ROAD."
103 DATA"YOU ARE IN MIDDLE OF FOREST."
104 DATA"YOU ARE AT TOP OF HILL."
105 DATA"YOU ARE AT BOTTOM OF HILL."
106 DATA"YOU ARE IN VALLEY IN FOREST."
107 DATA"YOU ARE AT SLIT IN ROCK."
108 DATA"YOU ARE IN 20 FOOT DEPRESSION."
109 DATA"YOU ARE SOUTH OF DEPRESSION."
110 DATA"YOU ARE ON ROUGH TRACK."
111 DATA"YOU ARE ON LONG EAST-WEST PATH WITH HIGHW
ALLS."
112 DATA"YOU ARE ON LONG EAST-WEST PATH."
113 DATA"YOU ARE ON LEVEL TRACK."
114 DATA"YOU ARE WALKING ALONG TRACK."
115 DATA"YOU ARE IN DEPRESSION WITH GRATE."
116 DATA"YOU ARE IN SMALL CHAMBER BENEATH
EPRESSION."
117 DATA"YOU ARE CRAWLING OVER COBBLES."
118 DATA"YOU ARE IN DEBRIS ROOM."
119 DATA"YOU ARE IN AWKWARD CANYON."
120 DATA"YOU ARE IN SPLENDID CHAMBER."
121 DATA"YOU ARE STUMBLING ALONG PASSAGE."
122 DATA"DEAD END."
123 DATA"YOU ARE IN GLOOMY TUNNEL."
124 DATA"DEAD END."
125 DATA"TOP OF SMALL PIT."
126 DATA"YOU ARE IN NARROW TUNNEL."
127 DATA"YOU ARE AT ONE END OF VAST HALL."
128 DATA"YOU ARE EAST OF HALL OF MISTS."
129 DATA"YOU ARE WEST OF HALL OF MISTS."
130 DATA"YOU ARE IN NARROW TUNNEL SMELLING OF SEAW
ATER."
131 DATA"YOU ARE IN SEDIMENTARY ROCK ROOM."
132 DATA"YOU ARE IN SHELL ROOM."
133 DATA"YOU ARE IN SMALL OPENING IN ROCK."
134 DATA"YOU ARE CLOSE TO HALL OF MISTS."
135 DATA"YOU ARE LOW MISTY TUNNEL."
136 DATA"YOU ARE IN LOW TUNNEL."
137 DATA"YOU ARE AT KINK IN ROCK."
138 DATA"YOU ARE IN NO-MIST ROOM."
139 DATA"YOU ARE ON DISTURBED GROUND."
```

140 DATA"YOU ARE CLOSE TO HALL OF MISTS."
 141 DATA"YOU ARE SOUTH OF HALL OF MISTS."
 142 DATA"YOU ARE IN LONG NARROW HALL."
 143 DATA"YOU ARE SOUTH OF GAPING CRACK."
 144 DATA"YOU ARE IN EAST SIDE CHAMBER."
 145 DATA"YOU ARE IN WEST SIDE CHAMBER."
 146 DATA"YOU ARE IN LOW TWISTING TUNNEL."
 147 DATA"YOU ARE IN SMOOTH LOW TUNNEL."
 148 DATA"YOU ARE IN LONG SLOPING CORRIDOR."
 149 DATA"YOU ARE IN BEDQUILT."
 150 DATA"YOU ARE EAST OF BEDQUILT."
 151 DATA"YOU ARE IN NOT-THE-SWISS-CHEESE ROOM."
 152 DATA"YOU ARE NORTH OF TWO PIT ROOM."
 153 DATA"YOU ARE IN TWO PIT ROOM."
 154 DATA"YOU ARE IN EASTERN PIT."
 155 DATA"YOU ARE AT TOP OF DROP."
 156 DATA"YOU ARE IN GIANT ROOM."
 157 DATA"YOU ARE EAST OF GIANT ROOM."
 158 DATA"YOU ARE NEAR TWISTING PASSAGEWAYS."
 159 DATA"YOU HAVE ENTERED MAZE OF PASSAGEWAYS."
 160 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 161 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 162 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 163 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 164 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 165 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 166 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 167 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 168 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 169 DATA"YOU ARE LOST IN MAZE OF PASSAGEWAYS."
 170 DATA"YOU ARE WEST OF TWO PIT ROOM."
 171 DATA"YOU ARE AT JUNCTION OF PATHS."
 172 DATA"YOU ARE IN ALCOVE."
 173 DATA"YOU ARE IN DARK ROOM."
 174 DATA"YOU ARE IN JUMBLE OF CORRIDORS."
 175 DATA"YOU ARE IN SERIOUS TROUBLE!"
 176 DATA"YOU ARE IN JUMBLE OF CORRIDORS."
 177 DATA"YOU ARE IN JUMBLE OF ROCKS."
 178 DATA"YOU ARE IN JUMBLE OF ROCKS."
 179 DATA"YOU ARE IN MYSTERIOUS ALCOVE."
 180 DATA"YOU ARE ON LOW TRACK NEAR ECCENTRIC
 REA."
 181 DATA"DEAD END."
 182 DATA"YOU ARE IN FRONT OF ARCHWAY."
 183 DATA"YOU ARE AT CROSS-ROADS."
 184 DATA"YOU ARE IN CONFUSION OF TUNNELS."
 185 DATA"YOU ARE TO EAST OF LAND."
 186 DATA"YOU ARE IN CONFUSION OF TUNNELS."
 187 DATA"YOU ARE TO WEST OF LAND."
 188 DATA"YOU ARE IN SHADOWY AREA."
 189 DATA"YOU ARE WEST OF CENTRE OF THIS LAND."

A

190 DATA"YOU ARE IN CENTRE OF ECCENTRIC AREA."
191 DATA"YOU ARE EAST OF CENTRE OF THIS LAND."
192 DATA"YOU ARE IN SHADOWY AREA."
193 DATA"YOU ARE AT FISSURES OF ROCK."
194 DATA"YOU ARE IN WINDY TUNNEL."
195 DATA"YOU ARE IN TUNNEL OF ROCK."
196 DATA"YOU ARE IN WINDY TUNNEL."
197 DATA"YOU ARE IN MAZE OF RUBBLE."
198 DATA"YOU ARE SOUTH OF ECCENTRIC AREA."
199 DATA"YOU ARE CLOSE BY DEEP CHASM."
200 DATA"YOU ARE ON NORTHERN SIDE OF CHASM."
201 DATA"YOU ARE ON SOUTHERN SIDE OF CHASM."
202 DATA"YOU ARE ON LOW EAST-WEST CORRIDOR."
203 DATA"YOU ARE AT TURN IN PATH."
204 DATA"YOU ARE ON WIDE PATH."
205 DATA"YOU ARE ON MAIN TRACK."
206 DATA"YOU ARE IN ENORMOUS CAVE."
207 DATA"YOU ARE IN VAST CAVE."
208 DATA"YOU ARE IN LOW TUNNEL."
209 DATA"YOU ARE IN SMLL HOLE."
210 DATA"YOU ARE IN PROPPED UP TUNNEL."
211 DATA"YOU ARE EAST OF GIANT ROOM."
212 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
213 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
214 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
215 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
216 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
217 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
218 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
219 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
220 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
221 DATA"YOU ARE IN AN ARRAY OF TWISTING TUNNELS."
222 DATA"YOU ARE AT CENTRE OF TUNNELS."
223 DATA"YOU ARE ON PATH HEADING DOWN."
224 DATA"YOU ARE AT OLD STEEL GRATE."
225 DATA"YOU ARE SOUTH OF GRATE."
226 DATA"YOU ARE IN THE HARD ROOM."
227 DATA"YOU ARE NEAR VOLCANO."
228 DATA"YOU ARE IN FRONT OF ROCK."
229 DATA"YOU ARE WEST OF LEDGE."
230 DATA"YOU ARE IN SMALL ROUND PASSAGE."
231 DATA"YOU ARE IN SMALL ROUND PASSAGE."
232 DATA"YOU ARE IN VERY DARK CORRIDOR."
233 DATA"YOU ARE ON EDGE OF PRECIPICE."
234 DATA"YOU ARE IN TINY CORRIDOR."
235 DATA"YOU ARE ON GENTLY SLOPING PSSAGE."
236 DATA"YOU ARE IN ODD-SHAPED ROOM."
237 DATA"YOU ARE IN LARGE OPEN SPACE."
238 DATA"YOU ARE IN HIGH ROOM."
239 DATA"YOU ARE IN HIGH ROOM."
240 DATA"YOU ARE IN WARM-WALLED ROOM."

241 DATA"YOU ARE IN COOL ROOM."
242 DATA"YOU ARE IN CLEAR-AIR ROOM."
243 DATA"YOU ARE NEAR MYSTERIOUS PATH."
244 DATA"YOU ARE IN WARM-WALLED ROOM."
245 DATA"YOU ARE IN SMALL CHAMBER."
246 DATA"YOU ARE ON EDGE OF BREATHTAKING VIEW."
247 DATA"YOU ARE WEST OF VOLCANO."
248 DATA"YOU ARE WEST OF LOW TUNNEL."
249 DATA"YOU ARE IN FRONT OF SPECTRE OF CAVES."
250 DATA"YOU ARE PAST SPECTRE OF CAVES."
251 DATA"YOU ARE IN DIRTY CHAMBER."
252 DATA"YOU ARE IN LOW ROUND CHAMBER."
253 DATA"YOU ARE IN STICKY ROOM."
254 DATA"YOU ARE IN THICK-MUD ROOM."
255 DATA"YOU ARE IN DISGUSTING SLIMY ROOM."
256 DATA"YOU ARE AT END OF MUD TUNNEL."
257 DATA"YOU ARE NORTH OF MAGNIFICENT CAVERN."
258 DATA"YOU ARE IN MAGNIFICENT CAVERN."
259 DATA"YOU ARE SOUTH OF MAGNIFICENT CAVERN."
260 DATA"YOU ARE IN ROOM WITH DEEP VOICE."
261 DATA"YOU ARE ON PATH OF MISTS."
262 DATA"YOU ARE ON PATH OF MISTS."
263 DATA"YOU ARE ON PATH OF MISTS."
264 DATA"YOU ARE ON PATH OF MISTS."
265 DATA"YOU ARE ON HARD GROUND."
266 DATA"YOU ARE IN VERY HIGH ROOM."
267 DATA"YOU ARE IN MYSTERIOUS SHADOWY ROOM."
268 DATA"YOU ARE ON WIDE CAVERNOUS PATH."
269 DATA"YOU ARE NEAR STAGNANT POOL OF WATER."
270 DATA"YOU ARE ON WIDE PATH."
271 DATA"YOU ARE IN LOW ROUND TUNNEL."
272 DATA"DEAD END."
273 DATA"YOU ARE UNDERNEATH ARCH OF ROCK."
274 DATA"YOU ARE SOUTH OF ARCHWAY."
275 DATA"YOU ARE AT DOORWAY."
276 DATA"YOU ARE IN MAGICAL LAND."
277 DATA"YOU ARE UNDERNEATH WATERFALL."
278 DATA"YOU ARE IN MAGICAL LAND."
279 DATA"YOU ARE ON ROCKY OUTPOST."
280 DATA"YOU ARE IN MAGIC LAND."
281 DATA"YOU ARE NEAR CENTRE OF MAGIC LAND."
282 DATA"YOU ARE AT CENTRE OF MAGIC LAND."
283 DATA"YOU ARE NEAR CENTRE OF MAGIC LAND."
284 DATA"YOU ARE IN MAGIC LAND."
285 DATA"YOU ARE IN SOUTH-WEST CORNER OF LAND."
286 DATA"YOU ARE HERE FOREVER!"
287 DATA"YOU ARE NEAR CENTRE OF LAND."
288 DATA"YOU ARE IN LARGE OPEN ROOM WITH
REATURES."
289 DATA"YOU ARE IN SOUTH-EAST CORNER OF LAND."
290 DATA"YOU ARE ON DEEP LEDGE."

C

291 DATA"YOU ARE ON LONG LOW DARK TUNNEL."
292 DATA"YOU ARE IN UNDERGROUND BAR."
293 DATA"YOU ARE BEHIND BAR."
294 DATA"YOU ARE IN STRAIGHT TUNNEL."
295 DATA"YOU ARE IN LOW CRAWL."
296 DATA"YOU ARE IN HIGH-CEILING ROOM."
297 DATA"YOU ARE NEAR ENORMOUS CANYON."
298 DATA"YOU ARE COLD ROOM."
299 DATA"YOU ARE IN ICE-WHITE CORRIDOR."
300 DATA"YOU ARE WHITE-SHEEN CORRIDOR."
301 DATA"YOU ARE IN SALTY SEA-WATER ROOM."
302 DATA"YOU ARE NEAR CLIFF-FACE."
303 DATA"YOU ARE LEDGE NEAR LAKE."
304 DATA"YOU ARE ON LOW LEDGE."
305 DATA"YOU ARE ON SHORE OF LAKE."
306 DATA"YOU ARE ON SURFACE OF LAKE."
307 DATA"YOU ARE ON SURFACE OF LAKE."
308 DATA"YOU ARE ON SURFACE OF LAKE."
309 DATA"YOU ARE ON SURFACE OF LAKE."
310 DATA"YOU ARE ON SURFACE OF LAKE."
311 DATA"YOU ARE ON SURFACE OF LAKE."
312 DATA"YOU ARE ON SURFACE OF LAKE."
313 DATA"YOU SEE LAND."
314 DATA"YOU ARE ON SURFACE OF LAKE."
315 DATA"YOU ARE ON SOUTHERN SHORE OF LAKE."
316 DATA"YOU ARE ON TOP OF LEDGE."
317 DATA"YOU ARE IN SMALL DARK CAVE."
318 DATA"YOU ARE ON ROCKY GROUND."
319 DATA"YOU ARE AT END OF CAVE."
320 DATA"YOU ARE NEAR ASTROTURF GARDEN."
321 DATA"YOU ARE ON SPRINGY SURFACE."
322 DATA"YOU ARE IN WELL-KEPT GARDEN."
323 DATA"YOU ARE IN GARDEN."
324 DATA"YOU ARE NEAR 4 BY 4 PLANTS."
325 DATA"YOU ARE NEAR WATER CREATURES."
326 DATA"YOU ARE IN ASTROTURF LAND."
327 DATA"YOU ARE FLOATING MAN."
328 DATA"YOU ARE ON WEST SIDE OF PIT."
329 DATA"DEAD END."
330 DATA"YOU ARE IN SHARPLY TURNING CORRIDOR."
331 DATA"YOU ARE NEAR OLD BRIDGE."
332 DATA"YOU ARE IN FRONT OF BRIDGE."
333 DATA"YOU ARE ON WEST SIDE OF BRIDGE."
334 DATA"YOU ARE ON THIRD LEVEL."
335 DATA"YOU ARE ON SECOND LEVEL."
336 DATA"YOU ARE ON FIFTH LEVEL."
337 DATA"YOU ARE ON FOURTH LEVEL."
338 DATA"YOU ARE ON SIXTH LEVEL."
339 DATA"YOU ARE ON SEVENTH LEVEL."
340 DATA"YOU ARE ON EIGHTH LEVEL."
341 DATA"YOU ARE ON NINTH LEVEL."

```

342 DATA"YOU ARE IN FRONT OF POOL OF WATER."
343 DATA"YOU ARE NORTH OF PIRANHA POOL."
344 DATA"YOU ARE AT BOTTOM OF CORRIDOR."
345 DATA"YOU ARE IN LOW ROUND CORRIDOR."
346 DATA"YOU ARE IN TEETH OF GALE."
347 DATA"YOU ARE IN DUSTY DARKNESS."
348 DATA"YOU ARE NEAR BRACKEN AND MOSS."
349 DATA"YOU ARE IN TINY ROOM."
350 DATA"YOU ARE IN SECRET SHADOWY CAVE."
351 DATA"YOU ARE AT NORTH-EAST END OF REPOSITORY."
352 DATA"YOU ARE AT SOUTH-WEST END OF REPOSITORY."

```

READY.

This listing is fairly easy to follow, as each data statement represents a separate location. The first five lines of code read in each description in turn, and then store the lot on disk under the filename SHORT.

If you are undertaking the task of typing in the entire listing, a word of advice. Save the introduction and the main part of the game on one disk, and this short room description maker and the other programs to come on another disk. Then, load in the data-making programs one by one and run them to create files on the disk that has the introduction and the main program. That way you'll end up with one disk that has the entire set of programs and files on it, and you can use that disk to store your progress on. There is just about enough room on one disk to cram everything on, if you keep the data-making programs separate from the others.

Having done all that, you can then load the complete game by typing in:

```
LOAD "0:*",8
```

and RUN. The introduction loads the main program, which in turn reads in all the files before starting to play the game.

This is what happens on the version sold by Duckworth, and you must use that same disk to LOAD and SAVE your progress.

More room descriptions

These are the long room descriptions, and as you can see they take up quite a bit of space. Take it a little bit at a time and it won't be

too bad.

But before typing in all the data statements you'll need to know how to file them on disk and check that you've typed them in properly, and the following piece of code is meant to be used with *all* the room description and message data statements.

```
0 DATA162,6,32,198,255,32,228,255,201,42,208,249,3
2,228,255,205,155,3,208
1 DATA241,76,204,255
2 FORI=900TO922:READA:POKEI,A:NEXT
4 READA:IFA=-1THENSTOP
5 A$=STR$(A):A$=MID$(A$,2):A$="COL"+A$
10 OPEN2,8,2,"@O:"+A$+",S,W"
20 READA$:FORI=1TOLEN(A$):Z$=MID$(A$,I,1)
21 IFZ$<>"^"THENPRINT#2,Z$;:NEXT:PRINT#2,CHR$(13);
:GOTO20
22 CLOSE2
23 GOTO4
25 B$="COL3":F2=0
26 OPEN6,8,6,"O:"+B$+",S,R"
27 PRINT:PRINT:INPUT "NO. ";A$:F3=F2:IFA$="-1"THENC
LOSE6:END
28 IFLEN(A$)=1THENF$=A$:A$="COLO":GOTO31
29 IFLEN(A$)=2THENF$=MID$(A$,2):A$="COL"+LEFT$(A$,
1):GOTO31
30 F$=RIGHT$(A$,1):A$="COL"+LEFT$(A$,2)
31 F2=VAL(F$):POKE923,F2+48
32 IFF2<=F3THENCLOSE6:OPEN6,8,6,"O:"+A$+",S,R":B$=
A$:GOTO35
33 IFA$<>B$THENCLOSE6:OPEN6,8,6,"O:"+A$+",S,R":B$=
A$
35 SYS900
37 GET#6,Z$
41 IFZ$="J"THENFORI=1TO2000:NEXT:PRINT:GOTO37
42 IFZ$="#"THEN27
43 PRINTZ$;:GOTO37
```

This is more or less the same as the program we saw in Chapter 3. A short machine-code routine is read in first, followed by a successive reading of all the data statements before filing them on disk. If we look at the first lot of room descriptions we can see how the program operates.

```
100 DATA0
102 DATA"*OYOU ARE STANDING AT THE END OF A ROAD
104 DATA"BEFORE A SMALL BRICK BUILDING. AROUND
106 DATA"YOU IS A FOREST. A SMALL STREAM FLOWS
```

108 DATA"OUT OF THE BUILDING AND MEANDERS OFF D
 OWN A VALLEY.#
 110 DATA"*1YOU ARE INSIDE THE BUILDING, A WELL
 112 DATA"HOUSE FOR A LARGE SPRING. THERE IS A
 114 DATA"TRAPDOOR IN THE FLOOR, ALTHOUGH THE ONLYE
 XIT OUT IS TO THE WEST.#
 122 DATA"*2YOU ARE STANDING AT THE END OF A ROAD
 124 DATA"BESIDE A SMALL BRICK BUILDING. AROUND
 126 DATA"YOU IS A FOREST. A SMALL STREAM FLOWS
 128 DATA"OUT OF THE BUILDING AND MEANDERS OFF D
 OWN A VALLEY.#
 130 DATA"*3YOU ARE IN THE MIDDLE OF THE FOREST,
 132 DATA"WITH TREES ALL AROUND YOU, AND*PATHS H
 EADING OFF EVERYWHERE.#
 140 DATA"*4YOU HAVE REACHED THE TOP OF A HILL IN
 142 DATA"THE FOREST.#
 150 DATA"*5YOU HAVE REACHED THE BOTTOM OF A HILL I
 NTHE FOREST.#
 160 DATA"*6YOU ARE IN A VALLEY IN THE FOREST,
 162 DATA"BESIDE A STREAM TUMBLING ALONG A ROCKY B
 ED.#
 164 DATA"*7AT YOUR FEET ALL THE WATER IN THE
 166 DATA"STREAM SPLASHES INTO A 2-INCH SLIT IN T
 HE ROCK.#
 170 DATA"*8YOU ARE IN A 20 FOOT DEPRESSION
 172 DATA"FLOORED WITH BARE DIRT. STRANGE, THERE
 174 DATA"USED TO BE A STEEL GRATE AROUND HERE
 176 DATA"SOMEWHERE, BUT I CAN'T SEEM TO SEE IT ATT
 HE MOMENT.#
 180 DATA"*9YOU ARE TO THE SOUTH OF THE DEPRESSION
 182 DATA"ON A ROUGH TRACK HEADING NORTH-SOUTH.#^
 200 DATA1
 202 DATA"*0THE PATH TAKES A TURN HERE, AND YOU CAN
 206 DATA"LOOK BOTH NORTH AND WEST. THE ROUGH
 208 DATA"NATURE OF THE SIDES OF THE PATH SUGGESTST
 HAT YOU CAN'T CLIMB UP.#
 212 DATA"*1YOU ARE ON A LONG WEST-EAST PATH. THE
 214 DATA"WALLS ARE QUITE HIGH NOW, AND CLOSE IN
 216 DATA"ABOVE YOUR HEAD.#
 220 DATA"*2YOU ARE ON A LONG WEST-EAST PATH.#
 230 DATA"*3YOU ARE ON A FAIRLY LEVEL TRACK MADE UP
 232 DATA"FROM ROUGH GRAVEL AND STONES. TO THE
 234 DATA"NORTH-EAST YOU CAN STILL MAKE OUT THE
 236 DATA"FOREST, WHILE TO THE SOUTH-WEST ARE THE
 238 DATA"TOWERING SLOPES OF A HIGH MOUNTAIN.#
 240 DATA"*4YOU ARE WALKING ALONG THE TRACK. THERE
 242 DATA"IS A DEPRESSION IN THE GROUND TO THE
 244 DATA"WEST, AND SIGNS OF A DEEPER TRACK TO
 246 DATA"THE EAST. THE VIEW NORTH REVEALS
 248 DATA"NOTHING.#

250 DATA"*5YOU ARE IN THE DEPRESSION, AND SET INTO
 252 DATA"THE FLOOR, AMONG THE DIRT, IS A STRONG S
 TEEL GRATE.#
 260 DATA"*6YOU ARE IN A SMALL CHAMBER BENEATH A 3
 262 DATA"BY 3 STEEL GRATE TO THE SURFACE. A LOW
 264 DATA"CRAWL OVER COBBLES LEADS INWARDS TO THE S
 OUTH.#
 270 DATA"*7YOU ARE CRAWLING OVER COBBLES IN A LOW
 PASSAGE.#
 280 DATA"*8YOU ARE IN A DEBRIS ROOM FILLED WITH
 282 DATA"STUFF WASHED IN FROM THE SURFACE. THE
 284 DATA"REMNANTS OF A NOTE ON THE WALL READ
 286 DATA"'MAGIC MOR', THE REST IS TORN AWAY.#
 288 DATA"*9YOU ARE IN AN AWKWARD SLOPING EAST-WEST
 CANYON.#^
 300 DATA2
 302 DATA"*0YOU ARE IN A SPLENDID CHAMBER 30 FEET
 303 DATA"HIGH. THE WALLS ARE FROZEN RIVERS OF
 304 DATA"ORANGE STONE. AN AWKWARD CANYON AND A
 306 DATA"GOOD PASSAGE EXIT FROM EAST AND WEST.#
 310 DATA"*1YOU ARE STUMBLING ALONG A LOW EAST-WEST
 312 DATA"PASSAGE. THERE ARE SIGNS OF A PIT
 314 DATA"AROUND HERE, BUT YOU CAN'T SEEM TO GET T
 O IT.#
 316 DATA"*2DEAD END.#
 320 DATA"*3A GLOOMY TUNNEL HEADS OFF TO THE WEST,
 322 DATA"WHILE A CLEAR PATH CAN BE SEEN TO THE E
 AST.#
 340 DATA"*4DEAD END.#
 350 DATA"*5AT YOUR FEET IS A SMALL PIT BREATHING
 352 DATA"TRACES OF WHITE MIST. A SMALL CRACK
 354 DATA"LEADS OFF TO THE SOUTH, AND THE SPLENDIDC
 HAMBER LIES TO THE NORTH.#
 360 DATA"*6A NARROW TUNNEL LEADS OFF TO THE NORTH,
 362 DATA"WHILE TO THE SOUTH LIES A STONE STAIR-
 364 DATA"CASE LEADING DOWNWARDS. THE STEPS HAVE
 366 DATA"CRUMBLD SLIGHTLY, BUT STILL SEEM R
 EASONABLY SAFE.#
 370 DATA"*7YOU ARE AT ONE END OF A VAST HALL
 372 DATA"STRETCHING OUT OF SIGHT TO EAST AND
 374 DATA"WEST. THE HALL IS FILLED WITH WISPS OF
 376 DATA"WHITE MIST SWAYING TO AND FRO ALMOST AS
 378 DATA"ALIVE.#
 380 DATA"*8YOU ARE IN THE EAST SIDE OF THE HALL OF
 MISTS.#
 390 DATA"*9YOU ARE IN THE WEST SIDE OF THE HALL OF
 MISTS.#^"
 400 DATA-1

READY.

This gives us our first twenty room descriptions, starting at location one, just outside the building. This is somewhat confusingly referred to as description zero by the data statements, but we won't worry about that.

As ever, when typing this lot in, look out for the 'chinaman's hat' symbol, which occurs in line 21 of the program that files everything away, and at the end of each group of ten room descriptions. This is meant to be the up-arrow key, which sits just above the RETURN key on the keyboard.

If you load in the file-making program, and then type in the program headed ROOM DESCRIPTIONS PART ONE, you can save the whole lot as a program called (for example) COLDATA0-2. Then you can run the program, and watch the disk drive light flash away at you as the various files are created on the disk. When it runs out of data, or if you've been a good boy and typed in the final -1 data statement, the program will stop and you can test to see whether or not you've typed in the descriptions without any spelling mistakes.

Enter:

GOTO25

The disk drive light will come on, and you'll be prompted for a number. Since we've created room descriptions zero to nineteen for the first twenty locations, type in a number between zero and nineteen. Since we're only checking at this point you might as well go through them all in sequence to make sure everything's okay. Say you typed in number zero, then lo and behold the following will appear on the screen:

You are standing at the end of a road
before a small brick building. Around
you is a forest. A small stream flows
out of the building and meanders off
down a valley.

Or at least it should do. Note that letters in *italics* (e.g. the *Y* of the first word *YOU* in line 102) are meant to be upper case, while everything else is lower case. I must change my printer sometime!

Typing in number one will produce the next room description, and so on. If you find any errors simply correct the appropriate data statement and re-run the entire program, which will wipe out the existing

files and create a whole load of new ones. Then you can do the GOTO25 check again just to make sure.

It doesn't happen in this first lot of descriptions, but occasionally you may find that the 64 appears to skip a line on the screen when a description is being printed, so that something might come out looking like this:

You are west of the GIANT room. The ceiling is lower here, although the vast

chasm still lies to the south. Many openings can be seen along the track to the east.

No problem, as they say. Go to the data statement that starts:

chasm still lies ...

and before the first letter of the word 'chasm', or whatever word is causing you trouble, insert a cursor-up symbol by putting the cursor over the letter 'c', pressing shift and INST/DEL, and then cursor-up. Hitting RETURN and re-running the program will correct the offending line.

Now for the rest of the descriptions.

```
100 DATA3
102 DATA"*0YOU ARE IN A NARROW TUNNEL.  THE AIR
104 DATA"SMELLS STRONGLY OF SEA-WATER HERE, AND
106 DATA"THE MIST IS QUITE THIN.  PATHS HEAD OFF T
O WEST, EAST AND SOUTH.*
110 DATA"*1YOU ARE IN A LARGE ROOM CARVED OUT OF
112 DATA"SEDIMENTARY ROCK.  THE FLOORS AND WALLS
114 DATA"ARE LITTERED WITH BITS OF SHELL
116 DATA"EMBEDDED INTO THE STONE.  A SHALLOW
118 DATA"PATH LEADS SOUTHWARDS, AND A LOW TUNNEL E
XITS TO THE EAST.*
120 DATA"*2YOU ARE IN THE SHELL ROOM.*
130 DATA"*3YOU ARE IN A SMALL OPENING IN THE ROCK.
132 DATA"THE AIR SMELLS STRONGLY OF SEA-WATER,
134 DATA"AND THERE ARE PATHS HEADING NORTH, WEST A
ND EAST.*
140 DATA"*4YOU ARE CLOSE TO THE HALL OF MISTS.
142 DATA"*1IST DANCES AROUND THE TUNNEL, MAKING IT
144 DATA"DIFFICULT TO SEE WHICH DIRECTION TO      T
AKE.*
150 DATA"*5YOU ARE IN A LOW, MISTY TUNNEL.  PATHS
```

152 DATA"CAN DIMLY BE MADE OUT HEADING TO WEST
 154 DATA"AND EAST, WHILE TO THE SOUTH THE AIR L
 OOKS TO BE SLIGHTLY CLEARER.#
 160 DATA"*6YOU ARE FORCED TO CRAWL THROUGH A LOW
 TUNNEL.#
 170 DATA"*7YOU ARE AT A KINK IN THE ROCK. YOU CAN
 172 DATA"BARELY FIT THROUGH THIS NARROW PASSAGE,
 174 DATA"THE ROCK TAKING ON STRANGE, CONTORTED G
 EOMETRIC PATTERNS.#
 180 DATA"*8NO MIST TO BE SEEN. PATHS LEAD OFF TO
 NORTH AND WEST.#
 190 DATA"*9THE GROUND IS QUITE DISTURBED HERE. A
 192 DATA"HINT OF A PIT LEADS DOWN, BUT IT'S HARD
 194 DATA"TO SEE HOW YOU COULD FIT THROUGH IT.#^
 200 DATA4
 202 DATA"*0YOU ARE CLOSE TO THE HALL OF MISTS. 7H
 E
 204 DATA"PATH IS QUITE CRUMBLY HERE, AND IT IS
 206 DATA"HARD TO KEEP A FOOTING. MIST DANCES
 208 DATA"ABOUT EVERYWHERE, AND ONLY DIMLY CAN YOU
 209 DATA"MAKE OUT PATHS TO NORTH, WEST AND EAST.#
 210 DATA"*1YOU ARE AT THE SOUTH END OF THE HALL OF
 212 DATA"MISTS. THINGS SEEM A LITTLE CLEARER TO
 214 DATA"THE SOUTH, MORE MISTY TO THE NORTH.#
 220 DATA"*2YOU ARE IN A LONG NARROW HALL. AT YOUR
 222 DATA"FEET IS A LARGE, GAPING CRACK, WHICH
 224 DATA"LOOKS MUCH TOO WIDE TO JUMP. LITTLE
 226 DATA"TRACES OF MIST PLAY ABOUT YOUR FEET, AND
 228 DATA"IN THE DIM LIGHT THE BOTTOM OF THE PIT
 229 DATA"LOOKS TO BE A MIGHTY LONG WAY OFF.#
 230 DATA"*3YOU ARE SOUTH OF THE GAPING CRACK, IN
 232 DATA"THE HALL OF THE MOUNTAIN KING.
 234 DATA"DIRECTIONS GET CONFUSED DOWN HERE, BUT
 236 DATA"IT SEEMS TO ME THAT PATHS LEAD NORTH,
 238 DATA"SOUTH, EAST AND WEST. SUCH A CHOICE!#
 240 DATA"*4YOU ARE IN A WEST SIDE CHAMBER OF THE
 242 DATA"HALL OF THE MOUNTAIN KING.#
 250 DATA"*5YOU ARE IN AN EAST SIDE CHAMBER OF THE
 252 DATA"HALL OF THE MOUNTAIN KING. THERE IS
 254 DATA"LITTLE HEADROOM HERE.#
 260 DATA"*6YOU ARE IN A VERY LOW, TWISTING TUNNEL.
 262 DATA"THE ROCK OVERHEAD IS SCARCELY HIGH
 264 DATA"ENOUGH TO CRAWL UNDER.#
 270 DATA"*7YOU ARE IN A VERY LOW, TWISTING TUNNEL.
 272 DATA"ALTHOUGH THE ROCK WALL IS QUITE SMOOTH
 274 DATA"TO THE TOUCH, YOU STILL HAVE TO BE
 276 DATA"CAREFUL TO AVOID THE LOW CEILING AND
 278 DATA"HEMMED-IN WALLS.#
 280 DATA"*8YOU ARE IN A LONG SLOPING CORRIDOR WITH
 RAGGED SHARP WALLS.#
 290 DATA"*9YOU ARE IN *BEDQUILT*, A LONG EAST-WEST

292 DATA"PASSAGE WITH HOLES EVERYWHERE.#^
 300 DATA\$
 302 DATA"*0YOU ARE EAST OF *BEDQUILT*. THE ROCK
 304 DATA"CLOSES IN, ALLOWING EXITS ONLY TO NORTH A
 ND WEST.#
 310 DATA"*1YOU ARE IN A ROOM WHOSE WALLS DON'T LOO
 K
 312 DATA"ANYTHING LIKE SWISS CHEESE. HOLES LEAD
 314 DATA"OFF EVERYWHERE, ALTHOUGH YOUR PROGRESS
 316 DATA"IS HAMPERED BY A LARGE BEDROCK BLOCK IN
 318 DATA"THE MIDDLE OF THE ROOM.#
 320 DATA"*2YOU ARE NORTH OF THE TWO PIT ROOM.#
 330 DATA"*3YOU ARE IN THE TWO PIT ROOM. TO WEST
 332 DATA"AND EAST LIE GOOD PASSAGES, BUT IN BOTH
 334 DATA"DIRECTIONS YOU CAN SCARCELY SEE BEYOND
 336 DATA"EXTREMELY DEEP PITS. THE TRACK NORTH
 337 DATA"LEADS BACK TO *BEDQUILT*, WHILE TO THE
 338 DATA"SOUTH LIES NOTHING BUT AN EXPANSE OF S
 OLID ROCK.#
 340 DATA"*4YOU HAVE WALKED DOWN INTO THE EASTERN
 342 DATA"PIT IN THE TWO PIT ROOM. YOU COULD
 344 DATA"WALK OUT TO THE WEST, BUT TO THE EAST
 346 DATA"YOU CAN SEE NOTHING BUT A ROUND HOLE, A
 348 DATA"GOOD 15 FEET OFF THE GROUND. THE SOLID
 349 DATA"ROCK HERE DOES NOT ALLOW YOU TO CLIMB UP A
 ND REACH THE HOLE.#
 350 DATA"*5YOU ARE STANDING AT THE TOP OF A 15
 352 DATA"FOOT DROP. TO THE EAST LIES A GIANT
 354 DATA"CORRIDOR, WHILE WEST LIES ONLY OBLIVION.#
 360 DATA"*6YOU ARE IN THE *GIANT* ROOM, WITH PATHS
 362 DATA"LEADING EAST AND WEST. TO THE SOUTH,
 363 DATA"SOME 100 FEET AWAY, YOU CAN MAKE OUT A
 364 DATA"WINDOW INTO ANOTHER ROOM, AND A SHADOWY
 365 DATA"FIGURE CAN BE SEEN THERE, PEERING BACK
 366 DATA"AT YOU. BETWEEN YOU AND IT IS A VAST
 367 DATA"CHASM, DROPPING OUT OF SIGHT TO THE
 368 DATA"FLOOR MANY HUNDREDS OF FEET AWAY. THE
 369 DATA"AWESOME ROCK WALLS ARE ALMOST SHEER, AS T
 HEY PLUNGE DOWN.#
 370 DATA"*7YOU ARE WEST OF THE *GIANT* ROOM. THE
 372 DATA"CEILING IS LOWER HERE, ALTHOUGH THE VAST
 374 DATA"CHASM STILL LIES TO THE SOUTH. MANY
 376 DATA"OPENINGS CAN BE SEEN ALONG THE PATH TO T
 HE EAST.#
 380 DATA"*8TO THE WEST LIES A DECENT TRACK TO THE
 382 DATA"GIANT ROOM, WHILE TO THE EAST YOU CAN
 384 DATA"DIMLY MAKE OUT A JUMBLE OF TWISTING,
 386 DATA"TURNING PASSAGEWAYS. THE FLOOR LOOKS
 387 DATA"TO BE CRUMBLING AWAY SLIGHTLY, AND YOU
 388 DATA"SENSE THAT ONE FALSE MOVE COULD EASILY L
 EAD TO DISASTER.#

390 DATA"*9YOU HAVE ENTERED A MAZE OF TWISTING,
392 DATA"TURNING PASSAGEWAYS. MYSTERIOUS
394 DATA"TUNNELS HEAD OFF IN ALL DIRECTIONS,
396 DATA"LEADING GOODNESS KNOWS WHERE.#^
400 DATA6
402 DATA"*0YOU ARE LOST IN A TERRIBLE MAZE OF
TWISTING PASSAGEWAYS.#^
500 DATA-1

READY.

100 DATA7
102 DATA"*0YOU ARE IN THE WESTERN SIDE OF THE TWO
PIT ROOM.#
110 DATA"*1YOU ARE AT A JUNCTION OF PATHS. PASSAGE
S
112 DATA"LEAD OFF IN ALL DIRECTIONS, ALTHOUGH IT
114 DATA"LOOKS TO BE A TIGHT FIT TO THE EAST.#
120 DATA"*2YOU ARE IN AN ALCOVE. A SMALL PATH
122 DATA"HEADS BACK TO THE WEST, WHILE AN
124 DATA"EXTREMELY NARROW TUNNEL HEADS EAST.
126 DATA"AN EERIE LIGHT CAN BE SEEN SHINING
128 DATA"AT THE END OF THE TUNNEL.#
130 DATA"*3YOU ARE IN THE DARK ROOM. A TINY CRACK
132 DATA"LEADS BACK TO THE EAST, WHILE AN EQUALLY
134 DATA"TINY CRACK LEADS WEST. THERE IS A STONE
136 DATA"TABLET ON THE WALL WHICH READS
138 DATA"'CONGRATULATIONS ON BRINGING LIGHT INTO T
HE DARK ROOM'..#
140 DATA"*4YOU ARE IN A JUMBLE OF CORRIDORS, WITH
142 DATA"VERY SHARP WALLS. THE LIGHT OF YOUR
144 DATA"TORCH FLICKERS IN A SLIGHT BREEZE.
146 DATA"BETTER BE CAREFUL IT DOESN'T BLOW OUT.#
150 DATA"*5WITH YOUR TORCH BLOWN OUT, YOU ARE IN
152 DATA"SERIOUS TROUBLE. ONE FALSE MOVE AND
154 DATA"YOU'LL PROBABLY FALL INTO A PIT.#
160 DATA"*6YOU ARE IN A JUMBLE OF CORRIDORS, WITH
162 DATA"VERY SHARP WALLS. THE LIGHT OF YOUR
164 DATA"TORCH FLICKERS IN A SLIGHT BREEZE.
166 DATA"BETTER BE CAREFUL IT DOESN'T BLOW OUT.#
170 DATA"*7YOU ARE STUMBLING ALONG OVER A JUMBLE O
FROCKS.#
180 DATA"*8YOU ARE STUMBLING ALONG OVER A JUMBLE O
FROCKS.#
190 DATA"*9YOU ARE IN A MYSTERIOUS ALCOVE. THERE
192 DATA"IS SOMETHING WRITTEN ON THE WALL, BUT IT
194 DATA"SEEMS TO BE WRITTEN IN AN ANCIENT
196 DATA"TONGUE THAT YOU CANNOT TRANSLATE.
198 DATA"SOMETHING ABOUT PLUGS, PLUGH, PLOUGH? I
T'S HARD TO SAY.#^
200 DATAB

202 DATA"*OYOU ARE ON A LOW TRACK THAT SEEMS TO
 204 DATA"TWIST AND TURN BENEATH YOUR VERY FEET.
 206 DATA"THE WALLS ARE QUITE HIGH HERE, AND A
 208 DATA"NOTE HIGH UP ON THE WALL READS
 209 DATA"'ECCENTRICS ONLY'. WHAT CAN IT ALL M
 EAN?#
 210 DATA"*1DEAD END.#
 220 DATA"*2YOU ARE IN FRONT OF AN ARCHWAY CARVED
 222 DATA"OUT OF THE ROCK. SOUTH YOU CAN SEE
 224 DATA"LIGHTS FLASHING, SCURRYING NOISES AS IF
 225 DATA"OF SMALL CREATURES SCAMPERING ABOUT CAN
 226 DATA"BE HEARD, AND IN THE DISTANCE SOMEONE
 227 DATA"SEEMS TO BE SINGING ABOUT GOLD. THE
 228 DATA"FOETID BREATH OF A TROLL CAN BE SMELT
 229 DATA"A LONG WAY OFF, AND YOU CAN HEAR HIS T
 HUMPING FEET.
 230 DATA" ":DATA"YOU ARE IN FRONT OF A VERY STRANG
 E LAND.#
 231 DATA"*3YOU ARE AT A CROSS-ROADS, WITH PATHS
 232 DATA"LEADING OFF IN ALL DIRECTIONS.#
 240 DATA"*4YOU ARE IN A CONFUSION OF TUNNELS AND
 241 DATA"HOLES, PASSAGES AND STAIRWAYS, LIGHTS
 242 DATA"AND SOUND. BE CAREFUL YOU DON'T TRIP A
 ND FALL.#
 250 DATA"*5YOU ARE ON THE EASTERN SIDE OF THIS
 252 DATA"STRANGE AND WONDERFUL LAND.#
 260 DATA"*6YOU ARE IN A CONFUSION OF TUNNELS AND
 261 DATA"HOLES, PASSAGES AND STAIRWAYS, LIGHTS
 262 DATA"AND SOUND. BE CAREFUL YOU DON'T TRIP A
 ND FALL.#
 270 DATA"*7YOU ARE ON THE WESTERN SIDE OF THIS
 272 DATA"STRANGE AND WONDERFUL LAND.#
 280 DATA"*8SHADOWS BOUNCE OFF THE WALLS, MOCKING
 282 DATA"LAUGHTER FILLS THE AIR, AND THE VERY
 284 DATA"WALLS THEMSELVES SEEM TO BE FULL OF
 286 DATA"STARING EYES AND WICKED FACES.#
 290 DATA"*9YOU ARE TO THE WEST OF THE CENTRAL PART
 292 DATA"OF THIS SECTION OF THE CAVES. PATHS
 294 DATA"LEAD IN ALL DIRECTIONS.#^
 300 DATA9
 302 DATA"*OYOU ARE IN THE CENTRE OF THE ECCENTRIC
 304 DATA"AREA. PATHS LEAD OFF IN EVERY DIRECTION,
 306 DATA"ALTHOUGH THEY TWIST AND TURN SO MUCH
 307 DATA"THAT IT IS HARD TO FIND YOUR WAY.#
 310 DATA"*1YOU ARE TO THE EAST OF THE CENTRAL PART
 312 DATA"OF THIS SECTION OF THE CAVES. PATHS
 314 DATA"LEAD IN ALL DIRECTIONS.#
 320 DATA"*2SHADOWS BOUNCE OFF THE WALLS, MOCKING
 322 DATA"LAUGHTER FILLS THE AIR, AND THE VERY
 324 DATA"WALLS THEMSELVES SEEM TO BE FULL OF
 326 DATA"WICKED FACES AND STARING EYES.#

330 DATA"*3YOU ARE SURROUNDED BY TORTURED FISSURES
 332 DATA"OF ROCK. THE TUNNELS SEEM TO GO ON
 334 DATA"FOREVER, AND IT IS HARD TO KNOW WHICH W
 AY TO TURN.#
 340 DATA"*47THE WIND ECHOES EERILY THROUGH THE
 342 DATA"TUNNELS AROUND HERE. LOW, TURNING PATHS
 344 DATA"HEAD OFF IN MOST DIRECTIONS, ALTHOUGH
 346 DATA"THE PATH TO THE WEST SEEMS MOST P
 ROMISING.#
 350 DATA"*5YOUR SENSE OF DIRECTION CONFUSED BY THE
 352 DATA"CONVOLUTED TURNINGS OF THE ROCK, YOU
 354 DATA"CAN NEVERTHELESS SEE WHAT SEEMS TO BE A
 356 DATA"MAJOR PATH TO THE SOUTH. THE TUNNEL OF
 357 DATA"ROCK, ALTHOUGH LOW AND TURNING, SEEMS TO
 358 DATA"OPEN UP SOMEWHAT IN THAT DIRECTION.#
 360 DATA"*67THE WIND ECHOES EERILY THROUGH THE
 362 DATA"TUNNELS AROUND HERE. LOW, TURNING PATHS
 364 DATA"HEAD OFF IN MOST DIRECTIONS, ALTHOUGH
 366 DATA"THE PATH TO THE EAST SEEMS MOST P
 ROMISING.#
 370 DATA"*7YOU ARE HOPELESSLY LOST IN THIS MAZE OF
 372 DATA"ROCK AND RUBBLE. PATHS APPEAR TO HEAD
 374 DATA"NORTH AND EAST, BUT IT'S HARD TO SAY
 376 DATA"WHETHER THESE ARE TRUE DIRECTIONS OR N
 OT.#
 380 DATA"*8YOU ARE AT LAST TO THE SOUTH OF THE
 382 DATA"ECCENTRIC AREA. THE PATH NORTH IS FILLED
 384 DATA"WITH ECHOING VOICES AND BLINDING
 386 DATA"FLASHES OF LIGHT, WHILE TO THE SOUTH A
 387 DATA"NARROW ROCK PATH LEADS OFF TO A DEEP C
 HASM.#
 390 DATA"*9YOU ARE CLOSE BY A DEEP CHASM. A NORTH
 -
 391 DATA"ERLY PATH LEADS AWAY, BUT TO THE SOUTH
 392 DATA"A HAZY MIST OBSCURES YOUR VIEW.#^
 400 DATA10
 402 DATA"*0YOU ARE ON THE NORTHERN SIDE OF AN
 403 DATA"EXTREMELY DEEP CHASM. THE TUNNELS HAVE
 404 DATA"OPENED UP HERE, AND TO EAST AND WEST YOU
 405 DATA"SEE NOTHING BUT A HAZY MIST. TO THE
 406 DATA"SOUTH LIES A RICKETY OLD BRIDGE, LONG
 407 DATA"DISUSED, THAT SEEMS TO HANG BY A MERE T
 HREAD TO THE ROCKFACE.#
 410 DATA"*1YOU ARE ON THE SOUTHERN SIDE OF THE
 412 DATA"CHASM. TO THE EAST A ROUGH TRACK LEADS
 413 DATA"AWAY, TO THE NORTH LIES THE OLD BRIDGE.#
 420 DATA"*2YOU ARE CRAWLING ALONG A LOW EAST-WEST
 CORRIDOR.#
 430 DATA"*3THE PATH TURNS HERE. TO THE SOUTH THE
 432 DATA"CAVES ARE STARTING TO OPEN UP, BUT TO
 434 DATA"THE WEST LIES ONLY A LOW CRAWL.#

440 DATA"*4THE PATH IS FAIRLY WIDE HERE, WITH ROOM
 442 DATA"FOR A NUMBER OF PEOPLE TO PASS QUITE
 444 DATA"EASILY. THE CEILING TOWERS HIGH ABOVE
 446 DATA"YOU, AND TO THE EAST YOU CAN JUST MAKE
 448 DATA"OUT THE BEGINNINGS OF A VERY LARGE CAVERN
 .#
 450 DATA"*5THE MAIN TRACK TURNS AGAIN AND AGAIN,
 451 DATA"AND YOU ARE FORCED TO HEAD EITHER NORTH O
 R WEST.#
 460 DATA"*6YOU ARE IN AN ENORMOUS OPEN CAVE. THE
 462 DATA"VIEW TO THE SOUTH IS STUPENDOUS, WITH
 464 DATA"TOWERING ROCK FISSURES, PINNACLES OF
 466 DATA"STONE, AND ENORMOUS BOULDERS LYING MANY
 468 DATA"HUNDREDS OF FEET BELOW YOU. DOWN BELOW,
 469 DATA"HIDDEN BY MIST, MYSTERIOUS GURLING N
 OISES CAN BE HEARD.#
 470 DATA"*7YOU ARE IN A VAST CAVE. TO THE SOUTH A
 471 DATA"NARROW HOLE LEADS OFF, TO THE EAST YOU
 472 DATA"CAN MAKE OUT A RAPIDLY NARROWING TRACK,
 474 DATA"BUT TO THE NORTH, SOME 100 FEET AWAY,
 475 DATA"YOU CAN MAKE OUT A WINDOW INTO ANOTHER
 476 DATA"ROOM. A SHADOWY FIGURE CAN BE SEEN
 477 DATA"THERE, PEERING BACK AT YOU. BETWEEN
 478 DATA"YOU AND IT IS A VAST CHASM DROPPING OUT
 479 DATA"OF SIGHT TO THE FLOOR MANY HUNDREDS OF F
 EET BELOW.#
 480 DATA"*8YOU ARE IN A LOW, NARROW TUNNEL.#
 490 DATA"*9YOU ARE IN A SMALL HOLE OF A TUNNEL. TO
 491 DATA"THE NORTH LIES THE VAST CAVE, BUT THE
 492 DATA"ONLY OTHER PATH OUT, TO THE EAST, LEADS
 493 DATA"MERELY TO A NARROW, TWISTING TUNNEL.#^
 500 DATA-1

READY.

100 DATA11
 102 DATA"*0YOU ARE IN A LOW TUNNEL. HERE AND THER
 E
 103 DATA"THE ROCK SHOWS SIGNS OF BEING PROPPED
 104 DATA"UP, AS IF IT MAY HAVE FALLEN DOWN IN
 106 DATA"YEARS GONE BY.#
 110 DATA"*1YOU ARE TO THE EAST OF THE GIANT ROOM.
 111 DATA"A NARROW TUNNEL HEADS SOUTH,AND A GROOVE
 112 DATA"DOWN THE MIDDLE OF THE TUNNEL SUGGESTS
 113 DATA"THAT AN ANCIENT STREAM ONCE RAN THIS
 114 DATA"WAY. TO THE EAST THERE SEEMS TO BE A
 115 DATA"BEWILDERING ARRAY OF CAVES AND TUNNELS.
 116 DATA"STALACTITES AND STALAGMITES ARE ALL
 117 DATA"AROUND, AND ALTHOUGH THE INITIAL OPENING
 118 DATA"IS QUITE LARGE, YOU SOON LOSE TRACK OF
 119 DATA"THE CAVES AND PATHS IN THE GLOOM BEYOND.#

120 DATA"*2YOU ARE LOST IN A BEWILDERING ARRAY OF
 TWISTING, TURNING TUNNELS.#
 130 DATA"*3YOU ARE LOST IN A BEWILDERING ARRAY OF
 TWISTING, TURNING TUNNELS.#
 140 DATA"*4YOU ARE LOST IN A BEWILDERING ARRAY OF
 TWISTING, TURNING TUNNELS.#
 150 DATA"*5YOU ARE LOST IN A BEWILDERING ARRAY OF
 TWISTING, TURNING TUNNELS.#
 160 DATA"*6YOU ARE LOST IN A BEWILDERING ARRAY OF
 TWISTING, TURNING TUNNELS.#
 170 DATA"*7YOU ARE LOST IN A BEWILDERING ARRAY OF
 TURNING, TWISTING TUNNELS.#
 180 DATA"*8YOU ARE LOST IN A BEWILDERING ARRAY OF
 TURNING, TWISTING TUNNELS.#
 190 DATA"*9YOU ARE LOST IN A BEWILDERING ARRAY OF
 TURNING, TWISTING TUNNELS.#^
 200 DATA12
 202 DATA"*0YOU ARE LOST IN A BEWILDERING ARRAY OF
 TURNING, TWISTING TUNNELS.#
 210 DATA"*1YOU ARE LOST IN A BEWILDERING ARRAY OF
 TURNING, TWISTING TUNNELS.#
 220 DATA"*2CONGRATULATIONS! YOU ARE AT THE CENTRE
 221 DATA"OF THE TWISTING, TURNING TUNNELS.#
 230 DATA"*3THE PATH SOUTH HEADS DOWN, TOWARDS AN
 231 DATA"OLD RUSTY DOOR PROBABLY USED TO CONTROL
 232 DATA"THE FLOW OF WATER YEARS AGO. NORTH LIEST
 HE GIANT ROOM.#
 240 DATA"*4THE GROOVE IN THE TUNNEL FLOOR IS QUITE
 241 DATA"WIDE HERE, WIDE ENOUGH TO WALK DOWN. SET
 242 DATA"INTO THE GROOVE IS AN OLD RUSTY DOOR.#
 250 DATA"*5YOU ARE TO THE SOUTH OF THE RUSTY DOOR.
 251 DATA"THE GROOVE IN THE FLOOR IS QUITE NARROW
 252 DATA"HERE, BUT THE TUNNEL ITSELF IS QUITE B
 IG.#
 260 DATA"*6YOU STEP INTO THE HARD ROOM. SOLID
 261 DATA"WALLS OF ROCK ARE ON EITHER SIDE OF YOU,
 262 DATA"A HARD ROCK CARPET COVERS THE FLOOR, AND
 263 DATA"ROCK IS EVERYWHERE. YOU CAN GO EITHER N
 ORTH OR WEST.#
 270 DATA"*7TO THE SOUTH A VERY SMALL HOLE LOOKS OU
 T
 271 DATA"ONTO AN ENORMOUS CANYON. A GLOWING
 272 DATA"LIGHT SEEMS TO FILL THE SPACE BEYOND,
 273 DATA"AND LOOKING DOWN YOU CAN SEE THAT THE
 274 DATA"GLOW COMES FROM A VAST VOLCANO, ALMOST
 275 DATA"EXTINCT, BUT STILL SHINING WITH A RED
 276 DATA"HEAT. A LOW TRACK HEADS WEST.#
 280 DATA"*8YOU ARE IN FRONT OF AN EXTREMELY HIGH
 281 DATA"WALL OF SOLID ROCK. THE ROCK IS SO
 282 DATA"SMOOTH THAT YOU CAN SEE NO FOOT OR HAND-
 283 DATA"HOLDS AT ALL. LOOKING UP REVEALS

284 DATA"NOTHING, ALTHOUGH THERE IS THE MEREST
285 DATA"SUGGESTION OF A LEDGE BEYOND THE WALL.
286 DATA" ":DATA"JOHN RYAN THINKS THAT IF YOU JUMP
HIGH
287 DATA"ENOUGH YOU'LL BE ABLE TO MAKE IT.":DATA"
":DATA"HI JOHN!#
290 DATA"*9TO THE EAST LIES A HIGH LEDGE, WHILE TO
291 DATA"NORTH, SOUTH AND WEST SMALL, ROUND
292 DATA"PASSAGES HEAD OFF. THE ONE TO THE WEST
293 DATA"SEEMS MURKIER THAN THE OTHERS.#^
300 DATA-1

READY.

100 DATA13
102 DATA"*OYOU ARE IN A SMALL, ROUND PASSAGE,
104 DATA"WHICH SLOPES ROUND A TIGHT BEND. THERE
106 DATA"IS A VERY DARK AREA TO THE SOUTH-WEST.#
110 DATA"*1YOU ARE IN A SMALL, ROUND PASSAGE,
112 DATA"WHICH TWISTS AND TURNS TO AVOID THE DARKA
REA.#
120 DATA"*2YOU ARE IN A VERY DARK CORRIDOR. THE
122 DATA"WALLS ARE BLACKER THAN BLACK, AND THE
124 DATA"SURROUNDING TUNNELS SEEM QUITE BRIGHT IN
126 DATA"COMPARISON.":DATA" ":DATA"FAR OVERHEAD TH
ERE IS A TINY PIN-PRICK
127 DATA"OF LIGHT, PERHAPS THE OUTSIDE WORLD?
128 DATA"WHATEVER IT IS, YOU CAN'T REACH IT.#
130 DATA"*3YOU HAVE STUMBLD ONTO THE VERY EDGE OF
131 DATA"A VAST PRECIPICE, MUCH TOO DEEP TO
132 DATA"SCRAMBLE DOWN. TO THE NORTH YOU CAN SEE
133 DATA"TOWERING ROCK FISSURES, PINNACLES OF
134 DATA"STONE, AND ENORMOUS BOULDERS LYING MANY
135 DATA"FEET BELOW YOU. DOWN BELOW, HIDDEN BY
136 DATA"MIST, MYSTERIOUS GURLING NOISES CAN BE
137 DATA"HEARD. FAR AWAY YOU CAN MAKE OUT THE
138 DATA"OPENING TO WHAT APPEARS TO BE AN E
NORMOUS CAVE.#
140 DATA"*4YOU ARE IN A TINY CORRIDOR, WITH PATHS
TO NORTH AND EAST.#
150 DATA"*5YOU'RE ON A GENTLY SLOPING WEST-EAST
152 DATA"PASSAGE, FILLED WITH BOULDERS.#
160 DATA"*6YOU ARE IN AN ODDLY-SHAPED ROOM, WITH
162 DATA"LARGE BOULDERS SCATTERED ABOUT. ON THE
164 DATA"WALL TO THE NORTH ARE CARVED THE WORDS
166 DATA"'FEE FIE FOE FUM'. THE CARVING HAS BEEN
167 DATA"MADE BY AN UNSTEADY HAND, AND GOES QUITED
EEP.#
170 DATA"*7YOU ARE IN A LARGE OPEN SPACE, WITH
171 DATA"CAVERNOUS PASSAGES LEADING OFF EVERY-
172 DATA"WHERE.#

180 DATA"*8YOU ARE IN A HIGH ROOM. THE WALLS ARE
 182 DATA"GENTLY WARM TO THE TOUCH, AND IN THE
 184 DATA"DISTANCE YOU CAN HEAR A DULL RUMBLING S
 OUND.#
 190 DATA"*9YOU ARE IN A HIGH ROOM, WHICH IS EVEN
 191 DATA"HIGHER TO THE SOUTH. IN THE DISTANCE
 192 DATA"YOU CAN HEAR A DULL RUMBLING SOUND.#^
 200 DATA14
 202 DATA"*0THE WALLS ARE QUITE WARM HERE. A STEAD
 Y
 204 DATA"ROAR COMES FROM THE NORTH-WEST, SHAKING
 206 DATA"THE WALLS. A PATH HEADS SOUTH, AND A L
 OW CRAWL HEADS WEST.#
 210 DATA"*1THE WALLS ARE QUITE COOL TO THE TOUCH,
 212 DATA"AND YOU CAN REST AGAINST THEM FOR A
 214 DATA"LITTLE WHILE. PATHS LEAD IN ALL D
 IRECTIONS.#
 220 DATA"*2THE DULL RUMBLING SOUND PERSISTS. THE
 222 DATA"AIR IS QUITE CLEAR HERE, AND YOU CAN
 224 DATA"MAKE OUT PATHS TO NORTH, WEST AND EAST.#
 230 DATA"*3A MYSTERIOUS PATH HEADS WEST, WHILE TO
 232 DATA"NORTH AND EAST ARE REGULAR CRAWLS. A
 234 DATA"STEADY STREAM OF MIST IS COMING OUT OF
 236 DATA"THE TUNNEL TO THE WEST, MAKING IT
 238 DATA"DIFFICULT TO SEE INTO, ALTHOUGH IT SEEMST
 O BE QUITE HIGH.#
 240 DATA"*4THE WALLS ARE WARM TO THE TOUCH, AND
 242 DATA"FROM THE NORTH THERE IS A STEADY ROAR OF
 244 DATA"NOISE.#
 250 DATA"*5YOU ARE IN A SMALL CHAMBER FILLED WITH
 252 DATA"BOULDERS. THE WALLS ARE NOW VERY WARM
 254 DATA"INDEED, AND THE AIR IS STIFLING.#
 260 DATA"*6YOU ARE ON THE EDGE OF A BREATHTAKING
 261 DATA"VIEW. FAR BELOW AN ACTIVE VOLCANO
 262 DATA"THUNDERS OUT MOLTEN LAVA WHICH CRASHES
 263 DATA"INTO THE DEPTHS. TO ONE SIDE LIES A
 264 DATA"VAST GORGE, FILLED WITH A BIZARRE CHAOS
 265 DATA"OF TORTURED ROCK. TO THE RIGHT AN
 266 DATA"IMMENSE GEYSER OF BLISTERING STEAM
 267 DATA"ERUPTS CONTINUOUSLY. A RATHER WARM
 268 DATA"PASSAGE EXITS TO THE SOUTH, WHILE TO THE
 269 DATA"WEST A CRUMBLING TRACK OF ROCK LEADS A
 ROUND THE VOLCANO.#
 270 DATA"*7YOU ARE ON THE WESTERN RIM OF THE
 272 DATA"VOLCANO.#
 280 DATA"*8YOU ARE TO THE WEST OF A LOW TUNNEL,
 282 DATA"FILLED WITH MYSTERIOUS SWIRLING MIST.
 284 DATA"ALTHOUGH YOU CAN'T SEE INTO THE TUNNEL
 286 DATA"PROPERLY, IT APPEARS TO BE LOW AND DARK.#
 290 DATA"*9A GHOSTLY SPECTRE SWATHED IN MIST
 291 DATA"BLOCKS YOUR PATH AND YOU CAN GO NO

292 DATA"FURTHER. A MISTY HAND REACHES OUT OF
294 DATA"THE DARKNESS AND COMES TOWARDS YOU ...#^
300 DATA-1

READY.

100 DATA15
102 DATA"*0LOW, RAMBLING CRAWLS LEAD NORTH AND
104 DATA"WEST, AND THE GHOSTLY SPECTRE OF THE
106 DATA"MISTS LIES BEHIND YOU. I WOULDN'T TRY
108 DATA"GOING BACK IF I WERE YOU.#
110 DATA"*1YOU ARE IN A DIRTY CHAMBER, WITH GREAT
112 DATA"CRACKS IN THE FLOOR AND WALLS.#
120 DATA"*2YOU ARE IN A LOW, ROUND CHAMBER. LITTL
E
122 DATA"TRACES OF MIST REMIND YOU OF YOUR
124 DATA"GHOSTLY ENCOUNTER.#
130 DATA"*3YOUR FEET ARE STICKING IN MUD OOZING
132 DATA"OUT FROM THE GROUND BELOW. PERHAPS IT'S
134 DATA"LAVA FROM THE VOLCANO WHICH HAS SOMEHOW
136 DATA"CREPT THROUGH THE ROCKS? #HATEVER IT
138 DATA"IS, IT MAKES MOVEMENT DIFFICULT. PATHS H
EAD SOUTH, WEST AND EAST.#"
140 DATA"*4THE MUD IS QUITE THICK HERE, AND YOU CA
N
142 DATA"ONLY HEAD WEST OR EAST. PROGRESS IS V
ERY SLOW.#
150 DATA"*5MUD COMES OUT OF THE WALLS AND THROUGH
152 DATA"THE FLOOR AND CEILING, COVERING YOU IN A
154 DATA"DISGUSTING SLIMY COATING. IT'S QUITE
156 DATA"HARD TO BREATHE NOW, AND THE LIGHT FROM
158 DATA"YOUR TORCH DOESN'T LOOK AS IF IT WILL
159 DATA"HOLD OUT FOR MUCH LONGER.#
160 DATA"*6YOU ARE AT THE END OF THE TUNNEL OF
162 DATA"MUD. PATHS LEAD SOUTH AND EAST, BACK I
NTO THE MUD AGAIN.#
170 DATA"*7TO THE NORTH LIES THE MUD, BUT TO THE
172 DATA"SOUTH A MAGNIFICENT CAVERN CAN BE SEEN
174 DATA"OPENING UP.#
180 DATA"*8YOU ARE IN A MAGNIFICENT CAVERN. THE
182 DATA"WALLS SHINE WITH A MAGICAL GLOW, AND ALL
184 DATA"AROUND SHADOWS DANCE AND FLICKER IN THE
186 DATA"LIGHT OF YOUR TORCH. STALACTITES HANG
187 DATA"DOWN FROM THE CEILING, DRIP-DRIP-DRIP,
188 DATA"AND AT YOUR FEET ARE SEVERAL LUMINOUS C
OINS.#
190 DATA"*9YOU ARE IN A SOUTH-SIDE CHAMBER, AWAY
192 DATA"FROM THE MAGICAL CAVERN. A ROUGH TRACK
194 DATA"GOES OFF TO THE SOUTH, AND THE TWISTS
196 DATA"AND TURNS IN THE ROCKS ARE ONLY TOO O
BVIOUS.#^

200 DATA16
 202 DATA"*0THE PATH TURNS TO HEAD EITHER NORTH OR
 203 DATA"EAST. A DEEP VOICE REVERBERATES THROUGH
 204 DATA"THE CAVES AND SAYS: 'YOU HAVE NOW GOT
 205 DATA"TO GET THROUGH THE PATH OF MISTS'.
 206 DATA"THE PATH EAST LOOKS COVERED IN MIST.
 207 DATA"7THERE ARE POTHOLES ALL OVER THE PLACE,
 208 DATA"AND YOU CAN ONLY DIMLY MAKE OUT WHERE
 209 DATA"YOU ARE GOING. THE GROUND BENEATH TURNST
 0 A MARSHY BOG AS YOU WALK.#
 210 DATA"*1YOU ARE ON THE PATH OF MISTS. THE CAVE
 GODS WATCH OVER YOU.#
 220 DATA"*2YOU ARE ON THE PATH OF MISTS. THE CAVE
 GODS WATCH OVER YOU.#
 230 DATA"*3YOU ARE ON THE PATH OF MISTS. THE CAVE
 GODS WATCH OVER YOU.#
 240 DATA"*4YOU ARE ON THE PATH OF MISTS. THE CAVE
 GODS WATCH OVER YOU.#
 250 DATA"*5THE GROUND IS SUDDENLY HARD AGAIN, BUT
 252 DATA"THE TRACK TO THE EAST IS STILL VERY
 254 DATA"MISTY. A FAMILIAR HAZY SPECTRE EMERGES
 255 DATA"FROM THE GLOOM, MOANING AND GROANING,
 256 DATA"FEELING HIS WAY TOWARDS YOU. A GHASTLY
 257 DATA"HAND, DRIPPING WITH MUD AND SLIME,
 258 DATA"STRETCHES OUT OF THE MIST AND REACHES F
 ORWARD ...#
 260 DATA"*6YOU ARE IN A VERY HIGH ROOM. THE CEIL-
 262 DATA"ING IS SO HIGH THAT NOT EVEN THE LIGHT
 264 DATA"OF YOUR TORCH CAN PENETRATE THE GLOOM.
 266 DATA"A WIDE TRACK LEADS SOUTH.#
 270 DATA"*7STENCH AND FILTH FILL THE AIR, MAKING I
 T
 272 DATA"HARD TO BREATHE. MYSTERIOUS. SHADOWY
 274 DATA"FIGURES STALK THE LAND, AND EYES PEER AT
 276 DATA"YOU FROM ALL SIDES. DEEP, DANK PATHS
 278 DATA"LEAD OFF TO NORTH AND SOUTH.#
 280 DATA"*8YOU ARE ON A WIDE, CAVERNOUS PATH,
 282 DATA"LEADING ROUGHLY NORTH-SOUTH.#
 290 DATA"*9THE PATH NARROWS, A TREACHEROUS STRETCH
 291 DATA"OF ROCK AND RUBBLE NOW, WITH A VAST
 292 DATA"DROP TO WEST AND EAST INTO A POOL OF
 293 DATA"MURKY, STAGNANT WATER.#^
 300 DATA17
 302 DATA"*0YOU ARE ON A WIDE PATH. FLEETING
 303 DATA"SHADOWS BRUSH BY YOUR FACE, HANDS APPEAR
 304 DATA"TO REACH OUT OF THE GLOOM AND CLUTCH AT
 305 DATA"YOU, THE VERY AIR ITSELF IS FULL OF
 306 DATA"DANGER. YOU CAN MAKE OUT A NARROW TRACK
 307 DATA"HEADING NORTH, A LOW, THREATENING HOLE
 308 DATA"LEADING EAST, AND A NARROW ARCH OF ROCK
 309 DATA"TO THE SOUTH.#

310 DATA"*1YOU ARE IN A LOW, ROUND, TUNNEL OF ROCK
 .#
 320 DATA"*2DEAD END.#
 330 DATA"*3YOU ARE UNDERNEATH A HIGH ARCH OF ROCK.
 #
 340 DATA"*4YOU ARE TO THE SOUTH OF THE ARCHWAY.
 342 DATA"A DECENT TRACK LIES TO THE SOUTH, BUT
 344 DATA"THE WAY BEYOND IS LOST IN SHINING LIGHTS
 346 DATA"AND DAZZLING LUMINANT ROCKS.#
 350 DATA"*5YOU ARE AT A DOORWAY INTO A MAGICAL,
 352 DATA"MYSTICAL LAND, OF UNDERGROUND PLANTS AND
 354 DATA"FLOWERS, TOWERING ROCKS AND TINY HOLES
 355 DATA"IN THE GROUND. WATERFALLS WILL FLOW,
 356 DATA"CASCADING TO THE GROUND, UNDERGROUND
 357 DATA"STREAMS AND RIVERS WILL BLOCK THE WAY,
 358 DATA"CREATURES OF THE CAVES WILL EMERGE, AND
 359 DATA"REALLY, JUST ABOUT ANYTHING MIGHT H
 APPEN.#
 360 DATA"*6YOU ARE IN THE MAGICAL, MYSTICAL, LAND,
 362 DATA"WHERE SHADOWS DANCE, ROCKS FALL, AND
 364 DATA"WATER FLOWS.#
 370 DATA"*7YOU ARE UNDERNEATH A GIANT WATERFALL,
 372 DATA"WHICH CASCADES OUT OF SIGHT TO THE
 374 DATA"GROUND MANY HUNDREDS OF FEET BELOW. A
 375 DATA"VERITABLE PINNACLE OF ROCK THIS, THE
 376 DATA"ONLY SAFE PATHS YOU CAN TAKE ARE TO WESTA
 ND SOUTH.#
 380 DATA"*8YOU ARE IN THE MAGICAL, MYSTICAL, LAND,
 382 DATA"WHERE SHADOWS DANCE, ROCKS FALL, AND
 384 DATA"WATER FLOWS.#
 390 DATA"*9YOU ARE ON A ROCKY OUTPOST. SAFE PATHS
 391 DATA"LEAD EAST AND SOUTH, BUT TO THE WEST,
 392 DATA"ACROSS A GAP IN THE ROCKS SO VAST THAT
 393 DATA"NOT EVEN A GIANT COULD JUMP ACROSS, YOU
 394 DATA"CAN GLIMPSE INTO A SECRET, SHADOWY CAVE.
 395 DATA"7HEREIN LIES THE SECRET OF THE CAVES,
 396 DATA"BUT THE PATH YOU HAVE TO FOLLOW IS LONG A
 ND TREACHEROUS.#^
 400 DATA-1

READY.

100 DATA18
 102 DATA"*0THIS MAGICAL, MYSTERY LAND SURE IS
 LARGE, ISN'T IT?#"

110 DATA"*1YOU ARE NEAR THE CENTRE OF THE LAND.#
 120 DATA"*2YOU ARE AT THE CENTRE OF THE MAGICAL
 122 DATA"MYSTERY LAND. PATHS LEAD OFF IN MANY
 124 DATA"DIRECTIONS, BUT I DON'T SUPPOSE THAT ANY
 126 DATA"OF THEM WILL GO IN THE DIRECTION THAT Y
 OU EXPECT.#

130 DATA"*3YOU ARE NEAR THE CENTRE OF THE LAND.#
 140 DATA"*4THIS MAGICAL, MYSTERY LAND SURE IS
 LARGE, ISN'T IT?#"

150 DATA"*5YOU ARE AT THE SOUTH-WEST CORNER OF THE
 152 DATA"LAND. AT YOUR FEET, IN A LITTLE TRENCH,
 154 DATA"YOU CAN SEE HUNDREDS UPON HUNDREDS OF
 156 DATA"TINY BLUE ROCK PLANTS, WHILE THE WALLS
 157 DATA"AROUND YOU ARE COVERED IN A DEEP BLUE
 158 DATA"MOSS, SOFT TO THE TOUCH.#
 160 DATA"*6THIS LAND GOES ON FOREVER.#
 170 DATA"*7YOU ARE NEAR THE CENTRE OF THE MAGICAL,
 MYSTICAL LAND.#

180 DATA"*8YOU ARE IN A LARGE, OPEN ROOM. IN
 182 DATA"MAGICAL, MYSTICAL LAND. STRANGE LITTLE
 184 DATA"BLUE ROCK CREATURES SCUTTLE ABOUT AT
 185 DATA"YOUR FEET, IMPEDING YOUR PROGRESS AND
 186 DATA"MAKING LITTLE CHIRRUPING NOISES AS THEY
 187 DATA"DO SO. THEY MOVE TOO QUICKLY TO CATCH,
 188 DATA"AND VANISH INTO THE TINNIEST OF CRACKS AS Y
 OU TRY TO GRAB THEM.#

190 DATA"*9YOU ARE IN THE SOUTH-EAST CORNER OF THE
 191 DATA"MAGICAL, MYSTICAL LAND. THERE IS AN
 192 DATA"EXTREMELY DEEP DROP TO A LEDGE, SOME 20
 193 DATA"OR SO FEET BELOW YOU, AND YOU CAN SEE
 194 DATA"THAT FROM THE LEDGE A DARK PATH GOES OFF
 195 DATA"TO THE SOUTH. THERE IS A SMELL OF SALT
 196 DATA"WATER IN THE AIR, ALTHOUGH THERE IS NO
 197 DATA"APPARENT SIGHT OF WATER.":DATA" "
 198 DATA"YOU COULD TRY JUMPING DOWN ONTO THE
 199 DATA"LEDGE, BUT YOU'D PROBABLY JUST BREAK A L
 EG OR SOMETHING.#^

200 DATA19
 202 DATA"*0YOU ARE ON THE DEEP LEDGE. THE PATH
 204 DATA"CONTINUES SOME 20 FEET ABOVE YOU, WHILE
 206 DATA"TO THE SOUTH YOU CAN SEE A DARK TUNNEL
 208 DATA"BUILT INTO THE ROCK.#
 210 DATA"*1YOU ARE IN A LONG, LOW, DARK TUNNEL. 7
 0
 212 DATA"THE EAST THERE IS A GARISH LIGHT, AND
 214 DATA"A FEW WOODEN TABLES AND CHAIRS CAN BE
 216 DATA"MADE OUT.#
 220 DATA"*2YOU ARE IN AN UNDERGROUND BAR, SEEMINGL
 Y
 222 DATA"DESERTED. EMPTY WOODEN TABLES AND
 224 DATA"CHAIRS LIE ABOUT THE PLACE, A FEW MUGS
 226 DATA"WITH LONG DEAD DRINKS LURK IN THE
 227 DATA"CORNERS OF THE BAR, AND IT LOOKS LIKE
 228 DATA"THE DRINKING EQUIVALENT OF THE MARIE C
 ELESTE.#
 230 DATA"*3YOU ARE NOW BEHIND THE UNDERGROUND BAR.
 232 DATA"A LONG, STRAIGHT TUNNEL HEADS OFF TO THE

234 DATA"NORTH, WHILE TO THE EAST A SHIMMERING
 235 DATA"HAZE OBSCURES MOST OF THE VIEW. THE
 236 DATA"AIR NOW SMELLS QUITE STRONGLY OF SEA
 237 DATA"WATER, AND YOU CAN HEAR THE CRIES OF
 238 DATA"LARGE WATER-BIRDS AS THEY SWOOP AND
 239 DATA"DIVE. AT LEAST, YOU HOPE THAT THEY'RE W
 ATER BIRDS.#
 240 DATA"*4YOU ARE IN A STRAIGHT TUNNEL HEADING
 242 DATA"ROUGHLY NORTH-SOUTH. THE WALLS ARE
 244 DATA"QUITE JAGGED, AND THE PATH APPEARS TO
 246 DATA"STEEPEN SLIGHTLY TO THE NORTH.#
 250 DATA"*5YOU ARE IN A LOW CRAWL OVER COBBLES AND
 252 DATA"DEBRIS, GETTING STEEPER TO THE NORTH.#
 260 DATA"*6THE CEILING IS NOW QUITE HIGH, AND IT
 262 DATA"LOOKS AS IF THE END OF THE WORLD LIES TO
 263 DATA"THE NORTH. IN THE DIM LIGHT YOU CAN'T
 264 DATA"MAKE OUT ANYTHING OTHER THAN A VAST E
 MPTY SPACE.#
 270 DATA"*7YOU ARE LOOKING DOWN INTO AN ENORMOUS
 272 DATA"CANYON THAT HOUSES AN ALMOST-EXTINCT
 274 DATA"VOLCANO. THE VOLCANO STILL GLOWS WITH
 276 DATA"A LOW RED HEAT, AND BEYOND IT YOU CAN
 277 DATA"MAKE OUT A LITTLE HOLE IN THE ROCK
 278 DATA"FACE EXACTLY OPPOSITE YOU AND MANY, MANY
 279 DATA"FEET AWAY. THE ONLY EXIT IS BACK TO THES
 OUTH.#
 280 DATA"*8YOU'RE IN A COLD ROOM. THE AIR IS FRES
 H
 281 DATA"AND CRISP HERE, AND YOU CAN SCARCELY
 282 DATA"BEAR TO TOUCH THE WALLS, SO COLD ARE
 283 DATA"THEY. CARVED INTO THE WALLS IS AN ICY
 284 DATA"MESSAGE, WHICH READS 'YME'.#
 290 DATA"*9THE WALLS ARE COLD TO THE TOUCH, AND YO
 U
 291 DATA"SHIVER SLIGHTLY AS YOU WALK ALONG THIS
 292 DATA"ICE-WHITE CORRIDOR.#^
 300 DATA-1

READY.

100 DATA20
 102 DATA"*0THERE IS A WHITE SHEEN TO THE WALLS OF
 104 DATA"THIS LOW, ROUND CORRIDOR. THE AIR IS
 106 DATA"FRESH, AND PLEASANT TO BREATHE, ALBEIT A
 108 DATA"LITTLE CHILL.#
 110 DATA"*1THE SMELL OF SALTY SEA WATER FILLS THE
 112 DATA"AIR, AND YOU CAN HEAR THE LAPPING OF
 114 DATA"WAVES ON A DISTANT SHORE. THE CAVE
 116 DATA"OPENS UP TO THE EAST, WHILE THE PATH
 117 DATA"BACK LEADS ONLY TO THE BAR. 70 THE NORTH
 118 DATA"IS A CURIOUS ROUND HOLE.#

120 DATA"*2THE ROUTE NORTH IS BLOCKED BY A HUGE,
 122 DATA"TOWERING CLIFF FACE, POCK-MARKED WITH
 124 DATA"LEDGES ON WHICH BIRDS RAISE THEIR YOUNG,
 125 DATA"FAR OUT OF YOUR REACH. THE ONLY PATH
 126 DATA"OUT IS BACK TO THE SOUTH, AS EAST WILL
 127 DATA"ONLY TAKE YOU TO A 50 FOOT DROP INTO THEW
 ATER.#
 130 DATA"*3YOU ARE ON QUITE A HIGH LEDGE OVER-
 132 DATA"LOOKING THE LAKE, BUT STILL YOU CANNOT
 134 DATA"SEE TO THE OTHER SIDE. THE PATH
 136 DATA"CONTINUES TO THE NORTH.#
 140 DATA"*4YOU ARE ON A LOW LEDGE, WITH THE LAKE
 142 DATA"TO THE EAST AND A SHINGLE PATH TO THE
 144 DATA"NORTH.#
 150 DATA"*5YOU ARE ON THE SHORE OF AN ENORMOUS
 151 DATA"UNDERGROUND LAKE, SO VAST THAT YOU
 152 DATA"CANNOT SEE TO THE OTHER SIDE. FAR OVER-
 153 DATA"HEAD THE CEILING OF THE CAVE ALMOST
 154 DATA"DISAPPEARS FROM VIEW. WAVES LAP GENTLY
 155 DATA"ON THE SHINGLY SHORE, AND BIRDS SWOOP
 156 DATA"AND DIVE IN SEARCH OF FOOD. THE SURFACE
 157 DATA"OF THE LAKE IS QUITE SMOOTH, BUT WHO
 158 DATA"KNOWS WHAT UNSEEN MONSTERS LURK IN THE
 159 DATA"DEPTHS?":DATA" ":DATA"A SHINGLE PATH HEAD
 S NORTH AROUND THE LAKE.#
 160 DATA"*6YOU ARE ON THE SURFACE OF THE LAKE,WITH
 162 DATA"A SIGHT OF LAND TO THE WEST. IN ALL
 164 DATA"OTHER DIRECTIONS THERE IS NOTHING BUT
 166 DATA"A VAST EXPANSE OF WATER.#
 170 DATA"*7YOU ARE ON THE SURFACE OF THE LAKE,WITH
 172 DATA"A SIGHT OF LAND TO THE NORTH. IN ALL
 174 DATA"OTHER DIRECTIONS THERE IS NOTHING BUT A
 176 DATA"VAST EXPANSE OF WATER.#
 180 DATA"*8YOU ARE LOST ON THE SURFACE OF THE LAKE
 .#
 190 DATA"*9YOU ARE LOST ON THE SURFACE OF THE LAKE
 .#^
 200 DATA21
 202 DATA"*0YOU ARE LOST ON THE SURFACE OF THE LAKE
 .#
 210 DATA"*1YOU ARE LOST ON THE SURFACE OF THE LAKE
 .#
 220 DATA"*2YOU ARE LOST ON THE SURFACE OF THE LAKE
 .#
 230 DATA"*3YOU ARE STILL ON THE SURFACE OF THE
 232 DATA"LAKE, AND TO YOUR JOY YOU CAN MAKE OUT A
 234 DATA"SIGHT OF LAND ON THE DISTANT HORIZON.
 235 DATA"ROCKY CRAGS SEEM TO LEAP OUT OF THE
 236 DATA"WATER, BUT SO CONFUSED IS YOUR SENSE OF
 237 DATA"DIRECTION THAT YOU NO LONGER KNOW WHICH
 238 DATA"ROUTE TO TAKE. IS IT THE SHORE YOU

239 DATA"LEFT, OR IS IT THE OTHER SIDE OF THIS I
MMENSE STRETCH OF WATER?#
240 DATA"*4YOU ARE ON THE SURFACE OF THE LAKE,WITH
241 DATA"A SIGHT OF LAND TO THE SOUTH. IN ALL
242 DATA"OTHER DIRECTIONS THERE IS NOTHING BUT A
244 DATA"VAST EXPANSE OF WATER.#
250 DATA"*5YOU ARE ON THE SOUTHERN SHORE OF THE
252 DATA"LAKE. TO THE SOUTH IS A HIGH ROCK WALL,
254 DATA"DOTTED WITH PITS AND CREVICES, BUT SINCE
255 DATA"YOU CAN'T SEE TO THE TOP IT COULD BE A
256 DATA"BAD MOVE TO ATTEMPT TO CLIMB IT. STILL,
257 DATA"THERE'S NOT MUCH ELSE TO DO. EVERYWHERE
258 DATA"ELSE YOU LOOK YOU CAN ONLY SEE WATER.#
260 DATA"*6YOU ARE ON TOP OF THE LEDGE, A LARGE
262 DATA"PLATEAU OVERLOOKING THE LAKE TO THE
263 DATA"NORTH. THERE ARE A NUMBER OF CRUMBLY
264 DATA"PATHS LEADING OFF, BUT THE ONLY SAFE ONE
265 DATA"IS TO THE WEST, WHERE YOU CAN SEE A
266 DATA"PROMISING TRACK LEADING TO A CAVE
267 DATA"WITHIN A CAVE.#
270 DATA"*7YOU ARE IN A SMALL, DARK CAVE HEADING
272 DATA"ROUGHLY EAST-WEST.#
280 DATA"*8YOU ARE STUMBLING OVER ROCKY GROUND,
282 DATA"STILL HEADING EAST-WEST.#
290 DATA"*9YOU ARE AT THE END OF A SMALL CAVE.
292 DATA"THE TRACK HEADS EAST-WEST, AND TO THE
294 DATA"WEST YOU CAN SEE A CURIOUS STRUCTURE
296 DATA"CARVED OUT OF THE ROCK FACE.#^
300 DATA-1

READY.

100 DATA22
102 DATA"*0YOU CAN ONLY GO EAST OR WEST HERE. A
103 DATA"NARROW PATH LEADS TO THE EAST, AND WEST
104 DATA"LIES A CURIOUS SELECTION OF WELL-TRODDEN
105 DATA"PATHS. IT WOULD APPEAR THAT SOMEONE
106 DATA"HAS PLANTED AN UNDERGROUND GARDEN, BUT
107 DATA"ELECTED TO USE ASTROTURF INSTEAD OF
108 DATA"GRASS. DO YOU PREFER GRASS OR ASTROTURF?
109 DATA"PERSONALLY I DON'T KNOW, NEVER HAVING
110 DATA"SMOKED ASTROTURF, BUT YOU CAN FIND OUT
111 DATA"BY GOING WEST.":DATA" ":DATA"CARVED INTO
THE ROCK TO YOUR RIGHT IS AN
112 DATA"INTERESTINGLY FAMILIAR FACE. LOOKING
113 DATA"A BIT LIKE TONY BRIDGES, EVERYTIME YOU
114 DATA"MOVE TOWARDS IT A HAND STRETCHES OUT OF
115 DATA"THE ROCK AND A HOLLOW VOICE REMINDS YOU
116 DATA"THAT YOU HAVE NOT PAID YOUR SUBSCRIPTION
117 DATA"TO POPULAR COMPUTING WEEKLY.#
118 DATA"*1YOU ARE WALKING ON A SPRINGY SURFACE OF

119 DATA"ASTROTURF. #
120 DATA"*2YOU ARE IN A WELL-KEPT ARTIFICIAL
122 DATA"GARDEN. PLANTS ARE GROWING IN
124 DATA"ABUNDANCE, LIVING AT YOUR FEET. THE
125 DATA"WALLS ARE QUITE HIGH HERE, AND THEY TOO
126 DATA"SEEM COVERED IN SOME SORT OF MOSSY
127 DATA"GROWTH. INSECTS FLIT AND BUZZ ABOUT,
128 DATA"GENERALLY GETTING IN THE WAY. #"
130 DATA"*3YOU ARE WALKING ABOUT IN THE ASTROTURF
GARDEN. #
140 DATA"*4THE SMALL, WELL-TENDED GARDENS HERE-
142 DATA"ABOUT WOULD INDICATE THAT SOMEONE HAS TO
144 DATA"LOOK AFTER THEM. NEAT ROWS OF 4 BY 4
146 DATA"BEDS OF PLANTS ARE ALL AROUND. TRY NOT T
O TREAD ON THEM. #
150 DATA"*5WHAT A CURIOUS PLACE THIS IS. WATER
152 DATA"DRIPS DOWN FROM THE CEILING, KEEPING
154 DATA"EVERYTHING NICE AND DAMP, AND A SMALL
155 DATA"POOL AT YOUR FEET HOUSES A COLLECTION OF
156 DATA"HITHERTO UNKNOWN WATER CREATURES. THE
157 DATA"WATER ITSELF FLOWS OUT OF THE POOL TO
158 DATA"THE WEST, CASCADING OVER A ROCK FACE AND
159 DATA"FALLING MANY HUNDREDS OF FEET TO THE
160 DATA"BOTTOM OF A DEEP, DEEP PIT. ACROSS THE
161 DATA"PIT AND ABOUT 20 FEET AWAY IS A DARK
162 DATA"PATH WHICH HEADS NORTH-SOUTH, BUT IT
163 DATA"LOOKS LIKE A MIGHTY BIG JUMP TO ME. #
165 DATA"*6YOU ARE STILL IN ASTROTURF LAND. #
170 DATA"*7YOU ARE FLOATING OVER THE BOTTOMLESS
PIT. #
180 DATA"*8YOU ARE ON THE WEST SIDE OF THE PIT.
182 DATA"DARK, NARROW PATHS HEAD NORTH AND SOUTH,
184 DATA"ALTHOUGH THE SOUTHERN ONE LOOKS A BIT M
ORE PROMISING. #
190 DATA"*9YOU SHOULD KNOW BETTER. DEAD END. #^
200 DATA23
202 DATA"*0THE CORRIDOR TO THE NORTH SEEMS DARK,
204 DATA"AND LOOKS AS IF IT TURNS SHARPLY.
206 DATA"BACK TO THE SOUTH IS THE BOTTOMLESS PIT, S
O. WHAT TO DO? #
210 DATA"*170 YOUR WEST LIES A RICKETY OLD BRIDGE
211 DATA"OVER A VERY DEEP DROP, COUPLED WITH A
212 DATA"STRANGE FEELING OF DEJA VU. SOUTH HEADS
214 DATA"BACK TO THE OLD PIT AGAIN, AND ALL OTHER
216 DATA"OPTIONS ARE CLOSED TO YOU BY THE
218 DATA"SOLID WALL OF ROCK SURROUNDING YOU ON A
LL OTHER SIDES. #
220 DATA"*2YOU ARE IN FRONT OF A RICKETY BRIDGE
221 DATA"HEADING WEST. THE STRANDS OF ROPE
222 DATA"HOLDING THE BRIDGE TO THE ROCK FACE LOOK
223 DATA"OLD AND WITHERED, AND THE BATTERED

224 DATA"PIECES OF WOOD WHICH CONSTITUTE THE
 225 DATA"BRIDGE DO NOT INSPIRE CONFIDENCE. ONE
 226 DATA"FALSE MOVE AND ...#
 230 DATA"*3YOU ARE ON THE WEST SIDE OF THE BRIDGE,
 231 DATA"AND STANDING ON THE FIRST OF NINE LEVELS
 232 DATA"OF ROCK FACE, WHICH HEAD UP AND DOWN IN
 233 DATA"THE MANNER OF AN ESCHER PAINTING. A
 234 DATA"DEEP AND RESONANT VOICE BOOMS OUT OF THE
 235 DATA"DARKNESS, AND TELLS YOU THAT 'THE ROCKS
 236 DATA"MUST BE TRAVERSED IN ORDER!'. MEAN OLD
 237 DATA"SOUL. THIS IS THE LORD OF THE CAVES,
 238 DATA"WHO WATCHES OVER YOUR EVERY STEP.";DATA"
 "

239 DATA"OBVIOUS PATHS HEAD NORTH, EAST AND WEST.#
 240 DATA"*4YOU ARE ON THE THIRD LEVEL.#
 250 DATA"*5YOU ARE ON THE SECOND LEVEL.#
 260 DATA"*6YOU ARE ON THE FIFTH LEVEL. ALL WELL
 AND GOOD SO FAR.#
 270 DATA"*7YOU ARE ON THE FOURTH LEVEL. I'D HEAD
 SOUTH IF I WERE YOU.#
 280 DATA"*8YOU HAVE REACHED THE SIXTH LEVEL. DON'
 T
 281 DATA"WORRY, YOU WON'T HAVE TO RETREAT THROUGHT
 HIS AREA.#
 290 DATA"*9YOU ARE ON THE SEVENTH LEVEL. PATHS
 291 DATA"LEAD TO NORTH AND EAST, BUT I WOULDN'T
 292 DATA"TRY GOING EAST.#^
 300 DATA24
 302 DATA"*0YOU ARE ON LEVEL EIGHT. NOW YOU CAN
 303 DATA"HEAD EITHER NORTH, SOUTH OR EAST. WHAT
 304 DATA"WOULD YOU GIVE FOR THE MAGIC WORD WHICH
 306 DATA"MIGHT TRANSPORT YOU THROUGH THIS LOT?#
 310 DATA"*1YOU HAVE REACHED LEVEL NINE. PATHS HEA
 D
 311 DATA"EAST, WEST OR SOUTH. HAVING GOT THIS
 312 DATA"FAR, I WOULDN'T WANT YOU TO BLOW IT NOW.
 313 DATA"SUFFICE IT TO SAY, BE VERY, VERY
 314 DATA"CAREFUL, AND WATCH OUT FOR THE PIT OF
 315 DATA"MONSTERS TO THE WEST (OOPS, WHAT A G
 IVEAWAY).#
 320 DATA"*2YOU ARE IN FRONT OF A POOL OF WATER,
 321 DATA"WHICH THRASHES ABOUT IN A WILD FRENZY.
 322 DATA"EVERY NOW AND AGAIN YOU CATCH A GLIMPSE
 323 DATA"OF SOME FEROCIOUS CREATURE, TEETH
 324 DATA"FLASHING IN THE SPARKLING LIQUID.";DATA"
 "

325 DATA"BEHIND YOU LIE THE NINE LEVELS, AHEAD,
 326 DATA"ACROSS THE 30 FEET OF WATER, YOU CAN
 327 DATA"SEE A NARROW, TWISTING, TURNING
 328 DATA"CORRIDOR. YOUR GOAL IS ALMOST IN SIGHT,
 329 DATA"AMONGST THE TWISTING DEPTHS OF THE

330 DATA"CORRIDOR. HOWEVER, THERE IS THE SMALL P
 ROBLEM OF THESE CREATURES ...#
 331 DATA"*3YOU ARE TO THE NORTH OF THE POOL OF
 332 DATA"MONSTERS. THE PATH TWISTS TO WEST AND
 333 DATA"SOUTH, AND A SLIGHT BREEZE CAN BE FELT
 334 DATA"BLOWING DOWN THE CORRIDOR. THE ROCK
 335 DATA"WALLS ARE QUITE COLD TO THE TOUCH. AS IF
 336 DATA"THE WIND OF AGES HAS COOLED AND
 337 DATA"WEATHERED THEM.#
 340 DATA"*4YOU ARE AT THE BOTTOM OF A CORRIDOR
 341 DATA"WHICH RISES TO THE NORTH. A BREEZE IS
 342 DATA"BLOWING DOWN THE CORRIDOR, AND YOUR
 343 DATA"TORCH FLICKERS AND CASTS EERIE, DANCING
 344 DATA"SHADOWS ON THE WALLS. A FEW SMALL
 345 DATA"BOULDERS LIE ABOUT YOUR FEET, THE WIND
 346 DATA"ECHOES AND SIGHS, THE PATH BECKONS YOU N
 ORTHWARD ...#
 350 DATA"*5YOU ARE IN A LOW, ROUND, NORTH-SOUTH
 351 DATA"CORRIDOR. THERE IS A STRONG BREEZE
 352 DATA"BLOWING FROM THE NORTH, AND A STRANGE,
 353 DATA"UNEARTHLY LIGHT CAN BE SEEN, PERHAPS THE
 354 DATA"SOURCE OF THE BREEZE? SOUTH LEADS ONLY
 355 DATA"TO THE POOL OF MONSTERS, AND HAVING
 356 DATA"DISPOSED OF YOUR COMPANION, THEY MAY BE
 357 DATA"GETTING HUNGRY AGAIN. YOU MUST HEAD N
 ORTH ...#
 360 DATA"*6YOU ARE IN THE TEETH OF A GALE. YOUR
 361 DATA"TORCH FLICKERS AND DANCES ABOUT
 362 DATA"ALARMINGLY, AND YOU CAN BARELY SEE YOUR
 363 DATA"WAY FORWARD AS THE DUST AND GRIT OF A
 364 DATA"THOUSAND YEARS BLOWS INTO YOUR STINGING
 365 DATA"EYES.#
 370 DATA"*7YOUR TORCH IS SNATCHED OUT OF YOUR HAND
 371 DATA"BY THE ROARING GALE, AND YOU CAN ONLY
 372 DATA"FEEL YOUR WAY SLOWLY OVER THE MANY ROCKS
 373 DATA"AND BOULDERS THAT LITTER THE FLOOR OF
 374 DATA"THE TUNNEL. THE GRIT AND DUST BEGIN TO
 375 DATA"BLOCK YOUR THROAT, YOU START TO COUGH
 376 DATA"ALARMINGLY, AND YOU SENSE THAT YOU HAVE
 377 DATA"ONLY A FEW MINUTES LEFT TO GET OUT OF H
 ERE ALIVE ...#
 380 DATA"*8YOU HAVE BUMPED INTO A COLLECTION OF
 381 DATA"ROUGH BRACKEN AND MOSS. BENEATH THE
 382 DATA"MOSS YOU CAN FEEL A DOOR, A HANDLE, AND
 383 DATA"A SMALL KEYHOLE. ON ALL OTHER SIDES IS
 384 DATA"NOTHING BUT THE FEROCIOUS HURRICANE NOW
 385 DATA"COMING FROM OVERHEAD. A LIGHT SHINES
 386 DATA"ABOVE YOU, ENABLING YOU TO SEE, BUT AS
 387 DATA"THE GRIT THICKENS IN YOUR THROAT, YOU
 388 DATA"KNOW THAT YOU HAVE JUST SECONDS LEFT TO E
 SCAPE ...#

```

390 DATA"*9YOU ARE IN THE TINIEST OF TINY ROOMS.
391 DATA"THE WIND HAS STOPPED, ALL IS CALM, AND A
392 DATA"SHIMMERING LIGHT ETCHED INTO THE CEILING
393 DATA"REVEALS THE WORD 'AUSTIN'. THE SOLUTION
394 DATA"TO THE NINE LEVELS? THE ONLY PATH OUT H
EADS EAST TO THE OLD DOOR.#^
400 DATA25
402 DATA"*0YOU ARE IN THE SECRET, SHADOWY CAVE
403 DATA"THAT YOU GLIMPSED LONG AGO.":DATA" "
404 DATA"70 THE EAST, ACROSS A VAST GAP IN THE
405 DATA"ROCKS, LIES THE SAFETY OF A ROCKY
406 DATA"OUTPOST, BUT THE WAY BACK IS LONG AND
407 DATA"TREACHEROUS. A PATH LEADS SOUTH, BUT O
NLY TO THE OLD DOOR ...#
410 DATA"*1YOU ARE AT THE NORTH-EAST END OF AN
411 DATA"IMMENSE ROOM WHICH APPEARS TO BE A
412 DATA"REPOSITORY FOR THE 'NOT THE COLOSSAL
413 DATA"CAVE' PROGRAM. MASSIVE TORCHES FAR
414 DATA"OVERHEAD BATHE EVERYTHING IN AN EERIE
415 DATA"LIGHT. EMPTY BOTTLES, YOUNG BEANSTALKS,
416 DATA"OYSTERS, BLACK RODS, BRASS LANTERNS, ALL
417 DATA"LITTER THE FLOOR. MANY DWARVES ARE
418 DATA"SNORING LOUDLY, THE SOUND OF THEIR
419 DATA"SNOOZING BOUNCING OFF THE WALLS.":DATA" "
420 DATA"AN IMMENSE MIRROR HANGS AGAINST ONE
421 DATA"WALL, AND STRETCHES OFF INTO THE D
ISTANCE.#
422 DATA"*2YOU ARE AT THE SOUTH-WEST END OF THE
423 DATA"REPOSITORY. YOU CAN SEE A PIT FULL OF
424 DATA"SNAKES, A ROW OF SMALL CAGES FULL OF
425 DATA"LITTLE SULKING BIRDS, BLACK RODS AND
426 DATA"VELVET PILLOWS ABOUND. A VAST MIRROR
427 DATA"STRETCHES OFF TO THE NORTH-EAST.":DATA" "
428 DATA"AT YOUR FEET IS A LARGE STEEL GRATE, AND
429 DATA"ABOVE IT A SIGN, WHICH READS:":DATA" "
430 DATA"'TREASURE VAULT. KEYS IN MAIN OFFICE'.":
DATA" "
432 DATA"OH WOW, YOU'VE FOUND THE ENDGAME ... N
OW WHAT DO YOU DO?#^
433 DATA-1

```

READY.

Well, that was a marathon exercise, wasn't it? Still, not much further to go now, as we come to all the messages used in the program.

Messages

A wide variety of messages and responses is vital to an adventure

game, which is why the next few pages concern themselves with giving you a mixture of humorous and serious replies to various situations.

These message programs take exactly the same file-making program as the room descriptions, so as ever load in that one first and then type in the following lines:

```
99 DATA26
100 DATA"*0CAN'T YOU TYPE SOMETHING IN RIGHT FOR A
    CHANGE?#\"
101 DATA"*1YOU JUST KEEP GETTING THINGS WRONG,
    DON'T YOU?#\"
102 DATA"*2YOU'LL NEVER SOLVE THE GAME AT THIS
    RATE.#\"
103 DATA"*3YOU KNOW, THAT IS NOT THE BEST IDEA YOU
    HAVE EVER HAD.#\"
104 DATA"*4EDDY IS A MYSTERIOUS PERSON.  HE STARTE
    D\"
105 DATA"OFF LIFE AS A FAILED COLLEGE GRADUATE,\"
106 DATA"BECAME A MILKMAN, LIKE, AND THEN DECIDED
107 DATA"TO BECOME AN UNDERGROUND CULT FIGURE.
108 DATA" \"
109 DATA"HE NOW JUST BLUNDERS ABOUT THE CAVES\"
110 DATA"TRYING TO BE HELPFUL.  BUT HE'S NOT THAT\"
111 DATA"HELPFUL REALLY.#\"
112 DATA"*5IT IS NOW PITCH DARK.  YOU CAN'T FIND\"
113 DATA"YOUR WAY IN THE DARK, AND ONE FALSE
114 DATA"MOVE WILL PROBABLY MAKE YOU FALL HEAD-\"
115 DATA"FIRST INTO A PIT.#\"
116 DATA"*6THERE'S NO POINT LOOKING FOR ANYTHING I
    N\"
117 DATA"THE DARK, SO I WOULDN'T BOTHER IF I WEREY
    OU.#\"
118 DATA"*7OF ALL THE THINGS YOU HAVE TO TRY AND\"
119 DATA"OPEN, WHY PICK ON THIS?#\"
120 DATA"*8OH NO!  YOU'VE OPENED THE PIRATE'S\"
121 DATA"CHEST!\"
122 DATA" \"
123 DATA"OUT OF THE CHEST JUMPS THE PIRATE, AND
124 DATA"WITH A LITTLE CHORTLE HE TAKES ALL YOUR\"
125 DATA"TREASURE AND RE-DISTRIBUTES IT ABOUT\"
126 DATA"THE CAVES.  OH WOW, REALLY HEAVY, IT
127 DATA"COULD BE ANYWHERE NOW.#\"
128 DATA"*9YOU BRAVE SOUL.  THERE'S NOTHING IN IT\"
129 DATA"NOW.\"
130 DATA" \"
131 DATA"CONSIDER YOURSELF VERY, VERY LUCKY.#^\"
139 DATA27
140 DATA"*0THERE'S NOT REALLY MUCH POINT YOU KNOW,
    THERE'S NOTHING IN IT.#\"
```


141 DATA"*1WELL, I NEVER THOUGHT IT WOULD LOOK LIKE
 142 DATA"THIS! I DON'T KNOW IF I OUGHT TO ALLOW YOU TO READ THIS.#"
 143 DATA"*2YOU KNOW, THERE ARE TIMES WHEN I THINK
 144 DATA"THAT YOU HAVEN'T THE FAINTEST IDEA WHAT YOU'RE DOING.#"
 145 DATA"*3HEAVENS, INSIDE THE BOX IS A SMALLER
 146 DATA"BOX. I WONDER WHAT HAPPENS IF YOU OPEN IT?#"
 147 DATA"*4BY 'ECK, THERE APPEARS TO BE A TREASURE
 148 DATA"IN IT. SEE, PERSEVERANCE WORKS WONDERS.#
 149 DATA"*5IT'S OPENED ONCE, AND IT ISN'T GOING TO OPEN AGAIN.#
 150 DATA"*6HMM. IT DOESN'T SEEM TO WANT TO OPEN.
 151 DATA"WHAT YOU NEED IS A ... BUT THAT WOULD BETELLING.#
 152 DATA"*7MY GOODNESS, THIS CLAM MUST REALLY BE AN
 153 DATA"OYSTER (SOMETHING'S WORKED, HARD TO BELIEVE, ISN'T IT?) B
 154 DATA" "
 155 DATA"A SMALL PEARL FALLS OUT AND ROLLS AWAY,"
 156 DATA"AND THEN THE CLAM SLAMS SHUT AGAIN.#"
 157 DATA"*8LOOK, IF THE DOOR IS ALREADY OPEN, WHY OPEN IT AGAIN?#
 158 DATA"*9YOU PUSH AND YOU SHOVE, BUT ABSOLUTELY
 159 DATA"NOTHING HAPPENS. METHINKS IT NEEDS OILING ...#^"
 169 DATA2B
 170 DATA"*0YOU MANAGE TO HEAVE THE DOOR OPEN, AND
 171 DATA"A NEW PATH OPENS OUT TO THE SOUTH.#"
 172 DATA"*1LOOK, THE DOOR'S OPEN, WHY DON'T YOU JUST WALK THROUGH IT?#
 174 DATA"*2HMM. SOMETHING SEEMS TO HAVE GONE WRONG
 175 DATA"SOMEWHERE. PERHAPS A FEW MAGIC WORDS ARE NEEDED.#"
 176 DATA"*3AS IF BY MAGIC THE DOOR CREAKS SLOWLY
 177 DATA"OPEN (O-P-E-N), AND YOU CAN NOW WALK
 178 DATA"THROUGH INTO A MAGICAL, MYSTICAL LAND.#
 179 DATA"*4THERE ARE SOME THINGS IN LIFE THAT ARE"
 180 DATA"NOT MEANT TO BE CLOSED, AND THIS IS ONE OF THEM.#
 181 DATA"*5YOU CANNOT CLOSE THE GRATE.#"
 182 DATA"*6WHAT ARE YOU ON ABOUT?#"
 183 DATA"*7YOUR BATTERIES ARE STARTING TO RUN LOW."
 184 DATA"BETTER FIND SOME NEW ONES, OR YOU'LL HAVE TO QUIT.#
 185 DATA"*8YOUR BATTERIES HAVE RUN OUT. DOESN'T IT LOOK DARK!#"

186 DATA"*9NOW, SINCE YOU DON'T APPEAR TO BE
 187 DATA"CARRYING THAT, HOW ON EARTH DO YOU
 188 DATA"EXPECT TO DROP SOMETHING YOU HAVEN'T GOT?#
 ^"
 189 DATA29
 190 DATA"*0YOU SHOULD HAVE KNOWN BETTER ... 3"
 191 DATA" "
 192 DATA"HAH. NOTHING TO WORRY ABOUT, IT COMES TO
 193 DATA"REST ON THE FLOOR.#
 194 DATA"*1EDDY LOOKS AT YOU IN HORROR, AND SAYS
 195 DATA"'WOW. DON'T GET THE HIPPY'.#
 196 DATA"*2AH. YOU DROP THE ROD BUT"
 197 DATA"IT FALLS ON THE BIRD AND CRUSHES IT!"
 198 DATA" "
 199 DATA"THE TINY LITTLE FEATHERS TURN TO DUST"
 200 DATA"BEFORE YOUR VERY EYES, AND IT FLOATS TO
 201 DATA"THE GREAT AVIARY IN THE SKY ... (AWWWW)#"
 202 DATA"*3THE TROLL IS SIGNIFICANTLY LARGER THAN"
 203 DATA"YOU ARE, SO DON'T TRY IT.#
 204 DATA"*4THE OOZY, SLIMY BODY IS IMPOSSIBLE TO
 GET HOLD OF.#
 205 DATA"*5DON'T BE RIDICULOUS. HE'S ABOUT FIFTY"
 206 DATA"TIMES YOUR SIZE.#"
 207 DATA"*6THAT IS NOT THE BEST IDEA YOU'VE EVER
 HAD IN YOUR LIFE.#"
 208 DATA"*7YOU'RE BEING SILLY, NOW AREN'T YOU?#
 209 DATA"*8NOW LOOK HERE, I CAN MANAGE TO DO MANY
 210 DATA"THINGS, BUT PERFORMING THE BLEEDIN'
 211 DATA"IMPOSSIBLE IS BEYOND EVEN ME.#"
 212 DATA"*9DWARVES HAVE EXCEPTIONALLY OILY SKINS"
 214 DATA"AND CANNOT BE PICKED UP. NOT EVEN THE F
 RIENDLY ONES.#^"
 219 DATA30
 220 DATA"*0ARE YOU BEING SERIOUS? I MEAN, ARE YOU
 221 DATA"REALLY SERIOUS? I HOPE NOT, BECAUSE I'MN
 OT GOING TO DO IT.#
 222 DATA"*1WHAT WOULD YOU DO IF YOU GOT IT? (DON'T
 ANSWER THAT).#"
 223 DATA"*2THE BIRD SEEMS TO BE FRIGHTENED OF"
 224 DATA"SOMETHING, PROBABLY YOU, BUT I'M NOT S
 URE.#
 225 DATA"*3AND WHERE IS THE CAGE TO PUT IT IN?#"
 226 DATA"*4IT APPEARS TO BE STUCK TO THE FLOOR.#"
 227 DATA"*5NO NO NO NO NO, YOU CAN'T.#
 228 DATA"*6YOU DROP THE BIRD IN FRONT OF THE SNAKE
 "
 229 DATA"AND IN AN ASTOUNDING FLURRY IT GETS E
 ATEN TO BITS ... OOOPS#"
 230 DATA"*7YOU DROP THE BIRD IN FRONT OF THE"
 231 DATA"DRAGON, BUT IT DOESN'T APPEAR TO RELISH T
 HE BATTLE ...#

232 DATA"*8AMAZING! THE BIRD MUST BE A SPIDER"
 233 DATA"EATING ONE! IT DEMOLISHES BOTH SPIDER"
 234 DATA"AND WEB.#"
 235 DATA"*9NOT IMPRESSED. IT ISN'T A TROGON YOU
 KNOW.#^"
 239 DATA31
 240 DATA"*0NOW. I DON'T KNOW IF YOU'VE NOTICED, BU
 T
 241 DATA"THIS ENORMOUS GREEN SNAKE APPEARS TO BE"
 242 DATA"IN THE WAY. IT WON'T LET YOU PAST.#"
 243 DATA"*1THERE IS AN ENORMOUS DEAD SNAKE HERE.#"
 244 DATA"*2THERE IS AN EXTREMELY LARGE CLAM SITTI
 N
 GHERE.#"
 245 DATA"*3A VERY LARGE BEDROCK BLOCK BARS YOUR
 PROGRESS.#"
 246 DATA"*4SOMETHING YOU'RE CARRYING WON'T FIT
 THROUGH THE CRACK.#"
 247 DATA"*5THERE'S A TINY LITTLE PLANT HERE,
 248 DATA"MURMURING 'WATER, WATER'.#
 249 DATA"*6THERE'S A SMALL PLANT HERE. MUTTERING
 'WATER, WATER'.#
 250 DATA"*7THERE'S A HEALTHY PLANT HERE, SHOUTING
 'WATER, WATER!'.#
 251 DATA"*8THERE'S AN ENORMOUS PLANT GROWING HERE"
 252 DATA"BELLOWING 'WATER WATER YOU B\$!%&+!'#"
 253 DATA"*9THERE IS A TINY, SHRIVELLED UP LITTLE
 254 DATA"PLANT GROWING HERE.#^
 259 DATA32
 260 DATA"*0THERE IS A GIANT VENDING MACHINE HERE,
 EVER-READY FOR BUSINESS.#
 261 DATA"*1MY GOODNESS, GOOD HEAVENS, STAND BACK I
 NAMAZEMENT!
 262 DATA" "
 263 DATA"SITTING ON A ROCK, IN A YOGA POSITION,
 264 DATA"IS A FORLORN LOOKING CHARACTER, WHO
 265 DATA"EYES YOU SUSPICIOUSLY. WEARING FLARED"
 266 DATA"JEANS, AND A BATTERED TEE-SHIRT, HE
 267 DATA"LOOKS YOU IN THE FACE. THIS IS EDDY.
 268 DATA" "
 269 DATA"EDDY IS GOING TO BE YOUR COMPANION IN"
 270 DATA"THE CAVES. WHETHER YOU LIKE IT OR NOT.
 271 DATA"'NOW MAN', HE SAYS, AS YOU PEER AT HIM,
 272 DATA"'HEAVY TRIP!'. HE STANDS UP SLOWLY AND S
 TARTS TO FOLLOW YOU.#"
 273 DATA"*2A GIANT TROLL. SKIN LIKE A RHINOCEROS,
 274 DATA"APPEARS AND BLOCKS YOUR PATH. HE IS"
 275 DATA"CARRYING A SIGN WHICH READS 'STOP! PAY T
 OLL'.#"
 276 DATA"*3A GIANT TROLL. SKIN LIKE A RHINOCEROS.
 277 DATA"APPEARS AND BLOCKS YOUR PATH. HE IS"
 278 DATA"CARRYING A SIGN WHICH READS 'STOP! PAY T

```

OLL'.#"
279 DATA"*4YOU CAN HEAR SOME FAINT RUSTLING NOISES
    IN THE DISTANCE.#"
280 DATA"*5A DWARF SUDDENLY SCUTTLES ROUND A"
281 DATA"CORNER, AND VICIOUSLY THROWS AN ANCIENT
282 DATA"AXE AT YOU.  HE THEN SCAMPERS OFF INTO T
    HE DISTANCE.#"
283 DATA"*6A DWARF APPEARS, AND A KNIFE HURTLES
    TOWARDS YOU.#
284 DATA"*7OUT FROM THE SHADOWS BEHIND YOU POUNCES
    "
285 DATA"A BEARDED PIRATE.  'HAR HAR HAR', HE
286 DATA"CHORTLES.  'I'LL JUST TAKE ALL THIS"
287 DATA"BOOTY AND HIDE IT WITH ME CHEST"
288 DATA"IN THE MAZE!'.  HE SNATCHES YOUR"
289 DATA"TREASURE AND VANISHES OFF INTO THE      G
    LOOM.#"
290 DATA"*8THERE IS A RUSTY DOOR IN YOUR WAY,
    BLOCKING FURTHER PROGRESS.#
291 DATA"*9GOOD HEAVENS!  THERE'S A GOLD COIN HERE
    '
292 DATA"EMBEDDED INTO THE FLOOR.  I WONDER HOW"
293 DATA"YOU'RE GOING TO GET IT OUT OF THE      G
    ROUND.#^
300 DATA-1

```

READY.

You'll see that, as before, everything starts off with a numerical data statement, followed by a whole host of string data. When you've finished typing it all in, run the program, and then we can start the job of checking it all again.

Enter GOTO25 as before, and when the prompt for the number appears type in 260, the first of the message numbers. This should read:

Can't you type something in right for a change?

with the prompt for another number sitting below it. Go through all the messages to make sure they're correct, and re-run the program if you find any mistakes. Don't forget about inserting the occasional cursor-up when the 64 decides to skip a line.

Having checked the first program, you can now type in the rest of the messages.

99 DATA33
 100 DATA"*ONOW. YOU PROBABLY HAVEN'T NOTICED IT,
 101 DATA"BUT RIGHT IN FRONT OF YOU, AND TWO FEET
 102 DATA"AWAY. IS A LARGE AND EXTREMELY CROSS
 103 DATA"DRAGON. WITH SMOKE POURING OUT OF HIS"
 104 DATA"NOSTRILS. AND HATRED POURING OUT OF HIS
 105 DATA"EYES. HE LOOKS A FEARSOME SIGHT.#
 106 DATA"*1ACROSS THE PATH IN FRONT OF YOU IS"
 107 DATA"STREWN A HUGE SPIDER'S WEB. AND RIGHT
 108 DATA"IN THE MIDDLE OF IT IS A HUGE BULBOUS"
 109 DATA"SPIDER. 'ECK!#" "
 110 DATA"*2THERE IS A LARGE DOOR IN FRONT OF YOU." "
 111 DATA"AND ON THE DOOR IS WRITTEN SOME ANCIENT"
 112 DATA"RUNIC MESSAGE. I WOULDN'T BOTHER"
 113 DATA"READING IT. IF I WERE YOU.#
 114 DATA"*3HMM. YOU APPEAR TO BE IN THE MIDDLE OF
 115 DATA"A LARGE UNDERGROUND BAR. THERE'S A HUGE
 116 DATA"BARMAN HERE, AND EMPTY POTS AND DUSTY C
 HAIRS LIE EVERYWHERE.#
 117 DATA"*4BY 'ECK. THERE'S A BLOODY GREAT TROLL
 118 DATA"HERE. HE COULD BE THE BROTHER OF THE"
 119 DATA"ONE YOU MET EARLIER. ALTHOUGH IT'S HARD
 120 DATA"TO TELL REALLY. ANYWAY, HE WANTS SOME
 121 DATA"TREASURE BEFORE HE'LL LET YOU PAST.#"
 122 DATA"*5A GIANT TROLL APPEARS AND BLOCKS YOUR
 PATH.#"
 123 DATA"*6YOUR TORCH BOUNCES ALONG IN THE TUNNEL,
 AND IS LOST TO SIGHT.#
 124 DATA"*7THE BRACKEN BLOCKS YOUR ACCESS TO THE
 DOOR.#"
 125 DATA"*8YOU HAVE SOME WEIRD IDEAS, AND THIS IS
 DEFINITELY ONE OF THEM.#
 126 DATA"*9HMMM, NOT BAD, BUT NOT A VINTAGE YEAR.#
 ^
 127 DATA-1

READY.

100 DATA34
 102 DATA"*0OH, LIKE, WOW MAN, THIS STUFF IS REALLY
 104 DATA"SOMETHING ELSE. YOU SEEM TO BE FLOATING
 106 DATA"ON AIR ...]":DATA" "
 107 DATA"IN FACT, YOU ARE FLOATING ...]#
 110 DATA"*1I REALLY WOULD HAVE THOUGHT THAT YOU'D
 111 DATA"HAVE KNOWN BETTER THAN TO ASK FOR ANY
 112 DATA"HELP. YOU'RE NOT GETTING ANY, NOT NOW,
 113 DATA"NOT LATER, NOT EVER!!!#
 120 DATA"*2CHOMP ... CHOMP ... CHOMP ...":DATA" "
 121 DATA"THAT WAS REALLY NICE. PITY YOU CAN'T S
 OLVE THE GAME NOW ...#
 130 DATA"*3WELL, I DON'T THINK THAT'S A VERY NICE

131 DATA"IDEA, SO I WON'T LET YOU DO IT. JUST
 132 DATA"LEAVE THE POOR BIRD ALONE, IT'S GOT
 133 DATA"ENOUGH TO DO IN THIS GAME AS IT IS.#
 140 DATA"*4FANCY A DRAGON BURGER DO WE? NO, YOU'LL
 141 DATA"HAVE TO THINK OF ANOTHER WAY TO GET PASTT
 HE DRAGON.#
 150 DATA"*5OH WOW, REALLY HEAVY TRIP. EDDY STARTS
 151 DATA"WAVING HIS ARMS ABOUT AND RUNNING AROUND
 152 DATA"ALL OVER THE PLACE, BUT SOON CALMS DOWN
 153 DATA"AND COMES BACK TO YOU.#
 160 DATA"*6I DON'T THINK THAT WOULD BE A VERY GOOD
 IDEA.#
 170 DATA"*7DO YOU REALLY WANT TO SOLVE THIS GAME?
 171 DATA" ":DATA"COME ON NOW, BE SENSIBLE.#
 172 DATA"*8GULP ... GULP ... GULP ... HIC!#
 180 DATA"*9WELL, LIKE, HEAVY, BUT YOU HAVEN'T GOT
 ANY LENTILS, MAN.#^
 185 DATA35
 190 DATA"*0THE BIRD ISN'T HUNGRY, IT'S MERELY
 191 DATA"PININ' FOR THE FJORDS. ANYWAY, IT LOOKS
 192 DATA"NOTHING LIKE A NORWEGIAN BLUE TO ME.#
 210 DATA"*1I DON'T THINK SO SOMEHOW.#
 212 DATA"*2YOU SEE NOTHING OF INTEREST.#
 214 DATA"*3I'M NOT GOING TO TELL YOU ANYTHING THAT
 215 DATA"MIGHT POSSIBLY HELP YOU SOLVE THIS GAME.#
 216 DATA"*4EEEH, YOU CAN'T GET A DECENT PINT FOR
 217 DATA"LOVE NOR MONEY THESE DAYS CAN YOU?
 218 DATA"STILL, IT SEEMS TO TASTE ALL RIGHT (
 HIC, WOBBLE, BELCH ...)#
 250 DATA"*5A NEW SET OF BATTERIES HAS APPEARED ON
 251 DATA"THE GROUND IN FRONT OF YOU. THAT WAS A
 252 DATA"TURN UP FOR THE BOOKS WASN'T IT?#
 260 DATA"*6WELL WOULD YOU ADAM 'N' EVE IT. THE
 261 DATA"COIN HAS RE-APPEARED AGAIN.#
 270 DATA"*7I'M TAKING THE LIBERTY OF REPLACING YOU
 RBATTERIES.#
 280 DATA"*8HMM, YOU KNOW IN THE ORIGINAL GAME WHEN
 281 DATA"YOU DID THIS IN THE RIGHT LOCATION, AND
 282 DATA"THE GOLDEN EGGS RE-APPEARED?":DATA"]"
 283 DATA"WELL, IT DOESN'T WORK IN THIS ONE. YOU
 284 DATA"HEAR FOOTSTEPS IN THE DISTANCE,
 285 DATA"THUNDEROUS FOOTSTEPS, AND ROUND A CORNER
 286 DATA"APPEARS THE BIGGEST GIANT YOU'VE EVER
 287 DATA"SEEN! HE'S BIGGER THAN YOU, ANYWAY.":DAT
 A" "
 288 DATA"WITH ONE MIGHTY SWIPE OF HIS ARM, HE
 289 DATA"KNOCKS YOU TO THE FLOOR AND KILLS YOU!":D
 ATA" "
 290 DATA"AND SINCE I HAVEN'T GOT ANY ORANGE SMOKEO
 N ME, THAT'S YER LOT!#

291 DATA"*9YOU APPEAR TO HAVE FOUND THE MAGIC WORD
 .":DATA" "
 292 DATA"THE GROUND SHIMMIES AND SHAKES,THE
 293 DATA"SPECTRE SEEMS TO DISSOLVE INTO NOTHING,
 294 DATA"AND AS IF BY MAGIC YOU ARE TRANSPORTED
 295 DATA"BEYOND IT. DON'T SUPPOSE IT'LL WORK A
 GAIN THOUGH.#^
 300 DATA36
 302 DATA"*0THAT IS AN OLD FASHIONED MAGIC WORD, AN
 DHAS NO EFFECT IN THE CAVES.#
 310 DATA"*1SOMETHING YOU'RE CARRYING WON'T FIT
 THROUGH THE CRACK.#
 320 DATA"*2YOU HAVE SPENT TOO LONG IN THE CAVES ..
 ."
 322 DATA"A HIDEOUS CREATURE. A CREATION OF THE
 324 DATA"ROCKS, STRETCHES OUT TOWARDS YOU, AND
 326 DATA"WITH A SINGLE MOVEMENT YOU ARE SWEEPED TO T
 HE FLOOR. OH DEAR ...#
 330 DATA"*3LOOK. THIS IS THE DARK ROOM. ISN'T IT?"
 :DATA" "
 332 DATA"IF I'VE TOLD YOU ONCE I'VE TOLD YOU A
 333 DATA"THOUSAND TIMES, YOU CAN'T DO ANYTHING IN
 334 DATA"THE DARK OR YOU MIGHT FALL INTO A PIT.#
 340 DATA"*4THIS IS NOT THAT KIND OF GAME, SO JUST
 GET ON WITH IT.#
 350 DATA"*5THE GROUND QUIVERS BENEATH YOUR FEET,
 352 DATA"MIGHTY RUMBLINGS FILL THE AIR, A CRYSTAL
 353 DATA"BRIDGE APPEARS AND ...]
 354 DATA" ":DATA"OH NO, IT FALLS TO THE BOTTOM OF
 THE
 356 DATA"CHASM, WHICH IS NOW MUCH WIDER, AND YOU
 357 DATA"CAN'T POSSIBLY JUMP ACROSS IT.#
 360 DATA"*6DON'T BE SILLY. NOTHING HAPPENS.#
 370 DATA"*7YOU PLUMMET INTO THE ABYSS BY YOUR SIDE
 AND DIE A HORRIBLE DEATH ...J#
 380 DATA"*8WITH A MIGHTY BOUND AND A JACK-KNIFE IN
 382 DATA"MID-AIR, YOU MAKE IT TO THE OTHER SIDE.#
 390 DATA"*9BOING ... BOING ... BOING. LOTS OF FUN
 ,
 391 DATA"BUT NOTHING MUCH HAPPENS.#^
 400 DATA-1

READY.

100 DATA37
 102 DATA"*0OH NO! THE CRACK IS MUCH TOO WIDE, YOU
 104 DATA"JUST FAIL TO REACH THE OTHER SIDE, AND
 106 DATA"WITH A DESPERATE SCRABBLING OF HANDS,
 108 DATA"FALL DOWN INTO THE BOTTOMLESS PIT FAR B
 ELLOW ...#
 110 DATA"*1PHEW, THAT WAS CLOSE, BUT YOUR HANDS

112 DATA"COULDN'T QUITE MANAGE TO HANG ON. #
 113 DATA"*2HA HA! MADE IT, AND YOU CAN NOW CARRY
 ON WITH THE GAME. #
 120 DATA"*3ALL THE OIL FALLS AT YOUR FEET AND SOAK
 S
 121 DATA"AWAY INTO THE FLOOR. #
 130 DATA"*4THE OIL FREES THE DOOR AND YOU CAN NOW
 132 DATA"MOVE IT, ALTHOUGH IT REQUIRES SOME E
 FFORT. #
 150 DATA"*5YOU'VE NOTHING TO PUT IT IN. #
 160 DATA"*6DRIP ... DRIP ... DRIP ...":DATA" ":DAT
 A"ALL OVER YOUR FEET. #
 170 DATA"*7THE TROLL CATCHES THE PYRAMID, AND WITH
 171 DATA"A CACKLE SCUTTLES OFF INTO THE DARKNESS. #
 180 DATA"*8THE TROLL CATCHES THE AXE, LOOKS AT IT,
 181 DATA"SAYS 'NICE WORKMANSHIP, BUT NOT VERY
 182 DATA"VALUABLE', AND THROWS IT BACK AT YOUR F
 EET. #
 190 DATA"*9THE TROLL CATCHES THE AXE, LOOKS AT IT,
 192 DATA"SAYS 'NICE WORKMANSHIP, AND PERHAPS IT
 194 DATA"IS WORTH SOMETHING', AND THEN LUMBERS
 195 DATA"OFF INTO THE DARKNESS. #^
 200 DATA38
 202 DATA"*0THE TROLL CATCHES YOUR TREASURE, AND
 204 DATA"WITH MUCH RUBBING OF HANDS, STROLLS OFF
 206 DATA"INTO THE GLOOM AND DARKNESS OF THE C
 AVES. #
 210 DATA"*1THE TROLL CATCHES IT, LOOKS AT IT, AND
 212 DATA"SAYS 'WORTHLESS, ABSOLUTELY WORTHLESS',
 214 DATA"BEFORE THROWING IT BACK AT YOUR FEET. #
 220 DATA"*2THE DWARF DARTS TO ONE SIDE AND DODGES
 YOUR THROW. #
 230 DATA"*3A DWARF DARTS OUT FROM BEHIND A ROCK AN
 D
 231 DATA"THROWS A KNIFE AT YOU! #
 240 DATA"*4YOU'VE KILLED A LITTLE DWARF.":DATA" J"
 241 DATA"LIKE ALL DWARVES, HIS BODY VANISHES IN A
 GREASY CLOUD OF BLACK SMOKE. #
 250 DATA"*5DWARVES' KNIVES ARE MAGIC, AND VANISH A
 SSOON AS THEY'RE THROWN. #
 255 DATA"*6THE KNIFE SINKS INTO YOUR SIDE, AND WIT
 H
 256 DATA"BLOOD DRIPPING FROM THE OPEN WOUND, YOU
 257 DATA"MANAGE TO UTTER THE WORD 'RATS' BEFORE
 258 DATA"COLLAPSING IN A DYING HEAP. #
 260 DATA"*7THERE IS A LITTLE GRINNING DWARF HERE. #
 270 DATA"*8THE EVER-PEACEFUL EDDY IS BUMBLING ALON
 GBEHIND YOU. #
 280 DATA"*9YOU'RE NOT STRONG ENOUGH TO DO THAT. #^
 300 DATA39
 301 DATA"*0I HAVEN'T THE FAINTEST IDEA WHAT YOU'RE

THINKING OF SOMETIMES. #
 302 DATA"*1IF YOU HAVEN'T GOT ANY WATER, HOW DO YO
 UEXPECT TO WATER SOMETHING?#
 320 DATA"*2YOU WATER THE LITTLE PLANT AND IT GROWS
 A TINY AMOUNT. #
 330 DATA"*3YOU WATER THE PLANT AND IT GROWS A
 LITTLE. #
 340 DATA"*4YOU WATER THE LARGE PLANT AND IT GROWS
 341 DATA"ENORMOUSLY, ALMOST FILLING THE PIT. #
 350 DATA"*5OH NO! YOU'VE OVERWATERED IT, AND THE
 351 DATA"POOR LITTLE THING SHRIVELS AND DIES. #
 360 DATA"*6OH, JUST LEAVE THE POOR LITTLE PLANT
 361 DATA"ALONE. IT'S NOT HAPPY WITH YOU, YOU K
 NOW. #
 370 DATA"*7YOU'RE NOT STRONG ENOUGH TO BREAK YOUR
 371 DATA"WAY OUT OF A PAPER BAG, SO I WOULDN'T B
 OTHER TRYING. #
 380 DATA"*8HMM, INTERESTING REQUEST, BUT I DON'T
 381 DATA"THINK I'LL CARRY IT OUT. #
 390 DATA"*9AND JUST WHAT DO YOU EXPECT TO MOVE IT
 WITH?#^
 400 DATA40
 401 DATA"*0YOU'VE ALREADY DONE THAT, SO DON'T TRY
 MY PATIENCE. #
 402 DATA"*1HURRAY, THE COIN CREAKS ITS WAY OUT OF
 403 DATA"THE GROUND, AND WITH A FINAL LITTLE
 404 DATA"'PLOP' FALLS AT YOUR FEET. #
 420 DATA"*2HOW DO YOU EXPECT TO DO THAT WITHOUT TH
 EAXE?#
 430 DATA"*3THE PLANT WON'T LET YOU. IT DODGES OUT
 431 DATA"OF THE WAY OF YOUR CLUMSY BLOW, SHAKES
 432 DATA"ITS ROOTS, AND SETTLES DOWN AGAIN. #
 440 DATA"*4I WORRY ABOUT YOU SOMETIMES. YOU'RE
 441 DATA"THE ONLY VAGUELY SENSIBLE THING AROUND,
 442 DATA"AND YET HERE YOU ARE TALKING TO NOTHING.0
 DD, DASHED ODD. #
 450 DATA"*5ARE YOU SCHIZOPHRENIC OR SOMETHING?
 451 DATA"EDDY'S HERE, TALK TO HIM. #
 460 DATA"*6WHAT DO YOU WANT POOR OLD EDDY TO DO?#
 470 DATA"*7THE INCREDIBLE EDDY TURNS GREEN, MUSCLE
 S
 471 DATA"RIPPLE, HIS SHIRT GETS TORN TO RIBBONS.
 472 DATA"AND WITH A MIGHTY ROAR HE TEARS THE
 473 DATA"DRAGON LIMB FROM LIMB.":DATA"]"
 474 DATA"COR, THAT SUPRISED YOU DIDN'T IT?]
 475 DATA" ":DATA"EDDY SHAKES HIS HEAD IN BEWILDERM
 ENT,
 476 DATA"LOOKS AT HIS TORN SHIRT, AND MUTTERS
 477 DATA"'NOW, HEAVY HEAVY HEAVY', BEFORE S
 ETTLING DOWN AGAIN. #
 480 DATA"*8'LIKE, I'M SUPPOSED TO BE A REALLY

482 DATA"PEACEFUL HIPPY, RIGHT, SO DON'T TRY AND
 484 DATA"GET ME TO DO THINGS THAT ARE HEAVY AND U
 N-COOL, MAN'.#
 490 DATA"*9EDDY LOOKS AT THE ROCK, SAYS 'COME ON
 491 DATA"LITTLE ROCK, OUT OF THE WAY', AND BEFORE
 492 DATA"YOUR VERY EYES IT MOVES TO ONE SIDE.
 494 DATA" "
 495 DATA"EDDY TELLS YOU THAT HE, LIKE, REALLY U
 NDERSTANDS ROCKS, RIGHT?#^
 500 DATA41
 502 DATA"*0'WELL, I REALLY FANCIED A SWIM', AND IN
 504 DATA"HE DIVES. THE WATERS THRASH INTO A
 506 DATA"FRENZY, AND HE DISAPPEARS, TAKING A
 508 DATA"FLURRY OF CREATURES WITH HIM. THE POOL
 509 DATA"NOW LOOKS CLEAR AND SAFE TO SWIM ACROSS.#
 510 DATA"*1'7THAT'S NICE, ISN'T IT, I HELP YOU OUT,
 512 DATA"I'M LIKE A REALLY GOOD COMPANION, RIGHT,
 514 DATA"AND NOW YOU WANT ME TO GET LOST. HUH!','
 515 DATA"AND WITH A SHRUG OF HIS SHOULDERS HE S
 HUFFLES OFF.#
 520 DATA"*2'HA, I KNOW THIS ONE', AND HE SITS DOWN
 522 DATA"AND STARTS SINGING ABOUT GOLD.#
 524 DATA"*3A FIGURE APPEARS OUT OF THE GLOOM, AND
 525 DATA"SAYS 'HELLO', BEFORE, LIKE, SQUATTING D
 OWN BESIDE YOU.#
 540 DATA"*4YOU FALL OVER IN AN ENTHUSIASTIC KICK,
 542 DATA"AND GET UP AGAIN, FEELING EMBARRASSED.#
 550 DATA"*5ERM, WELL, THIS IS A LOT LARGER THAN
 551 DATA"YOU ARE, AND IT'S ALL BULBOUS AND
 552 DATA"HORRIBLE, SO FOR YOUR OWN SAKE I WON'T L
 ET YOU DO IT.#
 560 DATA"*6TROLLS ARE BROTHERS OF THE ROCK AND HAV
 E
 561 DATA"SKINS LIKE THAT OF A RHINOCEROS. SO
 562 DATA"YOUR BLOWS HAVE NO EFFECT.#
 570 DATA"*7BEING ABOUT FIFTY MILLION TIMES YOUR
 571 DATA"SIZE, YOUR BLOWS HAVE NO EFFECT AND THE
 572 DATA"DRAGON JUST LIES THERE, IGNORING YOU.#
 580 DATA"*8OH, REALLY UN-COOL AND HEAVY. EDDY
 581 DATA"DODGES OUT OF THE WAY AND GIVES YOU A B
 ALEFUL GLARE.#
 590 DATA"*9COME ON, WITH WHAT, YOUR BARE HANDS?#^
 591 DATA-1

READY.

100 DATA42
 102 DATA"*0THIS IS NOT ONE OF YOUR BETTER IDEAS.#
 110 DATA"*1YOU'VE ALREADY ATTACKED AND KILLED THE
 111 DATA"SNAKE, IT IS DEAD, IT IS AN EX-SNAKE, W
 HAT MORE DO YOU WANT?#

120 DATA"*20H NO, OH DEAR OH DEAR OH DEAR!J":DATA"
 122 DATA"THE SNAKE FLIES AT YOU, AND WITH MUCH
 124 DATA"HISSING AND DARTING OF ITS EVIL LITTLE
 125 DATA"TONGUE IT WRAPS ITS COILS AROUND YOU ANDS
 QUEEZES ...J":DATA" "
 126 DATA"UNTIL YOU'RE DEAD!#
 130 DATA"*3AMAZING, ASTOUNDING, SUCH STRENGTH AND
 131 DATA"FEROCITY!":DATA" "
 132 DATA"YOU FLY AT THE SNAKE AND, IN AN
 133 DATA"INCREDIBLE DISPLAY OF STRENGTH, YOU KILL
 134 DATA"IT. HARD TO BELIEVE, ISN'T IT?#
 140 DATA"*4I'M AFRAID YOU HAVE NO DYNAMITE.#
 145 DATA"*5THROUGHOUT THE COURSE OF THIS GAME
 146 DATA"YOU'VE HAD SOME PRETTY STRANGE IDEAS, B
 UT THIS BEATS THEM ALL.#
 150 DATA"*6THE WALLS ARE CRUSHED BY THE FORCE OF
 151 DATA"THE EXPLOSION, A GAPING HOLE APPEARS IN
 152 DATA"THE WEST WALL, AND A VAST TIDE OF
 153 DATA"STEAMING LAVA FLOWS IN AND ENGULFES YOU.#
 160 DATA"*7THE SOUND OF THE EXPLOSION WAKES UP THE
 161 DATA"SLEEPING DWARVES. SOME OF THEM ARE
 162 DATA"CARRYING KNIVES, MANY OF THE DWARVES
 163 DATA"THROW THEM AT YOU, ALL OF THEM GET YOU!#
 180 DATA"*8THE WALLS ARE CRUSHED BY THE FORCE OF
 181 DATA"THE EXPLOSION, A GAPING HOLE APPEARS IN
 182 DATA"THE EAST WALL, AND A VAST TIDE OF
 183 DATA"STEAMING LAVA FLOWS IN AND ENGULFS THE
 184 DATA"DWARVES.":DATA" "
 185 DATA"7THROUGH ANOTHER HOLE IN THE WEST WALL
 186 DATA"COMES A HORDE OF CHEERING ELVES, WHO
 187 DATA"PICK YOU UP AND CARRY THE CONQUERING
 188 DATA"ADVENTURER OFF INTO THE DISTANCE.":DATA"
 J"
 189 DATA"AFTER DAYS OF FEASTING, YOU RETURN TO
 190 DATA"YOUR TREASURES IN THE BUILDING, AND
 191 DATA"FIND, TO YOUR AMAZEMENT, AN EXTREMELY
 192 DATA"DAMP EDDY SITTING ON TOP OF THE GOODIES,
 193 DATA"A WET FISH HANGING BEHIND HIS EAR.":DATA"
 "
 194 DATA"HE LOOKS AT YOU, AND SAYS 'PHEW, THAT
 195 DATA"WAS CLOSE'.":DATA" "
 196 DATA"HELL, WELL, WELL. IT LOOKS LIKE YOU'VE
 197 DATA"SOLVED THIS ADVENTURE AFTER ALL.#
 199 DATA"*9YOU ARE A LEADER, NOT A FOLLOWER, SO
 LEAD, DON'T FOLLOW.#^
 200 DATA43
 201 DATA"*0IF YOU WANT TO FIND SOMETHING, LOOK FOR
 IT YOURSELF.#
 210 DATA"*1YOU CUT AND YOU SWIPE, AND THE FORCE OF
 212 DATA"YOUR EFFORTS TEARS THE BRACKEN INTO T

INY PIECES.#
 220 DATA"*2YOU CAN'T. THE BRACKEN OBSCURES JUST
 ABOUT EVERYTHING.#
 270 DATA"*7#MM, IT WOULD APPEAR THAT YOU'VE
 FORGOTTEN THE KEY.#
 280 DATA"*8LO AND BEHOLD, IT OPENS, AND YOU CAN NO
 W
 281 DATA"GAIN ACCESS TO THE FINAL LOCATION ...#
 290 DATA"*9THEY ARE SMALL AND BLUE, AND DART OUT O
 F
 291 DATA"REACH AS SOON AS YOU TRY AND GET HOLD OF T
 HEM.#^
 300 DATA44
 301 DATA"*0LOOK, JUST GET ON WITH IT, WILL YOU?
 STOP WASTING TIME.#
 310 DATA"*1YOU START TO SWIM ACROSS THE LAKE, BUT
 311 DATA"REALISE AFTER A WHILE THAT YOU CAN'T
 312 DATA"SWIM, AND WITH A GULP AND A GASP YOU V
 ANISH BENEATH THE WATERS ...#
 320 DATA"*2AS SOON AS YOU SET FOOT IN THE WATERS
 321 DATA"THEY ARE CHURNED INTO A FRENZY, AND A
 322 DATA"MYRIAD OF RAZOR-SHARP TEETH DRAG YOU
 323 DATA"DOWN BENEATH THE WATERS.#
 325 DATA"*3JUST WHEN YOU THOUGHT IT WAS SAFE TO
 326 DATA"GO BACK INTO THE WATERS, ALONG COMES A
 327 DATA"DIRTY GREAT MONSTER AND SWALLOWS YOU UP I
 N ONE GO.#
 330 DATA"*4LOOK, NO SWIMMING ALLOWED HERE, ALL
 RIGHT?#
 340 DATA"*5NOW THAT EDDY HAS CLEARED THE WAY, YOU
 341 DATA"SWIM THROUGH UN-IMPEDED AND ARRIVE
 342 DATA"SAFELY ON THE OTHER SIDE.#
 360 DAATA"*6SNIFF SNIFF SNIFF SMELL SMELL SMELL.":
 DATA" "
 361 DATA"THE USUAL GHASTLY ODOURS ASSAULT YOUR N
 OSE.#
 370 DATA"*7SO WHERE'S THE LAKE, DUMBO?#
 380 DATA"*8YOU HAVE NO OARS, SO YOU CAN'T GET
 ANYWHERE.#
 390 DATA"*9YOU ROW ON AND ON ...#^
 400 DATA-1

READY.

100 DATA45
 102 DATA"*0THERE DOESN'T SEEM TO BE A BOAT ON THE
 104 DATA"GROUND BESIDE YOU TO ROW IN.#
 110 DATA"*1YOU CAN'T PAY BY CHEQUE OR VISA CARD IN
 112 DATA"THE CAVES, YOU KNOW, AND SINCE YOU
 114 DATA"HAVEN'T GOT ANY MONEY, THE BARTENDER T
 URFS YOU OUT.#

120 DATA"*2THAT WON'T WORK, YOU HAVEN'T GOT ANY
 MONEY.#
 122 DATA"*3YOU GET YOURSELF A DRINK, BUT JUST THEN
 124 DATA"EDDY LOOMS UP AND SAYS 'NOW, A DRINK,
 125 DATA"I'LL HAVE ONE OF THOSE', AND HELPS
 126 DATA"HIMSELF TO A DRINK. ":DATA" "
 127 DATA"UNFORTUNATELY, YOU HAVEN'T GOT ANY
 128 DATA"MONEY TO PAY FOR IT, SO YOU BOTH GET E
 JECTED EMPTYHANDED.#
 130 DATA"*4'T'ANK EE' KINDLY SIR', SAYS THE
 132 DATA"BARTENDER, TAKING YOUR MONEY.":DATA" "
 134 DATA"'BUT I DON'T ACCEPT BRIBES, YOU SEE', A
 ND HE THROWS YOU OUT.#
 150 DATA"*5YOU MANAGE TO BUY JUST ONE DRINK FOR
 152 DATA"YOURSELF, AND THE BARTENDER BIDS YOU ON Y
 OUR WAY.#
 160 DATA"*6YOU GIVE THE CREATURE YOUR DRINK, AND
 161 DATA"WITH A HIC AND A BELCH IT SCUTTLES OFF.#
 170 DATA"*7IT'S WRITTEN IN DWARVISH, AND NOT
 171 DATA"POSSESSING A DEGREE IN LANGUAGE
 172 DATA"TRANSLATION, YOU CAN'T READ IT. #IND
 173 DATA"YOU, PAGE THREE LOOKS NICE.#"
 180 DATA"*8IT'S ALL TORN AND BATTERED, AND YOU
 181 DATA"CAN'T MAKE ANYTHING OUT.#
 190 DATA"*9I'VE ALREADY TOLD YOU EVERYTHING THAT'S
 WRITTEN ON IT.#^
 200 DATA46
 202 DATA"*0I THINK IT PROBABLY MEANS THAT 'PLUGH'
 IS A MAGIC WORD.#"
 210 DATA"*1WHAT'S THE MATTER, CAN'T YOU READ
 YOURSELF?##"
 220 DATA"*2A VOICE BOOMS OUT AND SAYS 'ONE OF THE
 221 DATA"WORDS I USED TO KNOW NOW HAS A VERY
 222 DATA"DIFFERENT MEANING'.#
 230 DATA"*3MMM, THIS COULD HAVE A MAGIC EFFECT, IF
 USED WISELY.#
 240 DATA"*4AYE, WHY ME INDEED?#
 250 DATA"*5THERE DOESN'T APPEAR TO BE MUCH WRITTEN
 ON THIS.#
 260 DATA"*6YOU CAN'T CLIMB ANYWHERE AROUND HERE
 WITHOUT A ROPE.#
 270 DATA"*7THE PLANT ISN'T BIG ENOUGH YET.#
 280 DATA"*8THE PLANT IS LARGE AND BUSHY, AND YOU
 CAN CLIMB IT WITH EASE.#
 290 DATA"*9THERE DOESN'T SEEM TO BE MUCH TO CLIMB
 AROUND HERE.#^
 300 DATA47
 302 DATA"*0WITH AN ALMIGHTY EFFORT YOU SHIMMY YOUR
 303 DATA"WAY ALONG THE ROPE AND ARRIVE ON TERRA
 304 DATA"FIRMA ONCE MORE. BEING A WISE AND
 305 DATA"INTELLIGENT PERSON, YOU UNTIE THE ROPE

```

306 DATA"AND LEAVE IT ON THE GROUND NEXT TO YOU.#
308 DATA"*10OUT FROM BEHIND A ROCK JUMPS A BEARDED
310 DATA"PIRATE. 'HAR HAR HAR', HE CHUCKLES,
312 DATA"'I'LL JUST TAKE THIS BOOTY AND HIDE IT
314 DATA"WITH ME CHEST, DEEP IN THE MAZE'." :DATA"
"

```

```

316 DATA"BEFORE YOU CAN BLINK, HE STEALS YOUR
318 DATA"TREASURE AND SCURRIES OFF INTO THE
LOOM.#^
400 DATA-1

```

G

READY.

Phew! But we're nearly there now, with just verbs, nouns, objects and the map to go.

Verbs, nouns and objects

'Enormous Cave' has quite a wide vocabulary, and can understand some 84 verbs and 107 nouns. Not as much as 'Zork', I must admit, but a reasonably healthy number nevertheless.

```

1 DIMVB$(84).NO$(107).OB$(107).OB%(107,1):Z$=CHR$(
13)
2 FORI=0TO5:READD$(I):NEXT
3 FORI=1TO84:READVB$(I):NEXT
4 FORI=1TO107:READNO$(I):NEXT
5 FORI=1TO107:READOB$(I):NEXT
6 FORI=1TO107:READOB%(I,0):NEXT
7 GOSUB209
10 OPEN2,8,2,"O:STUFF,S,W"
20 FORI=0TO5:PRINT#2,D$(I):Z$::NEXT
30 FORI=1TO84:PRINT#2,VB$(I):Z$::NEXT
40 FORI=1TO107:PRINT#2,NO$(I):Z$::NEXT
50 FORI=1TO107:PRINT#2,OB$(I):Z$::NEXT
60 FORI=1TO107:PRINT#2,OB%(I,0):Z$::NEXT
70 FORI=1TO107:PRINT#2,OB%(I,1):Z$::NEXT
80 CLOSE2:END
99 REM DD$ STUFF
100 DATA"NORTH","SOUTH","EAST","WEST","UP","DOWN"
109 REM VB$ STUFF
110 DATAGO,N,S,E,W,U,D,NORT,SOUT,EAST,WEST,UP,DOWN
,LOAD,SAVE,HELP,SCOR,LOOK,REDE
112 DATAINVE,EXAM,****,****,****,****,****,EAT,DRIN,FEE
D,THRO,TALK,ASK,FEE .YME,YZZY
114 DATAABRA,OPEN,USE,LIGH,LAMP,GET,PUT,TAKE,DROP,
ATTA,HIT,KILL,KICK,DEST,JUMP
116 DATAWAVE,PUSH,MOVE.SHIF,ENTE,WATE,CLOS,OIL,BRE

```

A,LEVE,REMO,CUT,CHOP,CLIMB
 118 DATABUY,PAY,BRIB,ROW,GIVE,OFFE,SMOK,SWIM,SMAS,
 BLAS,FOLL,FIND,CLIM,READ,SNIF
 120 DATASMEL,CATC,TURN,QUIT,SHOR,LONG
 129 REM NO\$ STUFF
 130 DATATRID,COIN,RUG,CHES,JEW,PEAR,SILV,NEST,EGG
 ,VASE,ORB,SPIC,RING,SWOR
 132 DATANUGG,N,S,E,W,U,D,NORT,SOUT,EAST,WEST,UP,DO
 WN,PROG,EDDY,HELP,GRAT,KEY
 134 DATAFOOD,BOTT,BOTT,WATE,LAMP,LAMP,BATT,ON,OFF,
 BIRD,CAGE,ROD,SNAK,HAND,CLAM
 136 DATACHAS,FURT,ROCK,CRAC,PLAN,MACH,TROL,AXE,DOO
 R,LEDG,HIGH,STIL,GIAN,DWAR
 138 DATACROW,DRAG,BOX,SPID,WEB,SESA,ROPE,DRIN,BART
 ,BOAT,OARS,CREA,ASTR,PYRA,POOL
 140 DATAMONS,SWIM,BRAC,DYNA,BUIL,STRE,VALL,TRAP,TR
 EE,CHAM,NOTE,AIR,SHEL,TAB,WALL
 142 DATAWRIT,BRID,SPEC,MUD,CHAI,MUG,SHOR,STRU,TONY
 ,MAGA,MOSS,INSE,AUST,PIRA
 144 DATAOIL,OIL
 149 REM OB\$ STUFF
 150 DATA"A JEWELLED TRIDENT!","A GOLD COIN!","AN O
 RIENTAL RUG!"
 152 DATA"THE PIRATE'S CHEST!","A SHINING JEWEL!","
 A PEARL!","BARS OF SILVER!"
 154 DATA"A NEST OF GOLDEN EGGS!","A PLOVER'S EGG!"
 ,"A MING VASE!","A GOLDEN ORB!"
 156 DATA"RARE SPICES!","A PLATINUM RING!","A JEWEL
 LED SWORD!","A GOLD NUGGET!"
 157 DATA+,+,+,+,+,+,+,+,+,+,+,+,+,COMRADE EDDY,+,A
 STRONG GRATE,A KEY,SOME FOOD
 158 DATAAN EMPTY BOTTLE,A BOTTLE OF WATER,SOME WAT
 ER,AN OLD LAMP,A BLAZING LAMP
 160 DATASOME BATTERIES,+,+,A SINGING BIRD,A GILDED
 CAGE,A BLACK ROD
 162 DATAA HISSING SNAKE,+,A CLAM,+,+,+,+,A PLANT,A
 VENDING MACHINE
 164 DATAA HIDEOUS TROLL,AN AXE,+,+,+,+,AN ENORMOUS
 GIANT,A DWARF,A CROWBAR
 166 DATAA JOLLY GREEN DRAGON,A BOX,A BULBOUS SPIDE
 R,+,+,A ROPE,+,+,A BOAT
 168 DATASOME OARS,+,+,A CURIOUS PYRAMID,+,+,+,+,+,
 +,+,+,+,+,+,+,+,+,+,+,+,+,
 170 DATASEVERAL CHAIRS,SOME MUGS,+,+,+,A MAGAZINE,
 +,+,+,+,SOME OIL
 172 DATAA BOTTLE OF OIL
 199 REM OB\$(1,0) STUFF
 200 DATA59,159,168,123,0,33,187,78,105,142,163,199
 ,90,250,251,0,0,0,0
 202 DATA0,0,0,0,0,0,0,0,0,26,0,0,2,2,2,0,0,2,0,0,0
 ,0,22,21,19,28,0,33,0,0,0,0

```

204 DATA55,66,0,0,0,0,0,0,0,0,0,61,0,173,0,0,0,119,0
,0,199,186,0,0,203,0,0,0,0
206 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
221,0,0,0,0,0,76,0
209 REM OB%(I,1) STUFF
210 FORI=1TO15:OB%(I,1)=20:NEXT:FORI=16TO107:OB%(I
,1)=0:NEXT
211 RETURN

```

READY.

Line 1 just dimensions all the necessary arrays, while lines 2 to 6 read in all the appropriate data. Line 7 GOSUBs off to line 209, which sets the value of all the objects in the game.

Some of the data might look a bit confusing at first, and in particular the rows of asterisks in line 112. Well, when playing adventures people will type in rude words, so when you come to enter the program yourself you can insert your own words there: I refuse to have any mucky words printed in one of my books!

Some of the object descriptions might look a little peculiar, since they consist of nothing more than plus (' + ') signs. This is just used when we don't want an object to have a description. For example, the words ON and OFF, while theoretically being objects, are not going to exist in any particular location. It would look a bit strange if you were told that you could see an OFF sitting on a rock, for example.

Lines 10-80 then do the donkey work of filing everything away on to disk, and with that done we just have the map left to do.

Map making

Since the map is rather large (253 locations), and you've got to do some work yourself, I won't draw it out here. Each data statement represents one location, and as the REM statement in line 100 tells us, the data is stored in north, south, east, west, up, down format. In other words, the first number refers to which location you'll reach by going north, the second by going south, and so on. A zero indicates that you can't go in that particular direction, while a minus one indicates that you *can* go that way but it's going to be decidedly fatal.


```

10 REM MAP DATA OH MY GOODNESS!
20 DIMP%(253,5)
25 FORI=1TO253:FORJ=0TO5
30 READA:P%(I,J)=A
35 NEXTJ,I
40 PRINT"EVERYTHING READ IN: PRESS SPACE TO MAKE F
   ILE ON DISK.
42 FORI=1TO10:GETSP$:NEXT
45 GETSP$:IFSP$<>" "THEN45
50 OPEN2,8,2,"0:MAP1,S,W"
55 FORI=1TO125:FORJ=0TO5
60 PRINT#2,P%(I,J);CHR$(13);
65 NEXTJ,I
66 CLOSE2:OPEN2,8,2,"0:MAP2,S,W"
67 FORI=126TO253:FORJ=0TO5
68 PRINT#2,P%(I,J);CHR$(13);
69 NEXTJ,I
70 PRINT"FINISHED ... PHEW!
80 END
100 REMROOMS1-10, IN N,S,E,W,U,D FORMAT
101 DATA2,0,5,6,0,0
102 DATA0,0,0,3,0,0
103 DATA5,6,2,4,0,0
104 DATA5,6,7,3,5,6
105 DATA0,4,1,0,0,4
106 DATA4,0,1,0,4,0
107 DATA0,0,4,8,0,0
108 DATA0,0,7,9,0,0
109 DATA0,10,8,0,0,0
110 DATA9,11,0,0,0,0
111 DATA10,0,0,12,0,0:REM 11-20 NOW
112 DATA0,0,11,13,0,0
113 DATA0,0,12,14,0,0
114 DATA0,0,13,15,0,0
115 DATA0,0,14,16,0,0
116 DATA0,0,15,0,0,0
117 DATA0,18,0,0,0,0
118 DATA17,19,0,0,0,0
119 DATA18,26,20,24,0,0
120 DATA0,0,21,19,0,0
121 DATA0,0,22,20,0,0:REM 21-30 NOW
122 DATA0,0,23,21,0,0
123 DATA0,0,0,22,0,0
124 DATA0,0,19,25,0,0
125 DATA0,0,24,0,0,0
126 DATA19,27,0,0,0,27
127 DATA26,28,0,0,26,28
128 DATA27,0,0,0,27,0
129 DATA0,42,36,28,0,0
130 DATA0,35,28,31,0,0
131 DATA0,34,30,32,0,0:REM 31-40 NOW

```

132 DATA0,33,31,0,0,0
 133 DATA32,0,34,0,0,0
 134 DATA31,0,35,33,0,0
 135 DATA30,35,35,34,35,35
 136 DATA0,41,37,29,0,0
 137 DATA0,40,38,36,0,0
 138 DATA0,39,0,37,0,0
 139 DATA38,0,0,40,0,0
 140 DATA37,0,39,41,0,0
 141 DATA36,0,40,42,0,0:REM 41-50 NOW
 142 DATA29,43,41,0,0,0
 143 DATA42,-1,0,0,0,-1:REM -1 MEANS OBLIVION!
 144 DATA-1,50,45,46,0,-1
 145 DATA0,51,0,44,0,0
 146 DATA0,49,44,47,0,0
 147 DATA0,48,46,0,0,0
 148 DATA47,0,49,0,0,0
 149 DATA46,0,50,48,0,0
 150 DATA44,52,51,49,50,50
 151 DATA51,0,0,50,0,0:REM 51-60 NOW
 152 DATA50,0,52,52,52,52
 153 DATA52,54,0,0,0,0
 154 DATA53,0,55,71,0,55
 155 DATA0,0,0,54,55,0
 156 DATA0,0,57,-1,0,-1
 157 DATA0,-1,58,56,0,-1
 158 DATA0,0,59,57,0,0
 159 DATA0,0,60,58,0,0
 160 DATA61,60,60,59,60,60
 161 DATA61,60,62,61,61,61:REM 61-70 NOW
 162 DATA61,67,61,62,62,62
 163 DATA63,66,63,62,62,62
 164 DATA64,65,63,62,62,62
 165 DATA64,70,65,62,63,63
 166 DATA63,69,66,66,67,62
 167 DATA62,68,62,62,62,62
 168 DATA64,68,69,67,67,62
 169 DATA66,68,69,62,62,64
 170 DATA65,60,69,69,69,60
 171 DATA0,0,54,72,0,0:REM71-80 NOW
 172 DATA82,81,71,73,0,0
 173 DATA0,0,72,74,0,0
 174 DATA0,0,73,75,0,0
 175 DATA0,79,74,76,0,79
 176 DATA0,0,0,0,0,0
 177 DATA76,78,79,0,0,78
 178 DATA77,0,80,0,77,0
 179 DATA75,80,0,77,75,0
 180 DATA79,0,0,78,0,0
 181 DATA72,83,0,0,0,0:REM 81-90 NOW
 182 DATA0,72,0,0,0,0

```

183 DATA81,84,0,0,0,0
184 DATA83,91,87,85,0,0
185 DATA0,92,84,86,0,0
186 DATA0,93,0,86,0,0
187 DATA0,90,88,84,0,0
188 DATA0,89,0,87,0,0
189 DATA88,98,0,90,0,0
190 DATA87,99,89,91,0,0
191 DATA84,84,90,92,0,0
192 DATA85,95,90,93,0,0
193 DATA86,94,92,0,0,0
194 DATA93,0,95,0,0,0
195 DATA92,0,96,94,0,0
196 DATA99,91,97,95,0,0
197 DATA90,0,98,96,0,0
198 DATA89,0,0,97,0,0
199 DATA96,100,0,0,0,0
200 DATA99,101,0,0,0,0:REM ROOM ONE HUNDRED!
201 DATA100,102,-1,-1,0,0
202 DATA101,0,103,0,0,0
203 DATA0,0,104,102,0,0
204 DATA0,105,0,103,0,0
205 DATA104,106,0,0,0,0
206 DATA105,0,107,0,0,0
207 DATA0,-1,108,106,0,-1
208 DATA-1,110,109,107,0,-1
209 DATA0,111,0,108,0,0
210 DATA108,0,111,0,0,0
211 DATA109,0,112,110,0,0
212 DATA0,124,113,111,113,113
213 DATA114,113,113,112,113,113
214 DATA214,214,115,114,114,114
215 DATA115,122,116,114,116,114
216 DATA116,123,115,115,115,115
217 DATA117,117,117,116,117,117
218 DATA117,118,115,115,115,115
219 DATA118,119,119,119,119,119
220 DATA120,120,119,120,120,120
221 DATA121,121,120,122,122,122
222 DATA115,121,115,115,115,115
223 DATA118,120,118,122,123,113
224 DATA112,125,0,0,0,125
225 DATA124,0,0,0,124,0
226 DATA125,126,0,0,125,0
227 DATA126,0,0,128,0,0
228 DATA0,-1,127,129,0,-1
229 DATA0,0,128,0,0,0
230 DATA131,132,-1,133,0,-1
231 DATA0,130,0,134,0,0:REM 131 TO 140 NOW
232 DATA130,0,0,135,0,0
233 DATA134,135,130,136,0,0

```

234 DATA-1,133,131,0,0,-1
 235 DATA133,0,132,0,0,0
 236 DATA0,0,133,137,0,0
 237 DATA0,0,136,138,0,0
 238 DATA139,140,137,142,0,0
 239 DATA0,138,0,141,0,0
 240 DATA138,167,0,143,0,0
 241 DATA0,142,139,146,0,0:REM 141 TO 150 NOW
 242 DATA141,143,138,145,0,0
 243 DATA142,0,140,144,0,0
 244 DATA145,0,143,166,0,0
 245 DATA146,144,142,0,0,0
 246 DATA147,145,141,0,0,0
 247 DATA-1,146,0,148,0,-1
 248 DATA-1,0,147,149,0,-1
 249 DATA0,0,148,150,0,0
 250 DATA0,0,149,0,0,0
 251 DATA153,150,0,152,0,0:REM 151 TO 160 NOW
 252 DATA154,0,151,0,0,0
 253 DATA0,151,0,154,0,0
 254 DATA0,152,153,155,0,0
 255 DATA0,0,154,156,0,0
 256 DATA0,0,155,157,0,0
 257 DATA0,158,156,0,0,0
 258 DATA157,159,0,0,0,0
 259 DATA158,160,0,0,0,0
 260 DATA159,161,0,0,0,0
 261 DATA160,0,162,0,0,0:REM 161 TO 170
 262 DATA0,0,163,161,0,0
 263 DATA0,0,164,162,0,0
 264 DATA0,0,165,163,0,0
 265 DATA0,0,166,164,0,0
 266 DATA0,0,0,165,0,0
 267 DATA140,168,0,0,0,0
 268 DATA167,0,0,0,0,0
 269 DATA168,170,0,0,0,0
 270 DATA169,171,0,0,0,0
 271 DATA170,174,172,0,0,0:REM 171 TO 180
 272 DATA0,0,173,171,0,0
 273 DATA0,0,0,172,0,0
 274 DATA171,0,0,0,0,0
 275 DATA174,176,0,0,0,0
 276 DATA175,0,0,0,0,0
 277 DATA0,184,178,176,0,0
 278 DATA0,185,0,177,0,-1
 279 DATA0,182,176,180,0,0
 280 DATA0,181,179,-1,0,-1
 281 DATA180,186,182,0,0,0:REM 181 TO 190
 282 DATA179,187,183,181,0,0
 283 DATA188,176,182,184,183,183
 284 DATA177,189,185,183,0,0

285 DATA178,190,0,184,0,0
 286 DATA181,0,187,0,0,0
 287 DATA182,0,188,186,0,0
 288 DATA183,0,189,187,0,0
 289 DATA184,0,190,188,0,0
 290 DATA185,-1,0,189,0,-1
 291 DATA0,192,0,0,0,0:REM 191 TO 200
 292 DATA191,0,193,0,0,0
 293 DATA0,0,0,192,0,0
 294 DATA195,0,202,193,0,0
 295 DATA196,194,0,0,0,0
 296 DATA197,195,0,0,0,0
 297 DATA198,196,0,0,0,0
 298 DATA-1,197,0,0,0,-1
 299 DATA0,200,0,0,0,0
 300 DATA199,201,0,0,0,0
 301 DATA200,202,0,0,0,0:REM 201 TO 210
 302 DATA201,0,206,194,0,0
 303 DATA0,204,-1,0,0,-1
 304 DATA203,205,-1,0,0,-1
 305 DATA204,206,0,0,0,0
 306 DATA205,0,0,202,0,0
 307 DATA207,209,210,206,207,207
 308 DATA206,212,209,208,208,208
 309 DATA207,213,211,208,209,209
 310 DATA210,211,210,207,210,210
 311 DATA210,214,211,209,211,211:REM 211 TO 220 NOW
 312 DATA208,212,213,212,212,212
 313 DATA209,213,214,212,213,213
 314 DATA211,215,214,213,214,214
 315 DATA214,216,215,215,215,215
 316 DATA215,0,0,0,0,0
 317 DATA-1,0,0,218,0,-1
 318 DATA0,0,217,219,0,0
 319 DATA0,0,218,220,0,0
 320 DATA0,0,219,221,0,0
 321 DATA0,0,220,0,0,0:REM 221 TO 230
 322 DATA223,224,221,226,0,0
 323 DATA0,222,0,225,0,0
 324 DATA222,0,0,227,0,0
 325 DATA0,226,223,0,0,0
 326 DATA225,227,222,-1,0,-1
 327 DATA226,0,224,0,0,0
 328 DATA0,0,226,229,0,-1
 329 DATA231,230,-1,0,0,-1
 330 DATA229,0,0,0,0,0
 331 DATA232,229,0,0,0,0:REM 231 TO 240
 332 DATA233,231,0,0,0,0
 333 DATA0,0,232,0,0,0
 334 DATA236,234,234,234,234,234
 335 DATA234,234,234,238,234,234

```

336 DATA235,234,234,234,234,234
337 DATA234,239,234,234,234,234
338 DATA234,237,234,234,234,234
339 DATA234,234,234,240,234,234
340 DATA241,234,234,234,234,234
341 DATA242,234,234,234,234,234:REM 241 TO 250!
342 DATA234,234,234,243,234,234
343 DATA-1,0,242,0,0,-1
344 DATA0,-1,0,245,0,-1
345 DATA246,0,244,0,0,0
346 DATA247,245,0,0,0,0
347 DATA248,246,0,0,0,0
348 DATA0,247,0,0,0,0
349 DATA0,248,0,0,0,0
350 DATA0,0,249,0,0,0
351 DATA0,249,-1,0,0,-1
352 DATA253,253,253,253,253,253
353 DATA252,252,252,252,252,252

```

READY.

Lines up to and including 35 dimension an array and read all the data in. Then, a rare consideration allows you to press the space bar before everything is filed on to disk. For some reason I split the file up into two separate parts when writing this program, and that is why lines 45 to 69 successively write MAP1 and MAP2 to disk.

And that, dear reader, is the entire program!

Conclusion

It's been a long hard slog typing it all in, although you might have cheated and sent off for the disk: I wouldn't blame you!

'Enormous Cave' started off as just a bit of fun, and graduated into a complete adventure that should (I hope) take a long time to solve. I also hope that it will be enjoyable to play: I know it was enjoyable writing it. That is the beauty of working to a system when writing adventure games. A lot of the hard work is already done for you, and once you've got the framework for various verbs, for filing data, for generating maps, and so on, it does make producing adventures a lot of fun.

With a compiled version you'll see just how fast the program can become, and how we can now produce enormous adventures without really using any of the complex procedures discussed in

Chapters 2, 6 and 7. Imagine what, with a little thought, ingenuity and machine code, you could do.

Just before writing this chapter I read in one of the computer magazines that a couple of manufacturers think that adventures are going to become very, very big news, with the advent of microcomputers with built-in disk drives.

I look forward to seeing you prove them right.

10

A Compiler

Introduction

Don't look at the listing yet, you'll only get alarmed!

First of all, what is a compiler, and why should you want to type this enormous listing in at all? A compiler, which in itself is nothing more than an ordinary machine-code program, takes a clean (i.e. no machine code) Basic program and converts it into a form of machine code: pseudo-code, as it is often called. When this new program is run, it will operate at a much faster pace than in its original Basic form. It won't run as fast as a program originally written in machine code, but it is certainly a vast improvement over poor old Basic.

Thus you can do all the development work on your adventures in Basic, and then when you're satisfied that the program is working correctly you can compile it, to produce a program that operates at a very fast rate indeed. However, you must make sure that your program is perfect and contains no errors: you won't be able to change it once you've compiled it. Instead, any mistakes must be corrected in the original Basic version and then the whole program compiled once more.

The compiler presented here will work with just about everything in Basic, bar one command. That is the TAB command, so if any of your programs to be compiled contain a TAB statement, replace that with the appropriate number of cursor rights.

You'll need to have a disk drive to operate it, and when the program is run it first of all checks that there's a disk in the drive before asking for the file name of the program to be compiled. It's probably best to compile one program per disk, as the compiler

generates a number of files on the disk as it's working. Since the program originally came from an Austrian source, one or two error messages meant nothing to me when I was amending the program, so if you come across anything being printed out in Austrian you can probably assume that something has gone sadly wrong.

Now to the onerous task of typing the listing in.

Compiler listing

B*

	PC	SR	AC	XR	YR	SP			
..	97FE	30	00	00	00	F6			
.									
..	0800	00	1A	0B	01	00	9E	32	30
..	080B	37	36	20	4B	45	4C	4C	4F
..	0810	20	54	4B	45	52	45	20	56
..	081B	31	00	00	00	4C	41	0B	4C
..	0820	B5	0C	66	09	76	09	3B	16
..	082B	24	14	17	14	C0	14	C3	14
..	0830	32	0B	55	0B	EE	11	15	12
..	083B	73	15	66	15	4B	15	4B	15
..	0840	80	15	4F	0B	A9	15	FF	10
..	084B	EE	10	DC	10	CD	10	CD	10
..	0850	26	11	CD	10	CD	10	CD	10
..	085B	CD	10	CD	10	CD	10	CD	10
..	0860	CD	10	1E	11	2D	11	33	11
..	086B	CD	10	C1	10	47	11	C2	11
..	0870	C2	11	BA	11	C2	15	F5	15
..	087B	90	15	BE	0C	AB	12	6A	12
..	0880	6A	12	6A	12	D9	12	BB	12
..	088B	BB	12	EA	12	6A	12	EA	12
..	0890	E2	12	62	11	62	11	57	11
..	089B	AB	0C	2B	12	2B	12	EE	11
..	08A0	EE	11	AE	0C	B1	0C	1A	13
..	08AB	55	13	55	13	25	13	D0	13
..	08B0	25	13	25	13	55	13	55	13
..	08BB	05	13	0E	13	0B	13	11	13
..	08C0	D9	16	DE	16	DA	16	46	17
..	08CB	FA	12	7B	17	7B	17	7B	17
..	08D0	7E	17	7B	17	7B	17	7B	17
..	08DB	7B	17	7B	17	7B	17	7B	17
..	08E0	7B	17	7B	17	7B	17	81	17
..	08EB	27	03	00	00	10	E8	64	0A
..	08F0	6A	53	2B	12	7B	BB	BB	BA
..	08FB	BB	BF	A0	01	2C	A0	02	A9
..	0900	20	24	64	10	05	20	72	0C
..	090B	A9	2D	99	FE	00	84	71	A2

```

.:0910 00 86 62 A5 64 D0 06 A5
.:0918 65 F0 46 A2 02 BD E8 08
.:0920 85 22 BD EC 08 85 23 A0
.:0928 FF 38 C8 A5 65 E5 23 85
.:0930 65 A5 64 E5 22 85 64 B0
.:0938 F1 A5 65 65 23 85 65 A5
.:0940 64 65 22 85 64 A5 62 D0
.:0948 05 98 F0 0C E6 62 98 09
.:0950 30 A4 71 99 FF 00 E6 71
.:0958 E8 E0 04 90 C0 A4 71 A5
.:0960 65 09 30 4C 04 BF A5 0E
.:0968 10 06 20 72 0C 4C BE 0C
.:0970 20 B4 BF 4C BE 0C A5 0E
.:0978 30 03 20 1B 10 A5 64 49
.:0980 FF 85 64 A5 65 49 FF 85
.:0988 65 4C BE 0C A0 00 B1 6F
.:0990 18 71 64 90 03 4C 58 B6
.:0998 20 75 B4 20 7A B6 A4 51
.:09A0 D0 05 A5 50 20 DB B6 A0
.:09A8 02 B1 50 85 23 88 B1 50
.:09B0 85 22 88 B1 50 20 8C B6
.:09B8 A4 70 D0 05 A5 6F 20 DB
.:09C0 B6 4C CA B4 C0 07 B0 C4
.:09C8 A4 65 D0 03 20 B4 0B A5
.:09D0 6F A4 70 20 AA B6 86 6C
.:09D8 B4 6D A0 02 AA 38 E5 61
.:09E0 F0 08 A0 04 90 04 A0 01
.:09E8 A6 61 84 66 A0 FF E8 C8
.:09F0 CA D0 04 A5 66 10 0C B1
.:09F8 6C D1 62 F0 F2 A9 01 B0
.:0A00 02 A9 04 25 3C F0 02 A9
.:0A08 FF AA 4C 63 0E A5 6E 09
.:0A10 7F 25 6A 85 6A A9 69 A0
.:0A18 00 20 5B BC 18 AA E8 D0
.:0A20 01 38 8A 2A 10 DD C0 0A
.:0A28 90 58 C0 0C B0 76 8A 10
.:0A30 2F A0 00 84 69 84 6A 84
.:0A38 6B 84 6E A2 90 20 FA 0A
.:0A40 A5 6C D0 08 A2 88 A5 6D
.:0A48 F0 16 84 6D 30 06 CA 06
.:0A50 6D 2A 10 FA 85 6A A5 6D
.:0A58 85 6B 86 69 84 6C 84 6D
.:0A60 A5 0E 10 03 20 85 10 A4
.:0A68 3C C0 07 90 A0 B9 E9 08
.:0A70 85 55 B9 EE 08 85 56 A5
.:0A78 6E 45 66 85 6F A5 61 6C
.:0A80 55 00 E4 0E D0 12 8A 10
.:0A88 DE A5 3C C9 07 B0 06 20
.:0A90 13 0B 4C 1C 0A 4C C5 0B
.:0A98 8A 10 C5 20 AE 0F A5 0E

```

```

.:0AA0 10 8F 30 E5 8A 30 03 20
.:0AAB CE 0A A5 0E 30 03 20 1B
.:0AB0 10 46 3C B0 0C A5 6C 25
.:0ABB 64 85 64 A5 6D 25 65 90
.:0AC0 0A A5 6C 05 64 85 64 A5
.:0ACB 6D 05 65 85 65 60 A5 69
.:0AD0 30 06 A9 00 85 6B F0 11
.:0ADB 3B E9 90 30 03 4C 4B B2
.:0AE0 AA A5 6A 4A 66 6B EB D0
.:0AEB FA 06 6E 90 02 49 FF 85
.:0AF0 6C A5 6B 90 02 49 FF 85
.:0AFB 6D 60 A5 6C 10 14 85 6E
.:0B00 A5 6C 49 FF 85 6C A5 6D
.:0B0B 49 FF 85 6D E6 6D D0 02
.:0B10 E6 6C 60 A2 00 A5 6C C5
.:0B1B 64 D0 06 A5 6D C5 65 F0
.:0B20 0B CA A5 6C 45 64 30 06
.:0B2B B0 02 A2 01 8A 60 B0 FA
.:0B30 90 FA 20 60 0B A5 39 85
.:0B3B 50 A5 3A 85 51 A0 06 D0
.:0B40 05 20 96 0B A0 0A B9 84
.:0B4B 17 BE 85 17 4C 79 15 20
.:0B50 A5 0B 4C 9E 0C A5 50 85
.:0B5B 39 A5 51 85 3A 4C 9E 0C
.:0B60 A2 05 BD 84 17 95 2D CA
.:0B6B 10 FB A6 2E 86 60 85 5F
.:0B70 E4 30 D0 04 C5 2F F0 13
.:0B7B A0 01 A9 00 CB 91 5F C0
.:0BB0 06 D0 F9 9B 65 5F 90 E6
.:0BBB EB B0 E1 A5 37 A4 3B 85
.:0B90 33 84 34 20 E7 FF 6B AB
.:0B9B 6B A2 FB 9A 4B 9B 4B A9
.:0BA0 00 85 3B 85 11 AD 8C 17
.:0BAB 85 41 AD 8D 17 85 42 60
.:0BB0 A4 65 D0 0E A5 64 20 DB
.:0BBB B6 A6 62 A4 63 A5 61 4C
.:0BC0 C3 B6 A5 61 60 F0 10 4A
.:0BCB B0 45 3B A5 6D E5 65 AA
.:0BD0 A5 6C E5 64 4C E1 0B 1B
.:0BDB A5 6D 65 65 AA A5 6C 65
.:0BE0 64 85 64 86 65 70 01 60
.:0BEB A2 00 86 62 86 63 86 66
.:0BF0 90 05 C6 66 20 72 0C A9
.:0BFB 00 85 0E 85 70 4C DB BC
.:0C00 A5 62 05 63 D0 F1 A5 64
.:0C0B 30 ED A5 66 30 64 60 A5
.:0C10 64 45 6C 85 66 A5 64 10
.:0C1B 03 20 72 0C 20 FA 0A A5
.:0C20 64 A6 65 C5 6C 90 0A A4
.:0C2B 6C 85 6C A5 6D 86 6D AA

```

```

.:0C30 98 85 28 86 29 A9 00 85
.:0C38 62 85 63 85 64 85 65 85
.:0C40 6A 85 6B A5 28 05 29 F0
.:0C48 B7 46 28 66 29 90 19 18
.:0C50 A5 65 65 6D 85 65 A5 64
.:0C58 65 6C 85 64 A5 63 65 6B
.:0C60 85 63 A5 62 65 6A 85 62
.:0C68 06 6D 26 6C 26 6B 26 6A
.:0C70 90 D1 A5 64 49 FF 85 64
.:0C78 A5 65 49 FF 85 65 E6 65
.:0C80 D0 02 E6 64 60 A5 39 A6
.:0C88 3A 85 7A 86 7B 85 3D 86
.:0C90 3E 4C E1 A7 68 68 A5 7A
.:0C98 A6 7B 85 39 86 3A 24 11
.:OCA0 50 18 A5 91 C9 7F D0 12
.:CAB 20 B7 AB 38 B0 04 20 44
.:0CB0 A6 18 20 41 AB 68 68 20
.:0CB8 68 AB A0 00 84 3B E6 39
.:OCC0 D0 0B F0 07 18 65 39 85
.:OCC8 39 90 02 E6 3A A0 00 B1
.:OCD0 39 85 3C 30 47 F0 AE C9
.:OCD8 0E 90 10 0A AA BD 06 08
.:OCE0 85 55 BD 07 08 85 56 C8
.:OCE8 6C 55 00 AB 68 AA 30 0E
.:OCF0 D0 18 68 85 6E 68 85 69
.:OCF8 68 85 6A 68 85 6B 68 85
.:OD00 6C 68 85 6D 20 26 0A 4C
.:OD08 BE 0C 68 85 6F 68 85 70
.:OD10 20 C4 09 4C BE 0C 20 E0
.:OD18 A9 4C 9E 0C C9 E6 B0 0A
.:OD20 24 3C 50 08 A6 0E 86 0C
.:OD28 70 27 F0 EA A6 3B F0 1F
.:OD30 A5 0E D0 03 20 AE 0F A5
.:OD38 65 48 A5 64 48 A6 0E D0
.:OD40 0C A5 63 48 A5 62 48 A5
.:OD48 61 48 A5 66 48 8A 48 A0
.:OD50 01 84 3B A0 01 A2 00 A5
.:OD58 3C 29 3F C9 20 B0 0A 85
.:OD60 47 0A 0A 0A 38 E5 47 B0
.:OD68 3D C9 26 B0 59 C9 24 90
.:OD70 03 4C 6F 0E 29 03 C9 03
.:OD78 90 08 B1 39 E6 39 D0 02
.:OD80 E6 3A AA B1 39 E6 39 D0
.:OD88 02 E6 3A 86 48 85 47 86
.:OD90 60 0A 26 60 0A 26 60 0A
.:OD98 26 60 38 E5 47 85 5F A5
.:ODA0 60 E5 48 AA A5 5F 18 65
.:ODAB 2D 85 5F 8A 65 2E 85 60
.:ODB0 AA A5 5F 69 02 90 01 E8
.:ODB8 85 47 86 48 85 49 86 4A
.:ODC0 20 4C 0F 4C BE 0C 24 3C

```

```

.:0DCB C9 30 90 0C 29 0F 50 02
.:ODD0 09 10 20 63 0E 4C BE 0C
.:ODDB 70 33 C9 2A B0 29 C9 27
.:ODE0 90 06 D0 0E B1 39 AA C8
.:ODEB B1 39 20 63 0E C8 9B 4C
.:ODF0 C4 0C A5 39 69 00 A4 3A
.:ODFB 90 01 C8 20 A2 BB 84 0D
.:OE00 84 0E A9 06 4C C4 0C 20
.:OE0B 37 0E 4C BE 0C C9 2B 90
.:OE10 04 29 07 10 0B B1 39 E6
.:OE1B 39 D0 02 E6 3A 85 61 84
.:OE20 0E E6 39 D0 02 E6 3A A5
.:OE2B 39 85 62 A5 3A 85 63 20
.:OE30 CA B4 A5 61 4C C4 0C B6
.:OE3B 0D B6 0E D0 07 A9 AB A0
.:OE40 AE 4C A2 BB C9 2E 90 0A
.:OE4B 84 0E D0 03 4C 0B AF 4C
.:OE50 4B AF C9 2C 90 0B D0 03
.:OE5B 4C 7B AF 4C 0B AF A5 90
.:OE60 10 01 CA B6 64 85 65 A9
.:OE6B 80 85 0E 0A 85 0D 60 F0
.:OE70 09 B1 39 AA E6 39 D0 02
.:OE7B E6 3A B6 6A B1 39 E6 39
.:OE80 D0 02 E6 3A 0A 26 6A 3B
.:OE8B 65 2F 85 69 A5 6A 65 30
.:OE90 85 6A 8B B1 69 85 6B 65
.:OE9B 2F 85 5F C8 B1 69 AA 65
.:OEA0 30 85 60 8A 05 6B F0 4A
.:OEA8 B1 5F 85 46 8B B1 5F 85
.:OEB0 45 A0 04 B1 5F 85 0B 0A
.:OEB8 69 05 65 5F 85 5B A9 00
.:OEC0 85 71 85 72 65 60 85 59
.:OECB C8 6B 30 17 6B 85 6E 6B
.:OED0 85 69 6B 85 6A 6B 85 6B
.:OEDB 6B 6B 20 CE 0A A5 6D 4B
.:OEE0 A5 6C 4B 6B 85 6C D1 5F
.:OEEB 90 0B D0 06 C8 6B D1 5F
.:OEF0 90 05 4C 45 B2 C8 6B 85
.:OEFB 6D AA A5 72 05 71 1B F0
.:OF00 0A 20 4C B3 8A 65 6D AA
.:OF0B 9B A4 22 65 6C 86 71 85
.:OF10 72 C6 0B D0 B3 8A A6 72
.:OF1B 0A 26 72 24 45 30 10 24
.:OF20 46 30 03 0A 26 72 65 71
.:OF2B AB 8A 65 72 85 72 9B 65
.:OF30 5B AB A5 72 65 59 AA 9B
.:OF3B 4C BB 0D 8A 20 B0 B7 4C
.:OF40 45 0F 20 F3 BC 24 0E 30
.:OF4B 3C 4C D0 BB A0 00 84 0D
.:OF50 84 0E C8 24 3C B1 5F 30
.:OF5B 13 70 07 A5 47 A4 4B 4C

```

```

.:0F60 A2 BB A5 0C 10 03 20 85
.:0F68 10 4C D0 BB 88 B1 5F 10
.:0F70 1A A9 80 85 0E 70 0A B1
.:0F78 47 85 64 C8 B1 47 85 65
.:0F80 60 A5 0C 30 03 20 1B 10
.:0F88 4C CA A9 C6 0D E6 0E 50
.:0F90 08 4C 2C AA D0 FB 4C E0
.:0F98 A9 A5 47 85 64 86 65 B1
.:0FA0 64 85 61 C8 B1 64 85 62
.:0FAB C8 B1 64 85 63 60 24 70
.:0FB0 10 03 20 1B BC A5 61 D0
.:0FBB 04 85 64 F0 20 10 2B 38
.:0FC0 E9 90 10 26 AA A5 64 05
.:0FCB 65 D0 1F A5 63 85 28 A5
.:0FDO 62 4A 66 28 B0 14 EB D0
.:0FDB FB 85 64 A5 28 85 65 24
.:0FE0 66 10 03 20 72 0C A9 80
.:0FEB 85 0E 60 68 18 69 01 85
.:0FF0 55 68 69 00 85 56 68 85
.:0FFB 0E D0 0E 85 70 68 85 66
.:1000 68 85 61 68 85 62 68 85
.:1008 63 68 85 64 68 85 65 6C
.:1010 55 00 A5 0C C5 0E F0 D2
.:1018 AB F0 6A A0 00 24 70 10
.:1020 03 20 1B BC A5 61 30 06
.:1028 F0 8F A9 F0 30 0E 38 E9
.:1030 90 30 09 D0 49 98 10 46
.:1038 A5 62 30 14 AA 06 66 08
.:1040 90 03 20 4D B9 A5 62 28
.:1048 08 6A 66 63 EB D0 FB 28
.:1050 85 64 A5 63 85 65 A9 80
.:1058 85 0E 60 A5 0E 30 0B 24
.:1060 66 30 1B A0 80 20 1D 10
.:1068 30 04 A5 64 30 10 4C 04
.:1070 BB A5 0E 30 03 20 1B 10
.:1078 A6 65 A5 64 F0 42 4C 48
.:1080 B2 A5 0E 10 37 A0 00 84
.:1088 61 84 62 84 63 84 66 A5
.:1090 64 10 05 85 66 20 72 0C
.:1098 A2 90 A5 64 D0 08 A2 88
.:10A0 A5 65 F0 18 84 65 30 06
.:10AB CA 06 65 2A 10 FA 85 62
.:10B0 A5 65 85 63 86 61 84 64
.:10BB 84 65 84 70 A9 00 85 0E
.:10C0 60 20 B0 0B F0 BB A0 00
.:10CB B1 62 4C 28 11 20 81 10
.:10D0 A5 3C 18 69 14 0A 20 D5
.:10DB AF 4C BE 0C A5 0E 30 04
.:10E0 46 66 10 07 A5 64 10 03
.:10EB 20 72 0C 4C BE 0C A5 0E
.:10F0 30 F9 24 70 10 03 20 1B

```

```

.:10FB BC 20 CC BC 4C BE 0C A5
.:1100 0E 10 15 A2 FF A5 64 30
.:1108 06 EB 05 65 F0 01 EB 8A
.:1110 A2 00 AB 10 15 CA 30 12
.:1118 20 2B BC 4C 10 11 20 5B
.:1120 10 A0 00 B1 14 2C A5 CA
.:1128 A2 00 4C D2 0D 20 B0 0B
.:1130 4C 28 11 A5 0E 10 08 20
.:1138 FA 08 20 6F B4 10 0E 20
.:1140 44 11 10 09 20 6B B4 20
.:1148 71 10 20 54 11 A9 01 85
.:1150 0E 4C BE 0C 20 EF B6 20
.:1158 71 10 20 C6 FF 86 13 4C
.:1160 BA 0C 20 E4 FF 46 3C 90
.:1168 07 48 20 CC FF 85 13 6B
.:1170 A6 0D 30 1E C9 30 90 04
.:1178 C9 3A 90 01 BA 29 0F 85
.:1180 65 86 64 24 0E 30 05 20
.:1188 66 0F F0 03 20 CA A9 4C
.:1190 9E 0C AA F0 1A A4 61 8B
.:1198 F0 09 20 54 11 20 2C AA
.:11A0 4C 9E 0C A5 63 C5 32 90
.:11AB F1 8A 91 62 4C 9E 0C A5
.:11B0 61 F0 ED 86 61 20 CA B4
.:11B8 D0 E3 20 71 10 86 6A 20
.:11C0 EB 0F 20 71 10 68 85 0E
.:11C8 68 85 50 68 85 51 20 D4
.:11D0 11 4C BE 0C A0 00 A5 3C
.:11D8 C9 35 90 06 D0 08 BA 4C
.:11E0 2F B7 8A 4C 03 B7 A5 6A
.:11E8 85 65 8A 4C 4B B7 20 5B
.:11F0 10 20 EB 0F 20 71 10 A4
.:11F8 3C C0 4C 90 12 86 49 D0
.:1200 03 AA F0 06 20 EB 0F 20
.:1208 71 10 20 3C B8 D0 03 20
.:1210 27 B8 4C 9E 0C 20 5B 10
.:1218 A5 39 18 69 01 85 7A A5
.:1220 3A 69 00 85 7B 20 30 E1
.:1228 4C 96 0C 88 B1 41 10 05
.:1230 A2 0D 4C 37 A4 AA A5 41
.:1238 69 01 85 62 85 22 A5 42
.:1240 69 00 85 63 85 23 38 8A
.:1248 65 41 85 41 90 02 E6 42
.:1250 86 61 46 3C B0 0E A9 61
.:1258 85 64 A5 65 84 65 20 94
.:1260 0F 4C 9E 0C 20 3B 0F 4C
.:1268 9E 0C A5 0E 30 1D F0 20
.:1270 A4 65 D0 07 A9 19 85 16
.:1278 20 B9 0B A6 61 A0 00 EB
.:1280 CA F0 20 B1 62 20 D2 FF
.:1288 CB D0 F5 20 FD 08 D0 03

```

```

.:1290 20 DD BD A2 00 BD 00 01
.:1298 F0 06 20 D2 FF E8 D0 F5
.:12A0 20 3B AB A5 3C C9 3D 90
.:12AB 3E D0 2E 3B 20 F0 FF 9B
.:12B0 3B E9 0A B0 FC 49 FF 69
.:12BB 01 10 14 20 71 10 46 3C
.:12C0 90 0E 8A 4B 20 F0 FF 85
.:12CB 09 6B 3B E5 09 90 1B AA
.:12D0 E8 CA F0 13 20 3B AB D0
.:12DB FB 20 D7 AA A5 3C C9 43
.:12E0 90 05 20 CC FF 85 13 4C
.:12EB 9E 0C 20 71 10 20 C9 FF
.:12F0 86 13 A5 3C C9 44 90 EF
.:12FB B0 DF 20 71 10 8A EA 20
.:1300 CC E1 4C 9E 0C A9 40 2C
.:130B A9 B0 05 11 D0 07 A9 BF
.:1310 2C A9 7F 25 11 85 11 4C
.:131B BE 0C A5 39 85 3D A5 3A
.:1320 85 3E 4C 9E 0C 20 F9 AB
.:132B A5 13 F0 06 A5 90 29 03
.:1330 D0 1D AD 00 02 D0 1B A5
.:133B 13 D0 0E 24 11 10 42 20
.:1340 72 AB 85 39 84 3A 4C B1
.:134B 0C A5 90 29 40 F0 D6 86
.:1350 7A 84 7B D0 16 A5 43 85
.:135B 7A A9 02 85 7B 20 79 00
.:1360 D0 09 A5 13 D0 BF 20 45
.:136B AB D0 BA 20 73 00 24 0D
.:1370 30 3B 20 42 0F 20 79 00
.:137B F0 12 C9 2C F0 0E 20 62
.:1380 AB A5 3D A4 3E 85 39 84
.:138B 3A 4C 9E 0C A4 7A 84 43
.:1390 A6 3C E0 55 B0 11 A6 13
.:139B F0 07 20 CC FF 85 13 F0
.:13A0 06 AA F0 03 20 F4 AC 4C
.:13AB 9E 0C 85 07 C9 22 F0 07
.:13B0 A9 3A 85 07 A9 2C 1B 85
.:13BB 0B A5 65 4B A5 7A A4 7B
.:13C0 69 00 20 BD B4 A6 71 86
.:13CB 7A 6B 20 94 0F 4C 75 13
.:13D0 A5 3C 69 8B 85 3D A2 05
.:13DB 86 55 A0 AF AD 01 02 91
.:13E0 3C A9 40 A0 AC 91 3C AE
.:13EB 02 02 A9 01 20 0F 14 0A
.:13F0 D0 0F 90 0D 2A 20 0F 14
.:13FB 30 07 CA F0 0B C9 00 F0
.:1400 E9 C6 55 D0 D5 4C AE 0C
.:140B C5 65 D0 F5 4C BE 0C 91
.:1410 3C 4A 91 3C B1 3C 60 A5
.:141B 0E 85 0C 20 EB 0F 20 12
.:1420 10 4C 3A 14 A5 0E 85 0C

```



```

.:142B 30 09 A9 BC A0 B9 20 A2
.:1430 BB 30 07 A2 01 86 65 CA
.:143B 86 64 20 0F BC 20 EB 0F
.:1440 20 12 10 20 97 14 D0 06
.:144B 8A 18 65 0B AA 9A BA E0
.:1450 50 B0 03 4C 35 A4 A5 0C
.:145B F0 0F A5 65 4B A5 64 4B
.:1460 A5 6D 4B A5 6C 4B 4C 85
.:146B 14 A5 66 09 7F 25 62 85
.:1470 62 A9 7C A0 14 85 22 84
.:147B 23 4C 43 AE 20 FC BB 20
.:1480 2B BC 20 3B AE A5 3A 4B
.:148B A5 39 4B A5 4A 4B A5 49
.:1490 4B A5 0C 4B 4C 9E 0C BA
.:149B EB EB BD 01 01 0A D0 1F
.:14A0 A0 09 B0 02 A0 10 B4 0B
.:14AB A5 4A F0 13 DD 03 01 D0
.:14B0 07 A5 49 DD 02 01 F0 07
.:14BB 8A 1B 65 0B AA D0 DB 60
.:14C0 8B 84 4A 20 97 14 F0 03
.:14CB 4C 30 AD 9A BD 03 01 85
.:14D0 4A BD 02 01 85 49 A0 01
.:14DB 6B 4B 85 0E 30 27 BA 1B
.:14E0 69 06 4B 69 06 85 24 6B
.:14EB 20 A2 BB BA BD 0B 01 85
.:14F0 66 85 0C A5 49 A4 4A 20
.:14FB 67 BB 20 D0 BB A0 01 20
.:1500 5D BC 4C 2D 15 BD 07 01
.:150B 1B 71 49 91 49 85 65 BD
.:1510 06 01 10 02 A0 FF B4 0C
.:151B A0 00 71 49 91 49 85 64
.:1520 BD 09 01 85 6D BD 0B 01
.:152B 85 6C 20 13 0B 3B BA E5
.:1530 0C F0 0D BD 05 01 85 3A
.:153B BD 04 01 85 39 4C 9E 0C
.:1540 8A 1B 65 0B AA 9A D0 F5
.:154B C6 3B B1 39 85 6B 20 71
.:1550 10 8A F0 05 0A C5 6B 90
.:155B 04 A5 6B D0 62 AB 46 3C
.:1560 B0 11 A5 6B D0 02 A9 03
.:156B 65 39 4B A5 3A 69 00 4B
.:1570 A9 BD 4B B1 39 AA CB B1
.:157B 39 85 39 86 3A 4C CD 0C
.:1580 A9 FF 85 4A 20 97 14 9A
.:158B 6B C9 BD F0 13 4C E0 AB
.:1590 6B 85 49 6B 85 4A A0 00
.:159B 6B 91 49 CB C0 05 D0 FB
.:15A0 6B 85 3A 6B 85 39 4C CD
.:15AB 0C C6 3B A5 0E 30 0B A5
.:15B0 61 F0 0A A9 02 D0 0B A5
.:15BB 64 05 65 D0 F6 B1 39 4C

```

```

.:15C0 C4 0C B1 39 65 2D 85 64
.:15C8 C8 B1 39 65 2E 85 65 C8
.:15D0 B1 39 65 2D 85 6C C8 B1
.:15D8 39 65 2E 85 6D A9 05 65
.:15E0 39 85 6A A5 3A 69 00 85
.:15E8 6B B9 6A 00 91 64 88 10
.:15F0 FB A0 05 D0 C8 C8 B1 39
.:15F8 65 2D 85 47 88 B1 39 65
.:1600 2E 85 48 A5 39 69 03 48
.:1608 A5 3A 69 00 48 B1 47 D0
.:1610 03 4C AE B3 85 3A 88 B1
.:1618 47 85 39 A0 02 B1 47 85
.:1620 49 C8 B1 47 85 4A C8 B1
.:1628 49 48 88 10 FA A5 4A 48
.:1630 A5 49 48 A5 0E 20 64 0F
.:1638 4C 9E 0C B1 39 84 0C 85
.:1640 0B C8 B1 39 85 45 C8 B1
.:1648 39 85 46 A5 31 85 5F 38
.:1650 E5 2F 85 69 A5 32 85 60
.:1658 E5 30 AA 05 69 F0 21 A0
.:1660 05 18 B1 39 65 2F 85 6B
.:1668 88 B1 39 65 30 85 6C A0
.:1670 01 B1 6B 88 11 6B D0 5E
.:1678 A5 69 91 6B C8 8A 91 6B
.:1680 20 94 B1 20 08 A4 A0 00
.:1688 84 3B 84 72 A2 05 A5 45
.:1690 91 5F 10 01 CA C8 A5 46
.:1698 91 5F 10 02 CA CA 86 71
.:16A0 A5 0B A0 04 91 5F D0 03
.:16A8 20 EB 0F 84 22 20 5B 10
.:16B0 A4 22 A6 65 C8 EB D0 03
.:16B8 18 69 01 91 5F C8 8A 91
.:16C0 5F 20 4C B3 A4 22 86 71
.:16C8 85 72 C6 0B D0 DA 20 AA
.:16D0 B2 A9 06 4C C4 0C 4C 4D
.:16D8 B2 88 84 0A A0 01 20 15
.:16E0 17 46 0C AA F0 15 20 5A
.:16E8 E2 C6 0B F0 0E 20 2E 17
.:16F0 86 BA C6 0B F0 05 20 2E
.:16F8 17 86 B9 20 01 17 4C 9E
.:1700 0C A5 3C C9 5E 90 05 D0
.:1708 09 4C 59 E1 20 6F E1 4C
.:1710 AE A7 4C 6F E1 A2 00 86
.:1718 B7 86 B9 86 90 84 0C 84
.:1720 BA 84 7B B1 39 85 0B E6
.:1728 39 D0 02 E6 3A 60 68 18
.:1730 69 01 85 45 68 69 00 85
.:1738 46 46 0C B0 03 20 EB 0F
.:1740 20 71 10 6C 45 00 20 15
.:1748 17 20 2E 17 86 88 C6 0B
.:1750 F0 20 20 2E 17 86 BA E0

```

```

.:175B 03 90 02 C6 B9 C6 0B F0
.:1760 11 20 2E 17 B6 B9 C6 0B
.:1768 F0 08 20 EB 0F C6 0D 20
.:1770 5A E2 20 C1 E1 4C 9E 0C
.:177B 4C 08 AF 4C 08 AF 4C 08
.:1780 AF 4C 08 AF 46 3B D1 3D
.:1788 D1 3D 90 17 9B 17 99 17
.:1790 BB 10 01 DA AA 00 00 16
.:179B FF 15 B1 AB 90 50 20 00
.:17A0 00 17 B1 AB 90 50 21 00
.:17AB 00 17 FE 33 3E 59 B0 C0
.:17B0 B0 C1 B0 C2 B0 C3 B0 C4
.:17BB B0 C5 E7 10 20 20 20 20
.:17C0 20 20 20 20 20 20 20 20
.:17CB 20 20 20 20 C6 B6 B6 07
.:17D0 B6 07 B6 07 C6 B6 B6 07
.:17DB B6 07 B6 A6 3F 34 07 C6
.:17E0 B0 C7 B0 C8 B0 C9 A6 2D
.:17EB 2E A6 2E 2E A7 01 00 09
.:17F0 07 A7 01 1A 0B CA BA A6
.:17FB BD 07 CB A7 79 2C CC B0
.:1800 CD BE 33 3E E9 30 CE 1A
.:180B 36 E4 A7 09 C4 10 01 C2
.:1810 B0 00 07 A7 07 08 10 01
.:181B C3 B0 00 09 A7 07 08 10
.:1820 01 C4 B0 00 0B A7 03 EB
.:182B 10 01 C7 B0 00 0D A7 01
.:1830 2C 10 01 CB B0 00 0F A7
.:183B 01 2C 10 01 CB B1 00 11
.:1840 A6 2B 10 01 C9 B0 00 13
.:184B A6 2B 10 01 C9 B1 00 15
.:1850 A6 2B 10 01 C9 B2 00 17
.:185B FE 10 01 49 B0 00 19 FE
.:1860 10 01 49 B1 00 1B B7 10
.:186B 01 46 B0 00 1D BF 61 BF
.:1870 BB BF 60 03 E7 1B 13 11
.:187B 11 11 11 11 11 11 11 11
.:1880 11 11 11 11 11 11 11 11
.:188B 11 11 11 11 11 11 CF 1A
.:1890 35 11 90 B1 02 1F 05 19
.:189B 1B A7 91 B0 02 1F 05 19
.:18A0 39 9C 93 D2 19 1B D9 BF
.:18AB 42 E9 49 BE 07 43 50 E7
.:18B0 15 13 11 11 11 11 11 11
.:18BB 11 D0 52 4F 47 52 41 4D
.:18C0 20 4E 41 4D 45 3A 3C 93
.:18CB 53 EA 43 2F 93 BE 34 07
.:18D0 D2 EE 30 30 30 30 30 30
.:18DB E6 93 BE 34 D4 BF 61 BF
.:18E0 BB BF 60 03 B2 61 93 EC
.:18EB 2C 50 2C 52 07 B2 BB B2

```

```

.:18F0 60 04 1A 33 D6 B2 48 95
.:18F8 47 B2 48 95 47 BF 42 E9
.:1900 53 8E 07 EB 3A 5A 2F 07
.:1908 94 07 43 BF 42 E9 53 8E
.:1910 07 EB 3A 50 2F 07 94 07
.:1918 43 B4 61 8E EB 3A 50 2F
.:1920 07 94 07 EC 2C 53 2C 57
.:1928 07 B4 BB B4 60 04 1A 33
.:1930 D6 BF 42 E9 53 8E 07 EB
.:1938 3A 44 2F 07 94 07 43 1A
.:1940 36 E4 B0 D6 B0 D7 B0 DB
.:1948 B0 D9 EB 13 11 11 3C BD
.:1950 40 E7 0A D0 52 4F 47 52
.:1958 41 4D 20 3A 20 3C 93 3E
.:1960 9A 1F 05 E9 13 3E 9B 1F
.:1968 0D EE 13 50 41 53 53 30
.:1970 3E 19 19 7C EE 13 50 41
.:1978 53 53 31 3E B0 C1 B0 DC
.:1980 B0 DD 90 B4 04 9A B0 02
.:1988 0D 1F 08 F5 33 DE 1A 2D
.:1990 39 8A A6 56 07 18 3A 85
.:1998 B0 02 1F 05 19 2D CB 86
.:19A0 B3 A6 FF 36 32 86 B4 A6
.:19A8 FF 36 32 A7 01 00 09 07
.:19B0 DF A0 20 9F A7 7F FF 08
.:19B8 E4 05 A0 20 9C E4 04 A0
.:19C0 20 B1 07 E0 20 B3 C2 EA
.:19C8 13 11 3C 9F 3E A0 21 B0
.:19D0 02 1F 08 9F E0 22 B1 E0
.:19D8 21 8A 18 3A EB DE B0 E0
.:19E0 23 83 B0 02 84 B0 01 0C
.:19E8 1F 05 19 1A 12 83 1F 05
.:19F0 19 19 91 EB DE B0 E0 23
.:19F8 84 B0 02 1F 05 19 1A 35
.:1A00 8A 18 3A A0 24 1F 05 19
.:1A08 19 91 84 B0 02 1F 05 19
.:1A10 1A 35 8A 18 3A 83 1F 05
.:1A18 19 19 91 84 B0 02 1F 05
.:1A20 19 1A 35 8A 18 3A 83 1F
.:1A28 05 19 19 91 84 1F 05 19
.:1A30 19 ED 19 1A B9 B0 E0 25
.:1A38 81 A6 B0 04 1F 05 19 1B
.:1A40 18 81 A6 7F 08 C7 87 A6
.:1A48 23 01 1F 05 19 1A A7 87
.:1A50 1B 16 1B 64 1B BA 1C 58
.:1A58 1C 88 1D 1E 1D 40 1D C3
.:1A60 1E 2E 1B 29 1E 6D 87 BA
.:1A68 08 1B 16 1E D7 1E F0 1B
.:1A70 8C 1E 73 1B 92 1F 3C 1B
.:1A78 85 20 66 21 06 21 AB 87
.:1A80 F4 08 1B 16 21 AF 21 B6

```

```

.:1A8B 22 34 21 66 22 FE 23 4F
.:1A90 1B 9E 1B 9E 1B 9B 23 04
.:1A9B 87 FE 0B 1B 0C 21 7C 24
.:1AA0 02 24 94 24 B5 1B 7E 87
.:1AAB A6 4B 0B C7 87 B0 04 1F
.:1AB0 05 19 1B 7B 87 1B 04 25
.:1ABB 14 E7 0A 45 5B 54 45 4E
.:1AC0 53 49 4F 4E 53 E0 26 1A
.:1ACB 34 9C B0 33 B1 33 07 DE
.:1AD0 A6 3A E0 27 8A B3 07 1B
.:1ADB 3A B1 A0 27 02 B1 B0 02
.:1AE0 0D 1F 05 19 1A FB B1 A6
.:1AEB 22 02 1F 06 A6 22 E0 27
.:1AF0 9E B1 33 07 DE 19 1A D4
.:1AFB B1 A6 22 02 1F 09 A6 3A
.:1B00 E0 27 19 1A F0 9E E9 3A
.:1B0B 07 A6 9E 33 07 ED 33 32
.:1B10 32 30 3A 07 DE 19 1B 6F
.:1B1B B1 A6 41 04 B1 A6 5A 01
.:1B20 0D 1F 05 19 1A B9 19 1B
.:1B2B 2C 8A 1B 3A 1A 29 61 A0
.:1B30 29 32 A6 40 0D 33 A0 29
.:1B3B B2 A6 FF 36 07 E0 2B A0
.:1B40 2B E0 2A B1 A6 B2 05 1F
.:1B4B 05 19 1B 7B 8A 1B 3A 1A
.:1B50 25 2B A0 2B A0 2A 05 1F
.:1B5B 05 19 1B F4 9E A0 2B 07
.:1B60 DE 19 1B 6F A6 4F E0 2C
.:1B6B 8A 1B 3A A0 2C 33 DE 1A
.:1B70 2D 39 84 1F 05 19 19 ED
.:1B7B 1A 33 F0 19 19 ED A6 4E
.:1B80 E0 2C 19 1B 6B A6 49 E0
.:1B8B 2C 19 1B 6B FE E0 2C 19
.:1B90 1B 6B FD E0 2C 19 1B 6B
.:1B9B F5 E0 2C 19 1B 6B E7 12
.:1BA0 55 4E 45 52 4C 41 55 42
.:1BAB 54 45 52 20 42 45 46 45
.:1BB0 4B 4C E0 26 1A 34 B1 19
.:1BBB 19 ED 8A 1B 3A B1 E0 2D
.:1BC0 1A 29 61 A0 29 E0 2E A0
.:1BCB 29 32 A6 40 0D 33 A0 29
.:1BD0 B2 A6 FF 36 07 E0 2B A0
.:1BDB 2B E0 2A B1 A6 B2 05 1F
.:1BE0 05 19 1B 7B 8A 1B 3A 1A
.:1BEB 25 2B A0 2B A0 2A 02 1F
.:1BFO 05 19 1B FA 1A 34 67 19
.:1BFB 19 ED 9E A0 2B 07 DE B1
.:1C00 A6 A4 05 1F 05 19 1B 7B
.:1C0B 1A 2D 39 A0 2E E0 2B 8A
.:1C10 1B 3A 1A 25 23 A0 2B A0
.:1C1B 2A 05 1F 05 19 1B F4 9E

```

```

.:1C20 E0 2E B1 A6 A9 05 1F 0C
.:1C28 A0 2B F1 33 07 E0 2B 19
.:1C30 1C 4F A0 2B F2 33 07 E0
.:1C38 2B 8A 1B 3A 1A 25 23 A0
.:1C40 2E 9E 07 E0 2E A0 2B A0
.:1C48 2A 05 1F 05 19 1B F4 A0
.:1C50 2E A0 2B 07 DE 19 1B 6F
.:1C58 F3 E0 2C 8A 1B 3A 84 1F
.:1C60 05 19 1B 6B B1 E0 2D 1A
.:1C68 29 61 A0 29 F4 33 07 DE
.:1C70 1A 2D 39 84 1F 05 19 19
.:1C78 ED B1 A6 2C 05 1F 05 19
.:1C80 1B 7B 8A 1B 3A 19 1C 64
.:1C88 A0 2F B0 02 1F 17 B5 61
.:1C90 8E EB 3A 44 2F 07 94 07
.:1C98 EC 2C 53 2C 57 07 B5 B8
.:1CA0 B5 60 04 B0 E0 30 EB E0
.:1CAB 31 8A 1B 3A A6 3A E0 27
.:1CB0 A6 2C E0 32 19 1C BC 8A
.:1CB8 B3 07 1B 3A B1 A0 27 02
.:1CC0 B1 A0 32 02 0D B1 B0 02
.:1CC8 0D 1F 05 19 1C EE B1 A6
.:1CD0 22 02 1F 0C B0 E0 27 A6
.:1CDB 22 E0 32 19 1C B7 A0 30
.:1CE0 B1 07 E0 30 A0 31 B1 33
.:1CE8 07 E0 31 19 1C B7 B1 A6
.:1CF0 22 02 1F 0D A6 3A E0 27
.:1CF8 A6 2C E0 32 19 1C B7 B5
.:1D00 42 A0 30 33 3C A0 31 3C
.:1D08 45 A0 2F A0 30 07 B1 07
.:1D10 E0 2F B1 A6 2C 02 1F 05
.:1D18 19 1C A3 19 19 ED 8A 1B
.:1D20 3A 1A 25 23 A0 2B 1F 05
.:1D28 19 1B F4 9E A6 4B 33 07
.:1D30 DE B1 A6 2C 05 1F 05 19
.:1D38 1B 7B 8A 1B 3A 19 1D 6B
.:1D40 8A 1B 3A A6 50 33 DE B1
.:1D48 A6 22 05 1F 05 19 1D 6B
.:1D50 1A 27 65 B1 A6 3B 05 1F
.:1D58 05 19 1B 7B 9E A0 29 07
.:1D60 A6 3C 33 07 DE 8A 1B 3A
.:1D68 1A 2D 39 EB DE 1A 29 61
.:1D70 A6 53 E0 2C A0 29 32 A6
.:1D78 E6 02 1F 07 A6 AF 33 E0
.:1D80 29 B1 A6 2C 05 1F 0E 9E
.:1D88 A0 29 07 A0 2C 33 07 DE
.:1D90 19 1B 6F A0 2C B2 07 E0
.:1D98 2C 9E A0 29 07 A0 2C 33
.:1DA0 07 DE 1A 2D 39 EB DE 8A
.:1DAB 1B 3A 1A 29 61 A6 51 E0
.:1DB0 2C B1 A6 2C 05 1F 05 19

```

```

.:1DBB 1D 74 A0 2C B4 07 E0 2C
.:1DC0 19 1D 93 8A 18 3A B1 0E
.:1DCB E0 33 EB DE 1A 29 61 A0
.:1DD0 33 B0 04 1F 05 19 1E 1C
.:1DD8 A0 33 A4 0B A6 FF 0C E0
.:1DE0 34 A0 33 A0 34 E4 0B 9E
.:1DEB F0 33 07 A0 34 33 07 A0
.:1DF0 35 07 DE A0 33 B2 09 B7
.:1DFB 07 E0 36 A0 36 A7 01 00
.:1E00 0A 21 E0 37 A0 36 A0 37
.:1E08 A7 01 00 09 0B E0 34 9E
.:1E10 A0 37 33 07 A0 34 33 07
.:1E18 DE 1A 2D 39 84 1F 05 19
.:1E20 19 ED B1 A6 2C 05 1F 05
.:1E2B 19 1B 7B 19 1D C3 8A 1B
.:1E30 3A 1A 29 61 A6 4B E0 2C
.:1E38 A0 2B 1F 06 A6 4A E0 2C
.:1E40 A0 29 32 A6 E6 02 1F 07
.:1E4B A6 AF 33 E0 29 9E A0 29
.:1E50 07 A0 2C 33 07 DE 1A 2D
.:1E5B 39 EB DE 84 1F 05 19 19
.:1E60 ED B1 A6 2C 05 1F 05 19
.:1E6B 1B 7B 19 1E 2E F9 E0 2C
.:1E70 19 1E 76 FA E0 2C 8A 1B
.:1E7B 3A B1 A6 30 04 B1 A6 39
.:1E80 01 0D 1F 05 19 1B 7B 1A
.:1E8B 2D 05 A0 39 31 21 E0 3B
.:1E90 A0 3B B0 04 A0 3B AB 90
.:1E9B 7A 00 00 00 01 0D 1F 05
.:1EA0 19 1B 7B A0 3B E0 37 A0
.:1EAB 37 A7 01 00 0A 21 E0 3B
.:1EB0 A0 37 A0 3B A7 01 00 09
.:1EBB 0B E0 34 A0 3A 9C B1 07
.:1EC0 E4 03 A0 3A B1 07 E0 3A
.:1ECB A0 2C 33 A0 3B 33 07 A0
.:1ED0 34 33 07 DE 19 1B 6F F5
.:1EDB 33 DE 1A 2D 39 8A 1B 3A
.:1EE0 F9 E0 2C 84 1F 09 A0 22
.:1EEB E0 3B 19 1E A3 19 1E 79
.:1EF0 8A 1B 3A 1A 25 23 B1 A6
.:1EFB B9 02 1F 05 19 1F 1D B1
.:1F00 A6 A7 05 1F 05 19 1B 7B
.:1F0B 8A 1B 3A B1 A6 2F 01 B1
.:1F10 A6 3A 04 0C 1F 09 82 B1
.:1F1B 0B C2 A6 89 C1 1A 2D 39
.:1F20 A0 3A 9C B1 07 E4 03 A0
.:1F2B 3A B1 07 E0 3A FF 33 A6
.:1F30 FF 33 07 DE 1A 2D 39 EB
.:1F3B DE 19 19 ED 8A 1B 3A 83
.:1F40 1F 05 19 19 91 B1 A6 2A
.:1F4B 02 1F 05 19 1F 5C 8A 1B

```

```

.:1F50 3A B3 B0 02 1F 05 19 1F
.:1F58 4E 19 19 91 BA 18 3A B3
.:1F60 1F 05 19 19 91 B1 A6 2A
.:1F68 05 1F 05 19 1F 4E BA 18
.:1F70 3A B3 1F 05 19 19 91 B1
.:1F78 A6 45 02 1F 08 B1 E0 3B
.:1F80 19 1F 4E B1 A6 49 05 B1
.:1F88 A6 53 05 0C 1F 05 19 1F
.:1F90 4E A6 59 E0 2C B1 A6 49
.:1F98 02 1F 06 A6 5B E0 2C BA
.:1FA0 18 3A B3 1F 05 19 19 91
.:1FAB B1 A6 45 02 1F 05 19 1F
.:1FB0 C0 A0 2C B1 07 E0 2C B1
.:1FBB A6 41 05 1F 05 19 1F CA
.:1FC0 A0 2C 33 DE 1A 2D 39 19
.:1FC8 1F 4E B1 A6 50 05 1F 05
.:1FDO 19 1F 4E B1 E0 37 B5 A0
.:1FDB 37 11 BA 18 3A B1 A6 30
.:1FE0 08 E0 2C A0 2C B0 04 A0
.:1FEB 2C B9 01 0D 1F 05 19 1F
.:1FF0 4E A0 37 A0 2C E4 09 13
.:1FFB B2 B1 A4 09 B2 A4 09 07
.:2000 B3 A4 09 08 0B E0 2C A6
.:2008 47 E0 38 B2 B4 A4 09 B5
.:2010 A4 09 07 B4 08 0B E0 34
.:2018 A0 34 B8 04 1F 06 A6 71
.:2020 E0 38 A6 A6 33 A0 2C 33
.:2028 07 A6 A7 33 07 B2 33 07
.:2030 B2 33 07 F7 33 07 A6 A6
.:2038 33 07 DE 9E A0 38 33 07
.:2040 A6 A7 33 07 B2 33 07 B1
.:2048 33 07 F7 33 07 A6 A6 33
.:2050 07 A0 34 33 07 DE 9E A6
.:2058 B3 33 07 F7 33 07 A6 54
.:2060 33 07 DE 19 1F C4 BA 18
.:2068 3A 1A 25 23 A0 2B 1F 05
.:2070 19 1B F4 FC E0 2C B1 A6
.:2078 B9 02 1F 08 FB E0 2C 19
.:2080 20 8B B1 A6 BD 05 1F 05
.:2088 19 1B 78 1A 2D 39 A0 2C
.:2090 33 DE B0 E0 2C EB E0 29
.:2098 BA 18 3A 1A 2D 05 A0 39
.:20A0 31 21 E0 38 A0 38 B0 04
.:20AB A0 38 AB 90 7A 00 00 00
.:20B0 01 0D 1F 05 19 1B 7B A0
.:20BB 3B E0 37 A0 37 A7 01 00
.:20C0 0A 21 E0 38 A0 37 A0 38
.:20CB A7 01 00 09 0B E0 34 A0
.:20D0 2C B2 07 E0 2C A0 29 A0
.:20DB 3B 33 07 A0 34 33 07 E0
.:20E0 29 A0 3A 9C A0 2C 07 E4

```



```

.:20EB 03 A0 3A B1 07 E0 3A B1
.:20F0 A6 2C 02 1F 05 19 20 9B
.:20FB 9E A0 2C B2 07 33 07 A0
.:2100 29 07 DE 19 1B 6F 8A 1B
.:210B 3A 1A 25 23 9E E0 2B B1
.:2110 A6 2C 05 1F 05 19 1B 7B
.:211B A0 2B 1F 05 19 1B F4 8A
.:2120 1B 3A 1A 25 23 9E A0 2B
.:212B 07 E0 2B A0 2B 1F 05 19
.:2130 1B F4 A6 4D E0 2C 84 1F
.:213B 09 A6 4C E0 2C 19 21 5C
.:2140 B1 A6 2C 05 1F 05 19 1B
.:214B 7B 8A 1B 3A 1A 25 23 9E
.:2150 A0 2B 07 E0 2B A0 2B 1F
.:215B 05 19 1B F4 A0 2B A0 2C
.:2160 33 07 DE 19 1B 6F F7 E0
.:216B 2C 8A 1B 3A 1A 25 23 9E
.:2170 E0 2B A0 2B 1F 05 19 1B
.:217B F4 19 21 40 8A 1B 3A 1A
.:2180 25 23 A0 2B 1F 05 19 1B
.:218B F4 9E FB 33 07 DE 84 1F
.:2190 05 19 21 9F 9E B1 33 07
.:219B DE 8A 1B 3A 19 21 8E 9E
.:21A0 A6 3A 33 07 DE 19 1B 6F
.:21AB A6 5D E0 2C 19 21 BA A6
.:21B0 5E E0 2C 19 21 BA A6 5F
.:21BB E0 2C A0 2C 33 E0 2B B0
.:21C0 E0 3C 8A 1B 3A 84 1F 05
.:21CB 19 22 2A 1A 25 23 A0 2B
.:21D0 B0 02 1F 05 19 1B F4 9E
.:21DB A0 2B 07 E0 2B B1 E0 3C
.:21E0 84 1F 05 19 22 2A B1 A6
.:21EB 2C 05 1F 05 19 1B 7B 8A
.:21F0 1B 3A 1A 25 23 A0 2B 1F
.:21FB 05 19 1B F4 9E A0 2B 07
.:2200 E0 2B B2 E0 3C 84 1F 05
.:220B 19 22 2A B1 A6 2C 05 1F
.:2210 05 19 1B 7B 8A 1B 3A 1A
.:221B 25 23 A0 2B 1F 05 19 1B
.:2220 F4 9E A0 2B 07 E0 2B B3
.:222B E0 3C A0 2B A0 3C 33 07
.:2230 DE 19 1B 6F 8A 1B 3A B1
.:223B A6 A5 05 1F 05 19 1B 7B
.:2240 B1 E0 2D 8A 1B 3A B1 A6
.:224B B0 0D E0 3D B1 E0 3E 1A
.:2250 29 67 B1 A6 2B 05 A0 3F
.:225B B0 01 0D 1F 05 19 1B 7B
.:2260 A0 2B 1F 05 19 1B F4 A0
.:226B 36 B7 09 B2 07 E0 36 A0
.:2270 36 A7 01 00 0A 21 E0 37
.:227B A0 36 A0 37 A7 01 00 09

```

```

.:2280 08 E0 34 A6 37 33 A0 34
.:2288 33 07 A0 37 33 07 E0 28
.:2290 B1 E0 2D 8A 18 3A 1A 29
.:2298 61 81 A6 29 05 A0 3F B0
.:22A0 01 0D 1F 05 19 1B 7B A0
.:22A8 2B 1F 05 19 1B F4 A0 36
.:22B0 B7 09 B2 07 E0 36 A0 36
.:22B8 A7 01 00 0A 21 E0 37 A0
.:22C0 36 A0 37 A7 01 00 09 08
.:22C8 E0 34 A0 2B A0 34 33 07
.:22D0 A0 37 33 07 E0 28 8A 18
.:22D8 3A 81 A6 B2 05 1F 05 19
.:22E0 1B 7B 8A 18 3A 1A 25 23
.:22E8 9E 2F E0 34 A0 2B A0 34
.:22F0 B7 07 33 07 9E 07 A6 39
.:22F8 33 07 DE 19 1B 6F B1 E0
.:2300 3C 19 23 07 B0 E0 3C 8A
.:2308 18 3A 1A 25 23 A0 2B 1F
.:2310 05 19 1B F4 84 B0 02 1F
.:2318 05 19 23 31 A6 42 E0 2C
.:2320 A0 3C 1F 06 A6 44 E0 2C
.:2328 9E A0 2C 33 07 DE 19 1B
.:2330 6F 81 A6 2C 05 1F 05 19
.:2338 1B 7B 8A 18 3A 84 1F 05
.:2340 19 23 1C 9E A6 42 33 07
.:2348 DE 1A 2D 39 19 23 5F B0
.:2350 E0 3C 8A 18 3A 84 1F 09
.:2358 A6 3F 33 DE 19 1B 6F EB
.:2360 DE 81 A6 A3 02 1F 0A A6
.:2368 41 33 E0 2B 19 23 7D 81
.:2370 A6 A6 05 1F 05 19 23 97
.:2378 A6 40 33 E0 2B 8A 18 3A
.:2380 1A 25 23 81 A6 29 05 1F
.:2388 05 19 1B 6F 9E A0 2B 07
.:2390 DE 1A 2D 39 19 23 A0 81
.:2398 A6 3B 05 1F 05 19 23 AC
.:23A0 8A 18 3A 84 1F 05 19 23
.:23A8 F4 19 23 5F 81 A6 2C 02
.:23B0 1F 0A A6 3B 33 E0 2B 19
.:23B8 23 8C 1A 25 23 84 1F 05
.:23C0 19 23 DF 81 A6 2C 02 1F
.:23C8 0A A6 3D 33 E0 2B 19 23
.:23D0 8C 9E A6 3C 33 07 DE 1A
.:23D8 2D 39 EB DE 19 23 5F A6
.:23E0 3E E0 2C A0 3C 1F 06 A6
.:23E8 43 E0 2C 9E A0 2C 33 07
.:23F0 DE 19 1B 6F A0 3C 1F 09
.:23F8 A6 45 33 DE 19 1B 6F 19
.:2400 19 ED 8A 18 3A 81 E0 3C
.:2408 1A 25 23 A0 2B 1F 05 19
.:2410 1B F4 9E A6 60 33 07 E0

```

```

.:2418 2B 84 1F 05 19 24 8A 81
.:2420 A6 2C 05 1F 05 19 1B 78
.:2428 8A 18 3A 1A 25 23 A0 2B
.:2430 1F 05 19 1B F4 B2 E0 3C
.:2438 9E A0 28 07 E0 28 84 1F
.:2440 05 19 24 8A 81 A6 2C 05
.:2448 1F 05 19 1B 78 8A 18 3A
.:2450 1A 25 23 A0 2B 1F 05 19
.:2458 1B F4 B3 E0 3C 9E A0 28
.:2460 07 E0 28 84 1F 05 19 24
.:2468 8A 81 A6 2C 05 1F 05 19
.:2470 1B 78 8A 18 3A 1A 25 23
.:2478 A0 2B B0 02 1F 05 19 1B
.:2480 F4 B4 E0 3C 9E A0 28 07
.:2488 E0 28 A0 2B A0 3C 33 07
.:2490 DE 19 1B 6F 8A 18 3A 1A
.:2498 25 23 A0 2B 1F 05 19 1B
.:24A0 F4 9E A6 61 33 07 DE 84
.:24A8 B0 02 1F 08 8A 18 3A 19
.:24B0 24 A7 19 1B 6F 8A 18 3A
.:24B8 81 A6 23 02 1F 05 19 24
.:24C0 F2 E8 DE 1A 29 61 A6 46
.:24C8 E0 2C 84 1F 06 A6 47 E0
.:24D0 2C 9E A0 29 07 A0 2C 33
.:24D8 07 DE 84 1F 05 19 1B 6F
.:24E0 1A 2D 39 81 A6 2C 05 1F
.:24E8 05 19 1B 78 8A 18 3A 19
.:24F0 24 C1 8A 18 3A 1A 25 23
.:24F8 A0 2B 1F 05 19 1B F4 9E
.:2500 A6 48 33 07 DE 1A 2D 39
.:2508 81 A6 2C 05 1F 05 19 1B
.:2510 78 19 24 EC 8A 18 3A 81
.:2518 A6 A4 05 1F 05 19 1B 78
.:2520 19 1E 6D E8 DE B0 E0 23
.:2528 A0 23 B0 E4 09 A0 23 A0
.:2530 40 E4 0B A0 23 B1 07 E0
.:2538 23 B1 0E E0 40 B0 E0 41
.:2540 1A 26 C2 A0 41 1F 0C 8A
.:2548 18 3A A0 2B E0 40 19 26
.:2550 2E 9E A0 29 07 DE B0 E0
.:2558 42 A0 40 B0 03 1F 0C A0
.:2560 40 A0 2B 05 1F 05 19 34
.:2568 67 81 A6 AA 04 81 A6 B3
.:2570 01 0D 1F 05 19 26 9B 81
.:2578 A6 B1 04 1F 05 19 25 C6
.:2580 B1 E0 43 81 A6 B1 01 1F
.:2588 0E B2 E0 43 81 A6 B3 02
.:2590 1F 05 B4 E0 43 A0 42 A0
.:2598 43 0C 1F 05 19 33 F0 A0
.:25A0 42 A0 43 0D E0 42 8A 18
.:25AB 3A 81 A6 B0 01 81 A6 B4

```

```

.:25B0 04 0C 1F 05 19 25 B0 B4
.:25B8 E0 44 A0 42 B7 02 1F 05
.:25C0 19 33 F0 19 26 15 B1 A6
.:25C8 A3 08 E0 42 A0 2B B0 02
.:25D0 A0 42 B7 02 0D A0 42 BB
.:25D8 01 A0 23 B1 08 A4 09 B4
.:25E0 02 0C 0D 1F 05 19 25 EB
.:25E8 19 34 67 B5 E0 44 A0 42
.:25F0 B8 01 1F 20 B6 E0 44 A0
.:25F8 42 BA 01 1F 17 BB E0 44
.:2600 A0 42 BB 01 1F 0E B2 E0
.:2608 44 A0 42 BD 02 1F 05 B1
.:2610 E0 44 8A 1B 3A A0 2B E0
.:2618 40 A0 23 B1 08 A4 09 E0
.:2620 45 A0 45 A0 44 03 1F 08
.:2628 1A 26 63 19 26 19 A0 23
.:2630 A0 44 E4 09 A0 23 A0 42
.:2638 E4 0A A0 23 B0 E4 0B A0
.:2640 42 B7 02 1F 0B A0 23 A0
.:2648 2B E4 0B A0 42 BC 02 A0
.:2650 42 BD 02 0D 1F 06 B1 0E
.:2658 E0 40 A0 23 B1 07 E0 23
.:2660 19 25 3D A0 23 B1 08 E0
.:2668 23 9E A0 23 A4 0A 33 07
.:2670 DE A0 23 A4 0A BC 04 A0
.:2678 23 A4 0A BD 01 0D 1F 05
.:2680 19 26 94 A0 40 B0 04 1F
.:2688 06 A0 2B E0 40 A0 40 1F
.:2690 05 19 34 67 A0 23 A4 0B
.:2698 E0 40 1D A0 23 B1 0B A4
.:26A0 09 B0 01 1F 0B 1A 26 63
.:26A8 19 26 9B A0 23 B1 0B E0
.:26B0 23 A0 40 B0 03 1F 06 A0
.:26B8 40 E0 2B A0 23 A4 0B E0
.:26C0 40 1D B0 E0 2B B1 A6 2F
.:26C8 01 B1 A6 3A 04 0C B1 A6
.:26D0 2E 02 0D 1F 05 19 2C 0A
.:26D8 B1 A6 40 01 B1 A6 5B 04
.:26E0 0C 1F 05 19 29 5B B1 A6
.:26E8 AA 02 1F 0B 8A 1B 3A 19
.:26F0 26 C2 B1 A6 B3 01 B1 A6
.:26F8 CB 04 0C 1F 05 19 27 B2
.:2700 B1 A6 AB 02 1F 0C B3 E0
.:2708 44 BF E0 42 B1 E0 41 1D
.:2710 B1 A6 A5 02 1F 05 19 2B
.:2718 DB B1 A6 22 02 1F 05 19
.:2720 27 65 B1 A6 AB 02 1F 0C
.:2728 B7 E0 44 BE E0 42 B1 E0
.:2730 41 1D B1 A6 FF 02 1F 0E
.:2738 A6 AA 33 E0 29 B1 E0 34
.:2740 8A 1B 3A 1D B1 A6 2B 05

```

```

.:2748 1F 05 19 33 F0 8A 18 3A
.:2750 1A 25 28 B1 A6 29 05 1F
.:2758 05 19 33 F0 9E E0 29 EB
.:2760 DE 8A 18 3A 1D B0 E0 34
.:2768 EB E0 29 A6 FF E0 2B 8A
.:2770 B3 07 18 3A B1 A6 22 02
.:2778 1F 08 8A 18 3A 19 27 93
.:2780 B1 1F 12 A0 29 B1 33 07
.:2788 E0 29 A0 34 B1 07 E0 34
.:2790 19 27 6F A0 34 B8 04 1F
.:2798 0E A6 EB A0 34 07 33 A0
.:27A0 29 07 E0 29 1D A6 E7 33
.:27AB A0 34 33 07 A0 29 07 E0
.:27B0 29 1D A0 23 B1 A6 94 08
.:27B8 E4 09 A0 23 B1 07 E0 23
.:27C0 B1 A6 C7 01 1F 05 19 28
.:27C8 4A 8A 18 3A B1 A6 2B 05
.:27D0 1F 05 19 33 F0 8A 18 3A
.:27DB 1A 25 28 A0 23 B1 08 E0
.:27E0 23 A0 23 A4 09 E0 34 A0
.:27EB 34 A6 2F 02 A0 34 A6 31
.:27F0 02 0D A0 34 A6 32 02 0D
.:27FB 1F 05 19 28 2B A0 34 A6
.:2800 24 02 A0 34 A6 25 02 0D
.:2808 1F 05 B0 E0 2B A0 2B 1F
.:2810 05 19 34 67 A0 34 A6 30
.:2818 02 A0 34 A6 33 02 0D 1F
.:2820 06 A6 FF E0 2B 19 2B 34
.:2828 A0 2B B0 02 1F 05 19 2B
.:2830 25 B0 E0 2B B1 A6 29 05
.:2838 1F 05 19 33 F0 9E A0 34
.:2840 33 07 E0 29 EB DE 8A 18
.:2848 3A 1D 8A 18 3A B1 A6 2B
.:2850 05 1F 05 19 33 F0 8A 18
.:2858 3A 1A 25 28 A0 2B B0 02
.:2860 1F 05 19 34 67 B1 A6 2C
.:2868 05 1F 05 19 33 F0 8A 18
.:2870 3A 1A 25 28 A0 2B 1F 05
.:2878 19 34 67 A0 23 B1 08 A4
.:2880 09 A6 36 04 1F 05 19 2B
.:2888 B2 B1 A6 29 02 1F 0F 9E
.:2890 A6 A6 33 07 A6 FF 33 07
.:2898 DE 19 28 B2 B1 A6 2C 05
.:28A0 1F 05 19 33 F0 8A 18 3A
.:28AB 1A 25 28 A0 2B 1F 05 19
.:28B0 34 67 B1 A6 29 05 1F 05
.:28B8 19 33 F0 A6 FF E0 2B A0
.:28C0 23 B1 08 E0 23 A0 23 A4
.:28CB 09 E0 34 9E A0 34 33 07
.:28D0 E0 29 EB DE 8A 18 3A 1D
.:28DB B1 E0 2D 8A 18 3A B1 A6

```

```

.:28E0 B0 0D E0 3D B1 E0 3E 1A
.:28E8 29 67 B1 A6 2B 05 A0 3F
.:28F0 B0 01 0D 1F 05 19 33 F0
.:28FB A0 2B 1F 05 19 34 67 A0
.:2900 36 B7 09 B2 07 E0 36 A0
.:2908 36 A7 01 00 0A 21 E0 37
.:2910 A0 36 A0 37 A7 01 00 09
.:2918 08 E0 34 A0 46 A6 3B 33
.:2920 A0 37 33 07 A0 34 33 07
.:2928 E4 0C A0 46 B1 07 E0 46
.:2930 8A 1B 3A 1A 25 2B B1 A6
.:2938 29 05 A0 2B B0 01 0D 1F
.:2940 05 19 33 F0 A0 2B 1F 05
.:2948 19 34 67 A0 46 B1 08 E0
.:2950 46 A0 46 A4 0C E0 29 8A
.:2958 1B 3A 1D B0 E0 3E 19 29
.:2960 64 B1 E0 3E B1 E0 3D E8
.:2968 E0 29 B0 E0 34 B0 E0 47
.:2970 B1 E0 37 B0 E0 2B B0 E0
.:2978 3F B1 A6 41 04 B1 A6 5A
.:2980 01 0D 1F 05 19 33 F0 8A
.:2988 1B 3A B1 A6 40 01 B1 A6
.:2990 5B 04 0C B1 A6 2F 01 B1
.:2998 A6 3A 04 0C 0D 1F 05 19
.:29A0 29 D7 B1 A6 24 02 1F 13
.:29A8 A6 FF E0 2B A0 47 A6 80
.:29B0 0D E0 47 8A 1B 3A 19 29
.:29B8 E6 B1 A6 25 02 1F 17 A6
.:29C0 80 E0 3F A0 3D A6 80 0D
.:29C8 E0 3D A0 47 A6 80 0D E0
.:29D0 47 8A 1B 3A 19 29 E6 A0
.:29D8 37 B1 02 1F 0B B1 E0 47
.:29E0 B2 E0 37 19 29 B7 A0 3D
.:29E8 A7 01 00 09 A0 47 07 E0
.:29F0 37 A0 37 A7 7F FF 01 1F
.:29FB 0D A0 37 AB 91 00 00 00
.:2A00 00 0B E0 37 A0 3D 33 A0
.:2A08 47 33 07 E0 35 A0 2D B0
.:2A10 01 1F 0B B0 E0 2D 19 2A
.:2A18 22 B1 A6 2B 02 1F 05 19
.:2A20 2A EB B0 E0 36 A0 4B B0
.:2A28 02 1F 05 19 2A 44 BD C7
.:2A30 A0 37 CB A0 4B C9 BB 1B
.:2A38 3A B7 B0 03 1F 0B B7 E0
.:2A40 36 19 2A B1 A0 37 A7 54
.:2A48 C9 02 1F 09 A6 AF E0 37
.:2A50 19 2A D5 A0 37 A7 54 49
.:2A58 02 1F 09 A6 AC E0 37 19
.:2A60 2A C9 A0 37 A7 53 54 02
.:2A68 1F 09 A6 AB E0 37 19 2A
.:2A70 C9 A0 4B A0 37 E4 06 A0

```

```

.:2A7B 4B E0 36 A0 4B B1 07 E0
.:2AB0 4B A0 36 A6 20 04 1F 0B
.:2AB8 A0 36 A6 80 07 33 E0 29
.:2A90 1D A0 36 A7 01 00 0A 21
.:2A9B E0 37 A0 36 A0 37 A7 01
.:2AA0 00 09 0B E0 34 A0 37 B2
.:2AAB 01 1F 05 19 2A BB A6 A0
.:2AB0 A0 37 07 33 A0 34 33 07
.:2ABB E0 29 1D A6 A3 33 A0 37
.:2AC0 33 07 A0 34 33 07 E0 29
.:2ACB 1D A0 37 33 E0 29 A0 3E
.:2AD0 1F 05 19 33 F0 A0 37 A6
.:2ADB AF 02 A0 3E B0 01 0C 1F
.:2AE0 06 A6 E6 E0 37 A0 37 33
.:2AEB E0 29 1D A0 46 A0 35 E4
.:2AF0 0C B1 E0 34 8A 1B 3A A0
.:2AFB 46 A0 34 33 A0 2B 33 07
.:2B00 E4 0D A0 46 B1 07 E0 46
.:2B0B 1A 25 2B A0 2B 1F 05 19
.:2B10 34 67 A0 46 B1 0B E0 46
.:2B1B A0 46 A4 0D 32 E0 34 A0
.:2B20 46 A4 0D B2 A6 FF 36 32
.:2B2B E0 2B B1 A6 2C 02 1F 0E
.:2B30 A0 34 B1 07 E0 34 8A 1B
.:2B3B 3A 19 2A F7 B1 A6 29 05
.:2B40 1F 05 19 33 F0 A0 46 A4
.:2B4B 0C E0 35 A0 35 32 A7 01
.:2B50 00 09 A0 35 B2 A6 FF 36
.:2B5B 32 07 E0 37 A0 37 A7 7F
.:2B60 FF 01 1F 0D A0 37 AB 91
.:2B6B 00 00 00 00 0B E0 37 BA
.:2B70 1B 3A B0 E0 36 A0 49 B0
.:2B7B 02 1F 05 19 2B 94 BF C7
.:2B80 A0 37 CB A0 49 C9 BB 1B
.:2B8B 3A B7 B0 03 1F 0B B7 E0
.:2B90 36 19 2B AE A0 49 A0 37
.:2B9B E4 07 A0 49 A0 34 A7 01
.:2BA0 00 0D E4 0B A0 49 E0 36
.:2BAB A0 49 B1 07 E0 49 A0 36
.:2BB0 E0 33 A0 34 A0 36 A4 0B
.:2BBB A6 FF 0C 05 1F 14 A0 36
.:2BC0 A0 34 A0 36 A4 0B A7 01
.:2BCB 00 0C 0D E4 0B 1A 34 30
.:2BD0 A0 36 B3 07 E0 34 A0 36
.:2BDB A7 01 00 04 1F 0C A6 A4
.:2BE0 33 A0 34 33 07 E0 29 1D
.:2BEB A0 34 A7 01 00 0A 21 E0
.:2BF0 37 A0 34 A0 37 A7 01 00
.:2BFB 09 0B E0 34 A6 A5 33 A0
.:2C00 37 33 07 A0 34 33 07 E0
.:2C0B 29 1D 1A 2D 05 A0 39 E0

```

```

.:2C10 29 B1 A6 2E 02 1F 0F 1A
.:2C18 2D 02 A0 29 E9 2E 07 A0
.:2C20 39 07 E0 29 B1 A6 45 05
.:2C28 1F 05 19 2C 52 1A 2D 02
.:2C30 A0 29 E9 45 07 A0 39 07
.:2C38 E0 29 A0 39 31 A6 27 04
.:2C40 1F 05 19 2C 52 1A 34 4E
.:2C48 A0 29 A0 29 2F B1 08 34
.:2C50 E0 29 A0 29 31 E0 4A A0
.:2C58 4A 21 E0 4B A0 4A A7 7F
.:2C60 FF 01 A0 4A A7 7F FF 0E
.:2C68 04 0D A0 4B A0 4A 05 0D
.:2C70 1F 05 19 2C D5 A0 4A B0
.:2C78 04 A0 4A FF 01 0D 1F 05
.:2C80 19 2C 9A A6 B0 E0 4B A0
.:2C88 4A BF 01 1F 06 A6 E0 E0
.:2C90 4B A0 4B A0 4A 07 33 E0
.:2C98 29 1D A0 4A B0 04 A0 4A
.:2CA0 A6 FF 01 0D 1F 05 19 2C
.:2CAB B3 A6 A6 33 A0 4A 33 07
.:2CB0 E0 29 1D A0 4A A7 01 00
.:2CBB 0A 21 E0 4B A0 4A A0 4B
.:2CC0 A7 01 00 09 08 E0 34 A6
.:2CC8 A7 33 A0 4B 33 07 A0 34
.:2CD0 33 07 E0 29 1D A0 4A C0
.:2CDB A6 AB 33 E0 29 A6 2D 2E
.:2CE0 A6 2E 2E A7 01 00 09 07
.:2CE8 B2 07 E0 4B A0 4B E0 34
.:2CF0 A0 4B B4 07 A0 34 11 A0
.:2CF8 29 A0 34 2E 33 07 E0 29
.:2D00 13 1D 8A 18 3A E8 E0 39
.:2D08 B1 A6 AA 05 B1 A6 AB 05
.:2D10 0C 1F 05 19 2D 23 B1 A6
.:2D18 AB 02 1F 06 E9 2D E0 39
.:2D20 8A 18 3A B1 A6 30 04 B1
.:2D28 A6 39 01 0D 1F 03 1D A0
.:2D30 39 B1 33 07 E0 39 19 2D
.:2D38 20 9C 9E 2F 07 DC B4 42
.:2D40 9E 3C 45 1D 1D B0 E0 4C
.:2D48 8C A7 63 D5 07 CD A6 33
.:2D50 8D B2 07 17 A6 32 8D 17
.:2D58 B1 E0 4D F0 A0 4D 11 A6
.:2D60 31 8D 17 A6 30 8D 17 8D
.:2D68 2E A6 B0 04 8D 2E B8 0C
.:2D70 B0 01 0D 1F 05 19 2D A9
.:2D78 A6 31 8D 17 A6 30 8D 17
.:2D80 8D 2E A6 7F 01 8D 2E B8
.:2D88 0C B0 01 A0 4D F0 04 0C
.:2D90 0D 1F 05 19 2D A9 13 8D
.:2D98 2E A6 7C 02 1F 0D B0 8D
.:2DA0 17 B0 8D B2 07 17 19 2D

```



```

.:2DAB C4 A0 4C B1 07 E0 4C A0
.:2DB0 4C B5 04 1F 05 19 2D 4E
.:2DB8 B0 8D 17 B0 8D BC 08 17
.:2DC0 15 19 18 0A B0 CD 59 1D
.:2DC8 A0 20 9C E4 04 A6 4F 33
.:2DD0 DE AB 1F 05 19 2D EE 1A
.:2DD8 2D 39 B2 48 9E 47 9E EB
.:2DE0 02 1F 05 B0 33 DE AB B0
.:2DE8 02 1F 05 19 2D D7 1A 2D
.:2DF0 39 A0 2F 1F 04 B5 61 B2
.:2DF8 61 B3 61 B4 61 A0 3A A7
.:2E00 7F FF 0E E4 03 1A 2D 44
.:2E08 90 B1 01 1F 15 B6 61 BF
.:2E10 42 E9 53 8E 07 EB 3A 5A
.:2E18 2F 07 94 07 43 19 2E 41
.:2E20 B6 61 8E EB 3A 5A 2F 07
.:2E28 94 07 EC 2C 50 2C 57 07
.:2E30 B6 B8 B6 60 04 1A 33 D6
.:2E38 B6 42 B1 33 3C B8 33 3C
.:2E40 45 9B 1F 0B 1A 34 E3 19
.:2E48 33 0D EB DE B1 E0 33 A0
.:2E50 49 B0 02 1F 05 19 2F BE
.:2E58 B8 E0 33 A0 49 B1 0B C0
.:2E60 80 F0 04 1F 0B A6 B0 B0
.:2E68 07 33 DE 19 2E B8 B0 A6
.:2E70 20 04 1F 0B A6 E0 B0 07
.:2E78 33 DE 19 2E B8 A0 33 B1
.:2E80 07 E0 33 B0 A7 01 00 04
.:2E88 1F 0C A6 A6 33 B0 33 07
.:2E90 DE 19 2E B8 A0 33 B1 07
.:2E98 E0 33 B0 A7 01 00 0A 21
.:2EA0 E0 4B B0 A0 4B A7 01 00
.:2EAB 09 0B E0 4A A6 A7 33 A0
.:2EB0 4B 33 07 A0 4A 33 07 DE
.:2EB8 9E F0 33 07 B1 33 07 A6
.:2EC0 DA 33 07 A6 AA 33 07 B0
.:2EC8 33 07 B0 33 07 DE B0 E0
.:2ED0 37 B0 A0 37 11 A0 37 A4
.:2ED8 0B A7 01 00 04 1F 05 19
.:2EE0 2F BD A0 37 A4 0B A6 FF
.:2EE8 0C E0 34 A0 34 B1 02 1F
.:2EF0 0B A0 33 B7 07 E0 33 19
.:2EF8 2F BD A0 37 A4 07 E0 3B
.:2F00 A0 3B B0 04 1F 0D A0 3B
.:2F08 AB 91 00 00 00 00 07 E0
.:2F10 3B A0 3B A7 01 00 0A E0
.:2F18 2B A0 3B A0 2B A7 01 00
.:2F20 09 0B E0 3F EB E0 2B A0
.:2F28 3F A6 7F 01 1F 11 E9 24
.:2F30 E0 2B A0 2B A6 7F 01 1F
.:2F38 06 E9 25 E0 2B A0 3F A6

```

```

.:2F40 7F 0C 33 E0 29 A0 3F A6
.:2F48 7F B0 02 0C 1F 05 EB E0
.:2F50 29 A0 2B A6 7F 0C 33 A0
.:2F58 29 07 A0 28 07 E0 2B 9B
.:2F60 B1 07 D8 BF B5 97 07 34
.:2F68 3C 99 40 E7 19 3F 42 41
.:2F70 44 20 53 55 42 53 43 52
.:2F78 49 50 54 20 45 52 52 4F
.:2F80 52 20 20 49 4E 20 3C A0
.:2F88 28 3E 1A 34 BA 13 BF 42
.:2F90 E9 53 BE 07 E9 3A 07 92
.:2F98 07 43 BE E9 3A 07 92 07
.:2FA0 EC 2C 50 2C 57 07 B3 BB
.:2FAB B3 60 04 1A 33 D6 90 B3
.:2FB0 04 9A B0 02 0D 1F 05 19
.:2FBB 2F C9 A0 4E B1 07 E0 37
.:2FC0 1A 33 B6 A0 37 C1 19 2F
.:2FC8 E2 B3 42 B1 33 3C BB 33
.:2FD0 3C 45 A7 0B 01 C1 A0 4E
.:2FDB B1 11 B3 42 B1 2E 33 3C
.:2FE0 45 13 B1 BD 07 A0 33 07
.:2FEB A0 2F 07 9C 07 E0 37 90
.:2FF0 B4 02 1F 0D B1 BD 07 A0
.:2FFB 33 07 A0 4F 07 E0 37 1A
.:3000 33 B6 A0 37 A0 4B B7 09
.:300B 07 E0 37 1A 33 B6 1A 33
.:3010 B6 B1 BC 07 E0 37 1A 33
.:301B B6 A0 37 A0 33 07 E0 37
.:3020 1A 33 B6 A0 37 A0 2F 07
.:302B B1 07 E0 37 1A 33 B6 A0
.:3030 37 C1 B3 42 9E 3C 45 A0
.:303B 33 B9 04 1F 05 19 30 CD
.:3040 B0 E0 37 A0 49 B1 0B A0
.:304B 37 11 A0 37 A4 0B A7 01
.:3050 00 04 1F 05 19 30 CC A0
.:305B 37 A4 0B A6 FF 0C E0 34
.:3060 A0 34 B1 01 1F 05 19 30
.:306B CC A0 37 A4 07 E0 3B A0
.:3070 3B B0 04 1F 0D A0 3B AB
.:307B 91 00 00 00 00 07 E0 3B
.:3080 A0 3B A7 01 00 0A E0 2B
.:308B A0 3B A0 2B A7 01 00 09
.:3090 0B E0 3F A0 37 B2 09 B7
.:309B 07 E0 3C A0 3C A7 01 00
.:30A0 0A 21 E0 4A A0 3C A0 4A
.:30AB A7 01 00 09 0B E0 3C B3
.:30B0 42 A6 BA 33 3C F0 33 3C
.:30BB B1 33 3C A0 2B 33 3C A0
.:30C0 3F 33 3C A0 4A 33 3C A0
.:30CB 3C 33 3C 45 13 B3 42 F6
.:30D0 33 3C 45 A0 2F B0 02 1F

```

```

.:30DB 05 19 31 09 BE EB 3A 44
.:30E0 2F 07 94 07 B2 BB B2 60
.:30EB 04 B1 E0 50 A0 2F A0 50
.:30F0 11 B2 4B A0 51 47 A0 51
.:30FB EB 02 1F 06 B0 33 E0 51
.:3100 B3 42 A0 51 3C 45 13 B2
.:310B 61 B3 42 A6 FF 33 3C 45
.:3110 B0 E0 2C B0 E0 34 BE EB
.:311B 3A 50 2F 07 94 07 EC 2C
.:3120 53 2C 52 07 B2 BB B2 60
.:312B 04 E7 09 13 11 50 41 53
.:3130 53 32 20 20 3E BB A6 6F
.:313B 07 E0 52 A0 52 BF 07 E0
.:3140 53 B0 E0 54 9C B1 0B A0
.:314B 54 11 A0 52 1B 3A A0 54
.:3150 A0 2C A4 04 04 A0 2C A0
.:315B 20 02 0D 1F 05 19 31 CB
.:3160 A0 2C A0 20 04 1F 22 EB
.:316B 13 11 11 3C A0 2C A4 05
.:3170 A7 7F FF 07 3E A0 2C B1
.:317B 07 E0 2C A0 54 A0 2C A4
.:3180 04 03 1F 05 19 31 60 90
.:318B B1 01 1F 05 19 31 CB A0
.:3190 54 B1 07 A7 01 00 0A E0
.:319B 3F A0 54 B1 07 A0 3F A7
.:31A0 01 00 09 0B E0 2B B6 42
.:31AB B4 33 3C B4 33 3C A0 2B
.:31B0 33 3C A0 3F 33 3C E9 3D
.:31BB 3C A0 2C B1 0B A4 05 A7
.:31C0 7F FF 07 3C B0 33 3C 45
.:31CB A0 54 A0 34 A4 03 05 1F
.:31D0 05 19 32 1F A0 34 B1 07
.:31DB E0 34 B2 A6 FF 02 1F 0F
.:31E0 A0 2C A4 04 A0 54 0B B1
.:31EB 07 C2 19 32 1F B2 E0 2B
.:31F0 A0 52 1B 3A A0 2B A7 01
.:31FB 00 09 B2 07 A7 7F FF 0B
.:3200 E0 2B 1A 33 99 A0 2B A7
.:320B 01 00 0A C2 A0 53 1B 3A
.:3210 A0 54 B1 07 E0 54 A0 2B
.:321B B2 A7 01 00 09 0B C2 A0
.:3220 53 1B 3A 13 B2 61 90 B4
.:322B 02 9A B0 01 0C 1F 05 19
.:3230 32 BB 90 B4 02 1F 22 A0
.:323B 2F 9C 07 E0 37 A0 37 A0
.:3240 4F 04 1F 15 A0 37 E0 37
.:324B A0 4F B1 0B A0 37 11 B3
.:3250 42 A6 4F 33 3C 45 13 90
.:325B B1 02 1F 0D B6 42 B0 33
.:3260 3C B0 33 3C 45 B6 61 B0
.:326B 33 B0 33 07 B0 33 07 B0

```

```

.:3270 33 07 B0 33 07 DE B0 E0
.:3278 37 A0 4B B1 0B A0 37 11
.:3280 A0 37 A4 06 E0 3B A0 3B
.:3288 B0 04 1F 0D A0 3B AB 91
.:3290 00 00 00 00 07 E0 3B A0
.:3298 3B A7 01 00 0A E0 2B A0
.:32A0 3B A0 2B A7 01 00 09 0B
.:32AB E0 3F B3 42 A0 2B 33 3C
.:32B0 A0 3F 33 3C 9E 3C 45 13
.:32B8 B3 61 8F B5 97 07 34 3C
.:32C0 99 40 EF 45 52 52 4F 52
.:32C8 53 3A 3C 96 9B 07 3C E7
.:32D0 10 20 20 20 20 20 45 5B
.:32D8 54 45 4E 53 49 4F 4E 53
.:32E0 3A 3C 9D 3E BF 42 E9 53
.:32E8 8E 07 EB 3A 50 2F 07 94
.:32F0 07 43 BF 42 E9 53 8E 07
.:32F8 EB 3A 44 2F 07 94 07 43
.:3300 90 B1 02 A0 55 B0 01 0D
.:3308 1F 05 19 39 9C 96 9B 07
.:3310 B0 02 1F 05 19 33 50 B7
.:3318 33 3C B7 33 3C B7 33 3C
.:3320 E7 19 11 11 C6 4F 52 54
.:3328 53 45 54 5A 45 4E 3A 20
.:3330 12 4A 92 41 2F 12 4E 92
.:3338 45 49 4E 3E 95 47 95 E9
.:3340 4E 02 1F 05 19 39 9C 95
.:3348 E9 4A 05 1F 05 19 33 3C
.:3350 A0 56 B1 07 E0 56 A0 56
.:3358 B8 02 1F 10 A0 57 D1 B0
.:3360 E0 56 91 B0 02 1F 05 19
.:3368 34 D3 1A 3B B3 91 B0 02
.:3370 1F 05 19 34 D3 9A B1 07
.:3378 DA B7 33 3C B0 E0 2F B0
.:3380 E0 20 B0 E0 3A B0 E0 21
.:3388 B0 E0 3B 90 B4 04 1F 0B
.:3390 B0 E0 49 B0 E0 4B 19 1B
.:3398 A2 BB C7 A0 2B CB A0 20
.:33A0 C9 8B 1B 3A B7 B0 04 1F
.:33AB 07 B0 C7 1A 34 03 B7 A4
.:33B0 04 B1 07 E0 2B 1D A0 37
.:33B8 A7 01 00 0A 21 E0 3B A0
.:33C0 37 A0 3B A7 01 00 09 0B
.:33C8 E0 2B B3 42 A0 2B 33 3C
.:33D0 A0 3B 33 3C 45 1D BF 4B
.:33D8 A0 5B 55 A0 59 57 A0 5A
.:33E0 57 A0 5A 51 A0 5B B0 02
.:33E8 1F 03 1D BF D7 19 39 A7
.:33F0 E7 0C 53 59 4E 54 41 5B
.:33F8 20 45 52 52 4F 52 E0 26
.:3400 19 34 B1 E7 17 55 4E 44

```

```

.:340B 45 46 27 44 20 53 54 41
.:3410 54 45 4D 45 4E 54 20 45
.:341B 52 52 4F 52 E0 26 A0 2C
.:3420 B1 08 A4 05 A7 7F FF 07
.:342B DF 9B B1 07 DB 19 34 A5
.:3430 E7 13 42 41 44 20 53 55
.:343B 42 53 43 52 49 50 54 20
.:3440 45 52 52 4F 52 E0 26 9B
.:344B B1 07 DB 19 34 A5 E7 0E
.:3450 4F 56 45 52 46 4C 4F 57
.:345B 20 45 52 52 4F 52 E0 26
.:3460 9B B1 07 DB 19 34 A5 E7
.:346B 13 54 59 50 45 20 4D 49
.:3470 53 4D 41 54 43 4B 20 45
.:347B 52 52 4F 52 E0 26 19 34
.:3480 B1 A0 25 1F 03 1D 96 B1
.:348B 07 D6 B1 E0 25 1A 34 A5
.:3490 8A 1B 3A 84 B0 02 1F 05
.:349B 19 34 90 1D 9D B1 07 DD
.:34A0 A0 3B 1F 03 1D 8F B5 97
.:34AB 07 34 3C 99 40 E9 3F 3C
.:34B0 A0 26 3C EB 20 49 4E 3C
.:34BB 9F 3E 97 B1 07 D7 97 F2
.:34C0 02 1F 11 99 A6 2B 07 D9
.:34CB B0 D7 99 A6 2B 01 1F 04
.:34D0 B0 D9 1D 9B B0 02 1F 05
.:34DB 19 39 9C B0 DB 1A 3B AC
.:34E0 19 33 79 BF 42 E9 53 8E
.:34EB 07 EB 3A 50 2F 07 94 07
.:34F0 43 BF 42 E9 53 8E 07 EB
.:34FB 3A 44 2F 07 94 07 43 9C
.:3500 A0 2F 07 E0 37 A0 4F A0
.:350B 37 04 1F 06 A0 37 E0 4F
.:3510 1D B1 D0 19 36 B6 EB 13
.:351B 11 11 3C BD 40 E7 0D D0
.:3520 52 4F 47 52 41 4D 4D 57
.:352B 41 4B 4C 3A 3E E7 21 11
.:3530 11 31 2E 20 3D 20 20 C5
.:353B 49 4E 20 D0 52 4F 47 52
.:3540 41 4D 4D 20 4E 41 43 4B
.:354B 20 C4 52 49 56 45 20 31
.:3550 3E E7 26 11 11 32 2E 20
.:355B 3D 20 20 C1 4C 4C 45 20
.:3560 D0 52 4F 47 52 41 4D 4D
.:356B 45 20 56 4F 4E 20 C4 52
.:3570 49 56 45 20 30 20 41 55
.:357B 46 3E E7 23 20 20 20 20
.:3580 20 20 4C 45 45 52 45 20
.:358B C4 49 53 4B 45 54 54 45
.:3590 20 49 4E 20 C4 52 49 56
.:359B 45 20 31 20 4D 49 54 3E

```

```

.:35A0 E7 1C 20 20 20 20 20 20
.:35A8 C3 4F 4D 50 2D CD 41 53
.:35B0 43 48 49 4E 45 4E 50 52
.:35B8 4F 47 52 41 4D 4D 3E E7
.:35C0 26 11 11 33 2E 20 3D 20
.:35C8 20 C1 4C 4C 45 20 D0 52
.:35D0 4F 47 52 41 4D 4D 45 20
.:35D8 56 4F 4E 20 C4 52 49 56
.:35E0 45 20 30 20 41 55 46 3E
.:35E8 E7 24 20 20 20 20 20 20
.:35F0 4C 45 45 52 45 20 C4 49
.:35F8 53 4B 45 54 54 45 20 49
.:3600 4E 20 C4 52 49 56 45 20
.:3608 31 2C 20 4E 55 52 3E E7
.:3610 24 20 20 20 20 20 20 45
.:3618 52 53 54 45 53 20 D0 52
.:3620 4F 47 52 41 4D 4D 20 4D
.:3628 49 54 20 C3 4F 4D 50 2D
.:3630 D0 52 47 52 2E 3E E7 26
.:3638 11 11 34 2E 20 3D 20 20
.:3640 C1 4C 4C 45 20 D0 52 4F
.:3648 47 52 41 4D 4D 45 20 56
.:3650 4F 4E 20 C4 52 49 56 45
.:3658 20 30 20 41 55 46 3E E7
.:3660 23 20 20 20 20 20 20 4C
.:3668 45 45 52 45 20 C4 49 53
.:3670 4B 45 54 54 45 20 49 4E
.:3678 20 C4 52 49 56 45 20 31
.:3680 20 4D 49 54 3E E7 0D 20
.:3688 20 20 20 20 20 CF D6 C5
.:3690 D2 CC C1 D9 3E 8F 3C E7
.:3698 0E C2 49 54 54 45 20 57
.:36A0 41 45 48 4C 45 4E 21 3E
.:36A8 90 47 90 B1 04 90 B4 01
.:36B0 0D 1F 05 19 36 AB E7 0D
.:36B8 28 D0 52 4F 47 52 41 4D
.:36C0 4D 41 52 54 3A 90 30 07
.:36C8 E9 29 07 E0 5B 90 B4 02
.:36D0 1F 04 B1 DB A7 17 83 E0
.:36D8 4E 90 B1 01 1F 08 1A 36
.:36E0 E4 19 37 01 E9 93 3C BD
.:36E8 40 E7 0D C1 55 53 54 52
.:36F0 4F 20 2D 20 C3 4F 4D 50
.:36F8 3E BC 40 A0 5B 3E 19 2D
.:3700 44 E7 3E 11 11 C9 4E 20
.:3708 C4 52 49 56 45 20 30 20
.:3710 20 C4 49 53 4B 45 54 54
.:3718 45 20 4D 49 54 20 D0 52
.:3720 4F 47 52 41 4D 4D 45 4E
.:3728 20 5A 55 4D 20 43 4F 4D
.:3730 50 49 4C 49 45 52 45 4E

```

```

.:3738 20 45 49 4E 4C 45 47 45
.:3740 4E 3E E7 1A 11 C4 52 55
.:3748 45 43 4B 45 20 20 12 4A
.:3750 92 20 20 57 45 4E 4E 20
.:3758 46 45 52 54 49 47 3E 93
.:3760 47 93 E9 4A 05 1F 05 19
.:3768 37 5F BF 42 EA 49 30 43
.:3770 1A 33 D6 1A 39 CD E7 26
.:3778 11 11 C9 4E 20 C4 52 49
.:3780 56 45 20 31 20 20 4C 45
.:3788 45 52 45 20 C4 49 53 4B
.:3790 45 54 54 45 20 45 49 4E
.:3798 4C 45 47 45 4E 21 3E E7
.:37A0 2C 11 D3 4F 4C 4C 20 C4
.:37AB 49 53 4B 45 54 54 45 20
.:37B0 46 4F 52 4D 41 54 49 45
.:37B8 52 54 20 57 45 52 44 45
.:37C0 4E 20 12 4A 92 41 2F 12
.:37C8 4E 92 45 49 4E 3E 93 47
.:37D0 EE 30 30 30 30 30 30 E6
.:37D8 93 E9 4E 02 1F 05 19 38
.:37E0 A3 93 E9 4A 05 1F 05 19
.:37E8 37 CE E7 1A 11 C4 49 53
.:37F0 4B 45 54 54 45 20 57 49
.:37F8 52 44 20 46 4F 52 4D 41
.:3800 54 49 45 52 54 21 3E B6
.:3808 C0 A0 5C F2 02 1F 05 A6
.:3810 90 C0 EA 23 39 B9 B8 B9
.:3818 60 04 1A 33 D6 BF 42 EB
.:3820 55 31 3A 3C B9 3C B0 3C
.:3828 A0 5C 3C B0 43 B9 61 EB
.:3830 E0 51 80 E0 58 80 BF 07
.:3838 A0 58 11 BF 42 EB 4D 2D
.:3840 52 3C A0 58 33 3C A6 32
.:3848 33 43 BF 48 93 47 93 EB
.:3850 02 1F 06 A6 A0 33 D3 93
.:3858 32 A6 A0 04 1F 09 A0 51
.:3860 93 07 E0 51 13 BF 42 EB
.:3868 4D 2D 52 3C B0 F2 07 33
.:3870 3C A6 32 33 43 BF 48 95
.:3878 47 BF 42 EB 4D 2D 52 3C
.:3880 80 F3 07 33 3C A6 32 33
.:3888 43 BF 48 93 47 95 93 07
.:3890 D5 BF 42 EB 4E 31 3A 3C
.:3898 A0 51 3C E9 2C 3C 95 43
.:38A0 1A 33 D6 BF 42 EA 49 31
.:38AB 43 1A 33 D6 B0 DA B1 D1
.:38B0 B0 E0 56 EA 23 39 B9 B8
.:38B8 B9 60 04 BF 42 EB 55 31
.:38C0 3A 3C B9 3C B0 3C A0 5C
.:38C8 3C 91 43 B9 61 BF 42 EB

```

```

.:38D0 4D 2D 52 3C B0 33 3C A6
.:38D8 32 33 43 BF 48 A0 51 47
.:38E0 A0 51 EB 02 1F 08 B0 E0
.:38EB 57 19 39 0A BF 42 EB 4D
.:38F0 2D 52 3C B1 33 3C A6 32
.:38F8 33 43 BF 48 A0 51 47 A0
.:3900 51 EB 05 1F 07 A0 51 32
.:3908 E0 57 BF 42 EB 4D 2D 52
.:3910 3C A0 56 A6 20 09 B2 07
.:3918 33 3C A6 32 33 43 BF 48
.:3920 A0 51 47 A0 51 EB 02 1F
.:3928 06 B0 33 E0 51 A0 51 32
.:3930 A6 B2 05 1F 05 19 39 7D
.:3938 EB D3 A0 56 A6 20 09 B5
.:3940 07 E0 58 A0 56 A6 20 09
.:3948 F4 07 A0 58 11 BF 42 EB
.:3950 4D 2D 52 3C A0 58 33 3C
.:3958 A6 32 33 43 BF 48 A0 51
.:3960 47 A0 51 EB 02 1F 07 A6
.:3968 A0 33 E0 51 A0 51 32 A6
.:3970 A0 05 1F 08 93 A0 51 07
.:3978 D3 13 19 36 E4 A0 56 B1
.:3980 07 E0 56 A0 56 B8 04 1F
.:3988 05 19 39 0A A0 57 D1 B0
.:3990 E0 56 A0 57 B0 02 1F 03
.:3998 1D 19 38 B3 B0 E0 58 EA
.:39A0 4F 4B E0 59 E9 13 3E BF
.:39A8 B6 97 07 34 3C 99 40 E9
.:39B0 12 3C A0 58 3C A0 59 3E
.:39B8 B2 61 B3 61 B4 61 BF 61
.:39C0 B0 E0 58 B5 A0 58 11 B7
.:39C8 33 3C 13 4F 4F F2 E0 5C
.:39D0 EA 23 39 B9 B8 B9 60 04
.:39D8 1A 33 D6 BF 42 EB 55 31
.:39E0 3A 3C B9 3C B0 3C F2 3C
.:39EB B0 43 B9 61 BF 42 EB 4D
.:39F0 2D 52 3C B1 33 3C A6 32
.:39F8 33 43 BF 48 93 47 93 B1
.:3A00 33 05 1F 05 19 3A 26 BF
.:3A08 42 EB 4D 2D 52 3C B2 33
.:3A10 3C A6 32 33 43 BF 48 93
.:3A18 47 93 B1 33 02 93 A6 41
.:3A20 33 02 0D 1F 03 1D A6 27
.:3A28 E0 5C 1D 4F A9 00 2C A9
.:3A30 B0 B5 24 A0 2D B1 2D B5
.:3A38 22 C8 B1 2D B5 23 A9 00
.:3A40 A0 17 91 2D C8 91 2D A0
.:3A48 1E 91 2D C8 91 2D A0 11
.:3A50 B1 2D 18 69 01 91 2D A0
.:3A58 26 D1 2D 90 11 A9 00 A0
.:3A60 0A 91 2D A9 01 A0 18 91

```



```

.:3A6B 2D A0 1F 91 2D 60 AB B1
.:3A70 22 24 24 30 04 C9 20 F0
.:3A7B D5 A0 0A 91 2D C9 3A F0
.:3AB0 E8 60 A9 00 A0 25 91 2D
.:3AB8 C8 91 2D A0 2D B1 2D 85
.:3A90 22 C8 B1 2D 85 23 A2 02
.:3A9B 20 C6 FF A0 00 84 90 20
.:3AA0 E4 FF 91 22 D0 08 C0 01
.:3AAB F0 0C C0 04 B0 03 C8 D0
.:3AB0 EE 98 A0 26 91 2D 4C CC
.:3AB8 FF A9 FF A0 33 91 2D C8
.:3AC0 B1 2D AB B1 2F 18 65 2F
.:3AC8 85 5F C8 B1 2F 65 30 85
.:3AD0 60 A0 41 B1 2D 85 25 C8
.:3ADB B1 2D 85 24 A0 3A B1 2D
.:3AE0 85 26 C8 B1 2D 85 27 A0
.:3AEB 00 84 22 84 23 A9 07 2C
.:3AF0 A9 02 18 65 5F 85 5F 90
.:3AF8 02 E6 60 A5 26 D1 5F F0
.:3B00 13 E6 22 D0 02 E6 23 A5
.:3B08 23 C5 25 D0 E3 A5 22 C5
.:3B10 24 D0 DD 60 C8 B1 5F 88
.:3B18 C5 27 D0 E5 A0 33 A5 23
.:3B20 91 2D C8 A5 22 91 2D 60
.:3B2B A2 02 20 C6 FF 20 E4 FF
.:3B30 A0 11 91 2D 4C CC FF A2
.:3B38 03 20 C9 FF A0 11 B1 2D
.:3B40 20 D2 FF 4C CC FF 5A 00
.:3B48 00 00 00 00 00 C3 80 00
.:3B50 00 00 00 00 00 C3 B1 00 00
.:3B58 00 00 00 00 C3 B2 00 00 00
.:3B60 00 00 C3 B3 00 00 00 00
.:3B68 00 C3 B4 00 00 00 00 00
.:3B70 5A 80 00 00 00 00 00 C3
.:3B78 B5 00 00 00 00 00 C3 B6
.:3B80 00 00 00 00 00 C3 B7 00
.:3B88 00 00 00 00 00 CD 80 00 00
.:3B90 00 00 00 CE 80 00 00 00
.:3B98 00 00 D1 80 00 00 00 00
.:3BA0 00 51 00 00 00 00 00 00
.:3BAB 59 80 00 00 00 00 00 58
.:3BB0 80 00 00 00 00 00 D4 80
.:3BB8 00 00 00 00 00 D4 B3 00
.:3BC0 00 00 00 00 41 B2 00 00
.:3BC8 00 00 00 41 80 00 00 00
.:3BD0 00 00 41 B1 00 00 00 00
.:3BDB 00 42 B1 00 00 00 00 00
.:3BE0 58 31 00 00 00 00 00 58
.:3BEB 32 00 00 00 00 00 58 33
.:3BF0 00 00 00 00 00 58 34 00
.:3BF8 00 00 00 00 D4 B2 00 00

```

```

.:3C00 00 00 00 D4 BB 00 00 00
.:3C08 00 00 C5 80 00 00 00 00
.:3C10 00 5B 30 00 00 00 00 00
.:3C18 49 80 00 00 00 00 00 42
.:3C20 31 00 00 00 00 00 C2 B5
.:3C28 00 00 00 00 00 C2 B3 00
.:3C30 00 00 00 00 42 34 00 00
.:3C38 00 00 00 49 00 00 00 00
.:3C40 00 00 43 32 00 00 00 00
.:3C48 00 5B 35 00 00 00 00 00
.:3C50 5B B1 00 00 00 00 CB
.:3C58 B2 00 00 00 00 00 45 80
.:3C60 00 00 00 00 00 4A 80 00
.:3C68 00 00 00 00 C5 B2 00 00
.:3C70 00 00 00 CC B2 00 00 00
.:3C78 00 00 CA 80 00 00 00 00
.:3C80 00 46 41 00 00 00 00 00
.:3C88 45 B1 00 00 00 00 00 CB
.:3C90 80 00 00 00 00 00 CB B1
.:3C98 00 00 00 00 00 4B 80 00
.:3CA0 00 00 00 00 CB B3 00 00
.:3CAB 00 00 00 4B 31 00 00 00
.:3CB0 00 00 CA B1 00 00 00 00
.:3CBB 00 46 80 00 00 00 00 00
.:3CC0 47 31 00 00 00 00 00 46
.:3CCB 00 00 00 00 00 00 46 31
.:3CD0 00 00 00 00 00 4A B1 00
.:3CDB 00 00 00 00 C2 B2 00 00
.:3CE0 00 00 00 D9 80 00 00 00
.:3CEB 00 00 46 35 00 00 00 00
.:3CF0 00 C6 B1 00 00 00 00 00
.:3CFB 46 34 00 00 00 00 00 CC
.:3D00 B3 00 00 00 00 00 46 5B
.:3D0B 00 00 00 00 00 46 3B 00
.:3D10 00 00 00 00 49 31 00 00
.:3D1B 00 00 00 49 32 00 00 00
.:3D20 00 00 49 30 00 00 00 00
.:3D2B 00 49 33 00 00 00 00 00
.:3D30 49 34 00 00 00 00 00 C6
.:3D3B B2 00 00 00 00 00 47 00
.:3D40 00 00 00 00 00 4B 00 00
.:3D48 00 00 00 00 46 36 00 00
.:3D50 00 00 00 46 37 00 00 00
.:3D5B 00 00 D1 B2 00 00 00 00
.:3D60 00 D1 B1 00 00 00 00 00
.:3D6B D4 B1 00 00 00 00 00 D4
.:3D70 B7 00 00 00 00 00 43 00
.:3D7B 00 00 00 00 00 42 80 00
.:3D80 00 00 00 00 CE B1 00 00
.:3D8B 00 00 00 CE B2 00 00 00
.:3D90 00 00 CC 80 00 00 00 00

```

```

.:3D98 00 5B 36 00 00 00 00 00
.:3DA0 D4 B4 00 00 00 00 00 D4
.:3DAB B5 00 00 00 00 00 41 00
.:3DB0 00 00 00 00 00 41 B5 00
.:3DBB 00 00 00 00 41 B6 00 00
.:3DC0 00 00 00 54 B0 00 00 00
.:3DCB 00 00 CF B0 00 00 00 00
.:3DD0 00 FF 45 73 0D CD FF 3A
.:3DD8 1F FC 00 00 B1 AD 00 00
.:3DE0 29 00 00 00 00 00 00 00
.:3DEB 00 00 00 00 00 00 00 00
.:3DF0 00 00 00 00 00 00 00 00
.:3DFB 00 00 00 00 00 00 00 00
.:3E00 B4 FF FF FF FF FF FF FF
.:3E08 FF FF FF FF 7F FF FF FF
.:3E10 FF FF FF FF FF FF FF FF
.:3E18 FF FF FF FF FF FF FF FF
.:3E20 00 00 00 00 00 00 00 00
.
```

```

B27D B6 71          STX $71
.
```

```

.?
.?
.?
.
```

As you can see, this has been presented as a memory dump. A Basic loader would just take up too much space, and a disassembly would be meaningless since it is itself a compiled program. As we mentioned earlier, compilers produce pseudo-code, not true machine code.

So, you'll need a monitor of some kind when it comes round to typing the thing in. This monitor (or assembler) must not occupy any memory from locations \$0800 to \$3E20: this is the area that the compiler uses. Since there are 13,856 numbers to type in, this is going to take some time, and I suggest that you type in 1000 bytes per day, or evening, and continually re-save your progress as you go along. That way, at the end of a fortnight, you'll have a working compiler!

To assist you in your typing, the following checksum program checks every 100 bytes for any errors. At the end of the first day, run this short program (in direct mode: don't enter it as a program listing), and if any of the values returned disagree, then you know roughly where the error lies.

```
A = 0: B = 0: FOR I = 0 TO 9: FOR J = 0 TO 99: A = A + PEEK(2048 + I * 100 + J): NEXT J: ?A; I: B = B + A: A = 0: NEXT I: ?B
```

If you've correctly typed in all 1000 bytes, the following series of values should appear on the screen:

```
7043 0
7515 1
9564 2
11161 3
12128 4
11096 5
9479 6
11830 7
10742 8
12367 9
102925
```

If, for example, the first number reported is 7033 instead of 7043, then you'll know that somewhere in the first 100 bytes you've made an error (or two!) Thus we can selectively check blocks of 100 bytes to try to eliminate any errors that might have occurred. Once everything is correct, save the first day's progress. Then, on day two, type in the next 1000 bytes of data. The checksum program for this second 1000 bytes now changes to become:

```
A = 0: B = 0: FOR I = 0 TO 9: FOR J = 0 TO 99: A = A + PEEK(3048 + I * 100 + J): NEXT J: ?A; I: B = B + A: A = 0: NEXT I: ?B
```

The values on the screen should now be:

```
10957 0
11305 1
9960 2
9948 3
9552 4
10368 5
11026 6
10897 7
10003 8
10831 9
104847
```

This again checks every 100 bytes, and the last figure is the total for all 100 bytes put together.

And so we proceed for every 1000 bytes, with the checksum program successively PEEKing at locations 4048, 5048, 6048 and so on up to and including 15048. The following twelve tables are what you should see on your screen after running each individual checksum program.

Program number 3

```
10329 0
8929  1
9530  2
8427  3
10554 4
9403  5
10260 6
11907 7
10711 8
10257 9
100307
```

Program number 4

```
10220 0
9806  1
9626  2
9439  3
11406 4
10796 5
10818 6
10940 7
10245 8
9370  9
102666
```

Program number 5

```
10181 0
7594  1
6874  2
10677 3
10488 4
10423 5
6911  6
7816  7
8764  8
8859  9
88587
```

Program number 6

```
9176 0
8387  1
9259  2
8810  3
8004  4
9311  5
8704  6
8394  7
8415  8
8360  9
86820
```

Program number 7

7825 0
9713 1
9411 2
8508 3
7828 4
9225 5
7733 6
8614 7
8162 8
8024 9
85043

Program number 8

8061 0
8840 1
7645 2
8377 3
8899 4
9706 5
9186 6
9451 7
8111 8
9774 9
88050

Program number 9

9191 0
8257 1
7900 2
8495 3
9215 4
10067 5
9743 6
9198 7
9245 8
9831 9
91142

Program number 10

9612 0
9104 1
7756 2
9844 3
9096 4
9593 5
9099 6
9614 7
9757 8
9315 9
92790

Program number 11

9672 0
9228 1
9459 2
8797 3
9587 4
9715 5
8935 6
8682 7
9283 8
8949 9
92307

Program number 12

8208 0
10912 1
8777 2
9277 3
10018 4
8582 5
7369 6
7144 7
7674 8
9190 9
87151

Program number 13

And finally, program number 14!

8437 0
8094 1
8329 2
10847 3
11339 4
9653 5
10169 6
11830 7
9181 8
11705 9
99584

11539 0
7345 1
3251 2
3368 3
4039 4
2809 5
2956 6
4775 7
12302 8
13070 9
65454

In fact, the last day's typing only goes from memory locations 15048 to 15904, some 856 bytes, so you'd be advised to check the last 56 bytes by yourself, and not really use the last two entries in table 14 at all.

Once you've finished your marathon typing session, that's it. A working compiler that will make your adventures run at extremely high speeds. As an experiment, I compiled the 'Enormous Cave' adventure in Chapter 9, to see what difference it would make - 'phenomenal' is the only word for it. In the original Basic program you can see the room descriptions being printed on to the screen at a relatively slow pace: reading speed, really. In the compiled version it just runs like dynamite!

A bit like comparing me and Steve Cram, I suppose. As an athlete he's like a compiled version of myself: ten times faster!

Conclusion

It is a lot of typing, I know, and you're bound to make mistakes typing the whole thing in. Use the checksum programs wisely, and at the end of fourteen days you'll have saved yourself upwards of £50 for what is quite a decent compiler.

After you've compiled your adventure, it is easy to make 'custom' listings of the finished program. At the moment, a listing of the compiler reads:

```
1 SYS2076 HELLO THERE V1
```

This you can easily change to read:

1 SYS2076 MY ADVENTURE 1

or whatever, by using the standard Commodore editing functions. Just be careful that you keep the line length exactly the same (the program won't run otherwise), and ensure that you leave a space after the SYS2076. Then, you can load and save your compiled program just like a normal Basic one.

To run it, LOAD "COMPILER",8,1 and RUN

It will ask you for a file name, so enter the name of a file on disk, and then go and make a cup of tea or something: it does take a while. But the wait is well worth it.

When it's finished, there'll be a copy of the program on disk, prefixed by the symbols C/. Just load and run it, and stand back and be amazed.

Then all you need is a turbo loader (for tape *or* disk products) and away you go! There are many on the market at the moment, and any one of them will help to give your finished adventures that 'polished' look.

If you're going to submit any programs based on the ideas, utilities and programming lessons in this book, an acknowledgment would be nice. Have fun.

11

Role-playing Games

In the previous chapters we've been talking about the creation of adventure games, games where one character is pitted against a selection of problems and, through his own ingenuity, has to overcome them.

Admittedly 'Enormous Cave' featured Eddy the underground dweller, a pirate and some dwarves, but they were all under the control of the program and either would, or wouldn't, do certain things. In other words, the dwarf might be six feet tall and weigh fourteen stone, or five feet two inches and seven stone, but he'd still behave in exactly the same manner. No allowance is made for the physical or mental attributes of the characters involved.

This is where role-playing games come in. To cover the subject fully would require a book all of its own, but in this chapter I hope to nudge you gently in the right direction and whet your appetite for more.

Some background

'Dungeons and Dragons' was the forerunner of all role-playing games, and led on to others such as 'Tunnels and Trolls', 'Traveller', and many more. In these games there is no pre-determined outcome as there is in the more conventional adventure games. There is no straightforward conclusion, although there might be plenty of treasures to collect, and the games never really end. Since there's no goal to the game, the only point in playing it is playing it, really, and to carry on developing the character that you have created.

The only difference between computer role-playing games and their earlier forefathers is, literally, the computer. Whereas before you would have a referee sorting everything out, consulting manuals and charts and generally taking a long time about things, now you have

the computer doing all that, leaving you, the player, to concentrate on the game.

Admittedly it is rather different to wander (in real-life) into a bedroom and pretend it is the lair of some hideous monster than it is to be told on the screen 'YOU ARE IN THE LAIR OF A HIDEOUS MONSTER'. However, the computer can print out much lengthier descriptions than that, and it can also add some graphics to illustrate the location. Games like 'The Temple of Apshai' cheated somewhat, in that the graphics appeared on the screen and the words appeared in an accompanying manual! Whichever method you prefer is up to you, although, I have said before, I would always prefer a lengthy text description on the screen rather than any amount of illustrations.

What really sets role-playing games apart, though, is the number of characters involved in them. Instead of just you against a host of problems, there may be many characters roaming about inside the game trying to do the same things as you. Some of those characters may be more experienced than you, and may spend a lot of time moving objects about and generally getting in the way. On the other hand they may just be starting on the quest, and you can then help or hinder them as you see fit. You discover who your friends are in this type of game!

However, if it's just you in front of the computer in the wee small hours of the morning, the other characters in this universe will be beyond your control, and you'll just have to act according to your personal attributes, which are determined at the start of the game. So, how do we go about setting up these attributes, and what does each one of the (usually) six mean?

Setting up a character

The six attributes cover both the mental and physical side of a character. The three mental ones are usually ego, intelligence and intuition, while the physical aspect of a character is handled by strength, dexterity and constitution. There are a number of ways in which a character can be set up, but it is usually handled at the very start of a game by the computer. Occasionally you may be asked to assign your own character, in which case you'll have to be very careful when dishing out the points. For example, a strong character with no intelligence is not likely to get very far, so make sure you haven't used up all your points by the time you get round to assigning intelligence.

Points? Well, normally, each attribute is rated on a scale of 3 to 18, with 3 being the lowest possible value. This is in accordance with the original 'Dungeons and Dragons' where the points were determined by rolling three dice. Each roll covered one of the attributes, and by adding up the score on the dice you got the value for that particular attribute. Of course, using the 64 you'd just generate 6 random numbers between 3 and 18.

This gives you the opportunity to assume a possible 16,777,216 different characters: more than enough.

The way these points are dished out will determine whether or not the character is going to get on in the adventure world that you create for him, so it's probably fair to allow the player another chance if he doesn't like the first set of attributes.

Now we'll look at each one in turn, and see how they would affect a player's performance.

Ego

This is a measure of mental toughness and willpower. On one level, it determines how a player will react if presented with a difficult situation. A player with a low ego may well give up, and thus not be able to fight a monster, leap for a high shelf, or whatever, whereas a player with a high ego is much more likely to do well and increase his efforts when the going gets tough.

On another level, it affects the way a player will perform when dealing with another player or character in the game. If you're trying to barter for something, a higher ego will probably get a lower price, and vice versa. If you're trying to talk your way out of a fight with a monster, a higher ego is more likely to succeed than a lower one.

Intelligence

This basically affects the way a player will get on in a game, whether he plays a rational, logical game, or a totally irrational and illogical one.

Naturally enough the former is more likely to succeed than the latter, and so whenever the player encounters a monster or some other character that other character will respond accordingly. If you leap

into battle every time without a second thought for your own safety, you're not going to survive very long.

Intuition

In the role-playing world this behaves exactly as it would in real life. A character with high intuition is far more likely to be aware of a monster lurking around the corner, or a secret trapdoor hidden in the wall, than a player with very low intuition.

To put it another way, a high intuition character will find secret doors and hidden passageways with ease, whereas a low intuition one will only discover hidden traps by falling into them.

Those are the three mental attributes, but the three physical ones are just as important in determining how a player is going to succeed (or fail) in the game.

Strength

This chiefly affects the way a player will perform in combat, although it also governs other things such as how many treasures a player can carry, which is in turn related to fatigue etc.

In combat, strength determines your ability to defeat an opponent, by gauging how heavy your blows are and how much damage they will inflict. Also, since a stronger player can carry and wield heavier weapons than a weaker one, he will be able to do that much more damage as well.

Stronger players therefore have a much better chance of surviving than weaker ones. However, if you're playing a role-playing game, don't put all your eggs in one basket: each category is important if your character is to do well.

Dexterity

This governs such factors as reflexes, speed, hand-to-eye co-ordination, and so on. In a fight, a highly dextrous character will be able to jump out of the way, whereas a less dextrous one will probably only jump into the way.

Also, in combat a dextrous player will strike more blows where he aims them and block more blows from his opponent than one who is less dextrous. This means that his chances of survival are that much higher.

Dexterity is also related to manipulation of objects. A dextrous player may well be able to pick up a tiny bottle and pour a potion into it, but a non-dextrous player will probably only succeed in pouring it all over the floor.

Constitution

This is possibly the most important attribute of all, since it determines how much suffering a player can take before giving up and lying down to die.

It covers such things as the number of blows and amount of damage that can be sustained in battle, how rapidly a player will tire (from moving or fighting) and so on. To a player of low constitution, even running away can be a hazardous exercise.

Thus constitution is linked to strength, in that it governs fatigue, and how long a player can keep going when the going gets tough.

And a seventh

To all these attributes must really be added a seventh, namely that of experience.

As the game progresses a player will learn many things, the position of various objects, the power of the assorted monsters, possibly even the attributes of some of his fellow players. Experience can play a vital part when starting to play a game, and so this too should be stored along with the player's other six attributes.

Starting to write a game

Rather like writing a conventional adventure game, you must never sit down at the computer and think 'My God! 30K of code to produce'. That won't get you anywhere. You should really start in the same place and begin by drawing up a map of the area. This time,

though, you don't need to worry about verbs and nouns, since there aren't going to be that many. Fighting and casting spells, examining things and opening things, getting items and dropping them, and the traditional HELP, SCORE, QUIT, INVENTORY etc. are just about all we need.

Once you have drawn the map, and decided whether you're going to be graphical or textual, you might as well write or illustrate all the rooms and get that out of the way.

Then you'll need to have a little routine to set up the character's attributes. Here's how one program set about it, in a slightly unorthodox game.

```

10 POKE53281,0:POKE53280,9:PRINT"[CLR,YEL]
   ****TENNIS MENACE****"
12 GOSUB10000
100 REM START OF GAME PROPER
1000 POKE53272,23:PRINT"[2CD]IN THIS GAME YOU ARE
   IN CHARGE OF A      TENNIS HERO."
10001 REM GOSUB14100
10002 PRINT:PRINT"IT IS YOUR JOB TO GUIDE YOUR HER
   O TO      FAME, FORTUNE, AND WIMBLEDON.
10004 PRINT:PRINT"OF COURSE, DISASTERS MAY BEFALL
   YOU EN ROUTE.REFERRING TO THE ";
10006 PRINT"COURT THAT YOU'REPLAYING ON AS ";CHR$(
   34);"THE PITS OF THE EARTH";
10008 PRINTCHR$(34);"      WILL NOT ENHANCE YOUR CHAN
   CE OF QUALIFY-ING FOR WIMBLEDON."
10010 PRINT:PRINT"HOWEVER, SEEK OUT THE RIGHT SPON
   SORSHIP DEALS, ATTEND THE RIGHT ";
10012 PRINT"TOURNAMENTS, BUYTHE RIGHT KIT AND USE
   THE SERVICES OF  THE ";
10014 PRINT"BEST TRAINERS AND YOU MIGHT ([RVS]MIGH
   T[OFF]) STAND A CHANCE."
10016 GOSUB12000
10018 PRINT"[CLR,PUR]OKAY, LET'S START THE GAME."
10020 PRINT"[2CD,YEL]HAVE YOU A PREVIOUS GAME SAVE
   D TO TAPE (Y OR N)?"
10022 GETPB$:IFPB$="Y"THEN13000
10024 IFPB$="N"THEN10028
10026 GOTO10022
10028 PRINT"[2CD,GRN]LET'S HAVE YOUR NAME.":PRINT"
   [CD]NAME PLEASE ? ";:GOSUB60000
10030 FORI=1TOLEN(CM$):IFMID$(CM$,I,1)=" "THEN1003
   4
10032 NEXT
10034 N1$=LEFT$(CM$,I-1):N2$=MID$(CM$,I+1)
10036 PRINT"[CD,GRN]OKAY ";N1$;" WHAT DO YOU WANT

```

```

";PRINT"TO CALL YOUR PLAYER?"
10038 PRINT"[CD]PLAYER'S NAME ? ";:GOSUB60000
10040 FORI=1TOLEN(CM$):IFMID$(CM$,I,1)=" "THEN1004
4
10042 NEXT
10044 N3$=LEFT$(CM$,I-1):N4$=MID$(CM$,I+1)
10046 PRINT"[2CD,PUR]RIGHT, LET'S GIVE YOUR PLAYER
SOME CHARACTER.":GOSUB12000
10048 PRINT"[CLR] [WHT]***CHARACTER ROLE*
**"
10050 PRINT"[2CD,YEL]YOUR PLAYER HAS SIX ATTRIBUTE
S, WITH EACH ATTRIBUTE BEING ";
10052 PRINT"SCORED ON A BASIS OF 1 TO 5. FOR EXA
MPLE, AN ATTRIBUTE OF5 ";
10054 PRINT"FOR AGGRESSION WOULD MEAN THAT "
10056 PRINTN3$;" ";N4$;" WAS PRETTY AGGRESSIVE.":PR
INT
10058 PRINT"A RATING OF 1 WOULD MEAN THAT ":PRINTN
3$;" "N4$;" WOULD WANT THEIR MUM"
10060 PRINT"IF A POINT WAS AWARDED AGAINST THEM."
10062 PRINT"[2CD,CYN]YOU HAVE A TOTAL OF 20 POINTS
TO AWARD YOUR HERO OVER ALL 6 ";
10064 PRINT"ATTRIBUTES. AS A HINT, DON'T TRY AND
GIVE HIM/HER MAXIMUM";
10066 PRINT"POINTS FOR EVERYTHING.":GOSUB12000
10068 PRINT"[CLR] [WHT]***CHARACTER ROLE*
**"
10070 PRINT"[2CD,YEL]YOUR ATTRIBUTES ARE:"
10072 PRINT"[2CD,RED]INTELLIGENCE":PRINT"AGGRESSIO
N":PRINT"STRENGTH":PRINT"INTUITION"
10073 PRINT "SPEED"
10074 PRINT"DEXTERITY"
10076 PRINT"[CD,YEL]NOW AWARD YOUR POINTS."
10078 PRINT"[CD]INTELLIGENCE ? ";:GOSUB60000:A(1)=
VAL(LEFT$(CM$,1))
10080 PRINT"AGGRESSION ? ";:GOSUB60000:A(2)=VAL(LE
FT$(CM$,1))
10082 PRINT"STRENGTH ? ";:GOSUB60000:A(3)=VAL(LEFT
$(CM$,1))
10084 PRINT"INTUITION ? ";:GOSUB60000:A(4)=VAL(LEF
T$(CM$,1))
10086 PRINT"SPEED ? ";:GOSUB60000:A(5)=VAL(LEFT$(C
M$,1))
10088 PRINT"DEXTERITY ? ";:GOSUB60000:A(6)=VAL(LEF
T$(CM$,1))
10090 A=0:FORI=1TO6:A=A+A(I):IFA(I)>5THENPRINT"NAU
GHTY!":FORJ=1TO1000:NEXT:GOTO10068
10092 NEXTI:IFA>20THENPRINT"NAUGHTY!":FORJ=1TO1000
:NEXT:GOTO10068
10094 GOSUB12000
10096 PA=50:REMPLOYER APPEAL ON A SCALE OF 0 TO 30

```

```

0
10098 PRINT"[CLR,CD]FINAL QUESTION : IS YOUR PLAYE
R MALE OR FEMALE (M OR F)?"
10100 GETPS$:IFPS$="M"THENM=1:F=0:GOTO10106
10102 IFPS$="F"THENM=0:F=1:GOTO10106
10104 GOTO10100
10106 PRINT"[CD,GRN]YOUR LAST LOOK AT EVERYTHING B
EFORE WE START:"
10108 PRINT"[2CD]AS MENTOR, YOUR NAME IS ":PRINTN
1$;" "N2$
10110 PRINT"[CD]OUR HERO'S NAME IS ":PRINTN3$;" "
N4$
10112 PRINT"[CD]AND ";N3$;" IS HOPEFULLY ":IFM=1T
HENPRINT"[PUR]MALE.":GOTO10116
10114 PRINT"[PUR]FEMALE."
10116 PRINT"[CD,YEL]PERSONALITY RATINGS ARE:"
10118 PRINT"[2CD,RED]INTELLIGENCE ";A(1):PRINT"AGG
RESSION ";A(2):PRINT"STRENGTH ";A(3)
10120 PRINT"INTUITION ";A(4):PRINT"SPEED ";A(5):PR
INT"DEXTERITY ";A(6)
10122 GOSUB12000
11999 END
12000 PRINT"[CD]          [RVS,YEL]PRESS 'SPACE' TO C
ONTINUE."
12002 FORI=1TO10:GETSP$:NEXT
12004 GETSP$:IFSP$<>" "THEN12004
12006 RETURN
13000 REM LOAD PREVIOUS GAME
13999 GOTO100
14000 REM 25 TOURNAMENTS, 20 MALE PLAYERS, 20 FEM
ALE, 20 SPONSORS
14100 FORI=1TO25:READTR$(I):NEXT:FORI=1TO20:READPM
$(I):NEXT
14110 FORI=1TO20:READPF$(I):NEXT:FORI=1TO20:READSS
$(I):NEXT
14112 RETURN
50000 DATA"WIGAN ODEON","BLACK HILL FIELDS","ROCHD
ALE OLYMPIA","HOOLIGANS"
50002 DATA"UGANDAN OPEN","SWISS CHAMPIONSHIPS","WR
ECKYAVIC OPEN","BUMBLEDON"
50004 DATA"HENLEY HILL","CRYSTAL PALACE","ALBERT H
ALL","BAMPSTEAD BEATH"
50006 DATA"WEASDEN OPEN","UGLYSVILLE","SCARSDALE",
"ARGUS OPEN"
50009 DATA"STALLA ERTUIS","HENSON & BEDGES","SCOTC
H WHISKY","BURLIMANS"
50010 DATA"FOREST HILLS","FRENCH OPEN","AUSTRALIAN
OPEN","WIMBLEDON"
50011 DATA"DAVIS CUP"
50020 DATA"BJORN BORG","JOHN MCENROE","IVAN LENDL"
,"JIMMY CONNORS","STAN SMITH"

```



```

50022 DATA"ILIE NASTASE","ROSCOE TANNER","ROD LAVE
R","PETER DENNING"
60000 CM$=""
60002 PRINT"[RVS,YEL]*[OFF,CL]";
60004 GETZ$: IFZ$="" THEN 60004
60006 Z=ASC(Z$): IFZ<32 AND Z<>13 AND Z<>20 THEN 60004
60008 ZL=LEN(CM$): IFZL>20 THEN 60012
60010 IFZ>31 THEN CM$=CM$+Z$: PRINTZ$: GOTO 60002
60012 IFZ=13 AND ZL THEN PRINT " ": RETURN
60014 IFZ=20 AND ZL THEN CM$=LEFT$(CM$,ZL-1): PRINTZ$:
60016 GOTO 60002

```

READY.

No need to explain the listing, it's all fairly straightforward, but you can see how the attributes are distributed in this one, and checks made to see whether or not a player has tried to create a super-human character for himself.

Then to the complicated part of the process: the other characters in your particular role-playing universe. Are they all going to be governed by the computer, or are you going to allow more than one player to take part? Indeed, can your player control more than one character?

This, and a whole lot more questions, will have to be answered before you can begin to complete the program.

Conclusion

Although we haven't got the space to go too deeply into role-playing games in this book, you should have begun to get some idea of the concepts involved.

Role-playing games are a real alternative to adventure games, because they are different every time you play them. For any one particular game you might like to have a whole host of different characters stored on disk or tape and see how the game varies each time you play it using those different characters. You might have one strong character, one intelligent one, one dextrous one, and so on.

As we said at the start, the games never really finish, and you could go on developing your various characters for months without ever getting tired of playing the game.

A good place to start, if you want some experience of playing this type of game before attempting to write one, is with the Epyx series of games under the general heading of 'Dunjonquest'. They are marketed by Automated Simulations Inc. The main game, 'The Temple of Apshai' was once voted the computer game of the year, and although that may be stretching a point it is an excellent game to begin with. It covers most of the points that we've mentioned in this chapter, as well as a good deal more, and is well worth a look.

There are also a number of expansion units to this main game, such as 'The Curse of Ra', 'The Upper Reaches of Apshai', and so on, but you'll still need the original 'Temple of Apshai' to play them, so you might as well get that one first to see if you like the genre.

And with that, happy role-playing!

12

Conclusion

Adventure games are not that difficult to write, but not everyone can write an adventure. Rather like a novelist, the would-be adventure writer must have a fertile imagination, although as we've seen it is possible to crib from the classics and get your storyline that way. However, even that is not a guarantee that the finished game will be on a par with those of Level Nine, Infocom, and the like.

Unlike a novel writer, though, the adventure programmer needs a lot more than just a typewriter (or word processor!) and some sheets of paper. A blank screen can be just as daunting as a blank sheet of paper, but I hope that the routines presented in this book will enable a programmer to get the best out of the both the Commodore 64 and his abilities as an adventure writer.

Text or graphics, splitting the screen or not, using disk or tape, what sort of parser to use, whether the text should be compressed or not, are all problems that we've looked at throughout the course of this book. Whichever way you turn your programming abilities is of course up to you, but it is best to stick with a subject that you know something about. If your interest in reading tends towards the science-fiction world, why not try to write a science-fiction adventure game? On the other hand, if you prefer the novels of Agatha Christie, there haven't been that many good detective adventure programs.

Developing adventures

At a recent computer exhibition, a group of us were gathered around talking about adventure games. Three of us were writers (and players) of these games, and another couple were very experienced adventure players. For would-be adventure game writers, our conversation would have made an interesting transcript, but since I

neither have that nor a photographic memory, perhaps just a few hints will serve my purpose.

The writers among us were complaining (as usual) about the level of complexity that a game has to offer. Produce a relatively simple game and everyone solves it first time around, but a game that is so difficult that most people never even leave the first location will probably put a number of players off adventure games for life. Since you have written the game, the solution will be obvious to you, but to another player the game may be hopelessly complicated.

A useful thing to do, after having written the game, is to invite some friends around and let them play it while you're watching. A nerve-racking experience, but the final game will be all the better for their comments. You may think that dropping the rod and then picking up the bird before putting it in the cage is an obvious solution, but I've heard of people trying to feed the bird sandwiches, strangle the bird, throw the cage in the bizarre hope that it will land on top of the bird, and many more weird suggestions that aren't worthy of repetition in a book as wholesome as this one.

The best adventures tend to be the most logical ones, and that is a point that every adventure author should bear in mind. Even great writers suffer from time to time, though, and in (I think) 'Zork III' there is a ridiculous situation where you have to turn a bucket into a well by pouring water into it, or so I'm reliably informed. Personally, that is *not* a solution that I would have thought of.

Still, I think we can forgive Infocom the odd lapse or two: I've yet to hear of a serious bug in any of their programs.

The future

But what of the future? Traditional adventure games like 'Colossal Cave Adventure' have now been played for upwards of six years, and newer ones like 'Zork' and 'Multi-user Dungeon' look likely to have an equally long life. As newer and more powerful computers appear, so will newer and more powerful adventure games. The first games are (generally speaking) never among the best.

People keep trying to persuade us that the so-called arcade-adventures herald the future of this type of game, but I doubt it. I have seen one game that was definitely of this genre, requiring a joystick and most of the keyboard to play it. What has happened

to the times when we used to spend hours hunched over a keyboard, peering at the computer screen (usually the only source of light in the room), trying to glean some idea of what the next move should be?

Despite reviewers and shops, thankfully a lot of people still seem keen on the traditional adventure way of thinking and writing, and I hope you do too. Looking outside, the lights grow dim and it is time, for the umpteenth occasion, to load 'Zork' into the 64 and go on another journey in the underworld.

I hope this book will nudge you along into your own world of adventure writing, and if just one of you can come up with a game that will keep me glued to the keyboard trying to solve it, then the purpose of this book will have been achieved.

Happy adventuring!

FURTHER ADVENTURES ON THE COMMODORE 64/128

A disk containing the complete *Enormous Cave* adventure, *The Lost Wallet* and the Basic Compiler is available direct from Duckworth at £9.95, post free. Send your cheque/postal order to the address shown below or alternatively telephone 01-485 3484 with your Access or Barclaycard number.

Peter Gerrard's first adventure book, *Exploring Adventures on the 64*, is also available at £6.95, post free. It contains three complete adventures for the 64.



DUCKWORTH

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



Index

Adams, Scott, 9, 102

Away verb, 192

Butterfield, Jim, 61

Buy verb, 182-3

Character designer, 15-19

Character sets, 14ff

Chase verb, 188

Close verb, 176

Colossal Cave, 9, 99, 322, 323

Colour changes, 76-8

Compiler, 270ff

Computed GOTOs, 160

Cross verb, 177

Crowther, Willie, 9

Data conservation, 63-4

Data storage, 68-71

Discard verb, 182-3

Disk drives, 60ff, 99, 100, 194ff

Drop verb, 172-3

Enormous Cave data, 260ff

Enormous Cave introduction,
194-6

Enormous Cave listings, 194ff

Enormous Cave main listing, 196ff

Enormous Cave messages, 245ff

Enormous Cave program, 99,
194ff, 311

Enormous Cave rooms, 214ff

Enter verb, 192

Examine verb, 178-9

Find verb, 180-1

Flash command, 78-81

Function keys, 72-8

Get verb, 170-1

Give verb, 191

Go verb, 167

Graphics, 13ff, 99

Help verb, 165

High resolution, 29-34

Insert verb, 186-7

Interrupts, 35ff

Inventory verb, 166

Jump verb, 167

Listen verb, 192

Load verb, 189

Look verb, 166

Lost Wallet listing, 105-10, 159ff

Lost Wallet program, 99ff, 158ff

Lost Wallet verb lists, 164-93

Map designing, 101, 102

Memory saver, 47, 48

Message storage, 68-71

M.U.D., 10

Musical interrupts, 49-59

Offer verb, 191

Open verb, 174-5

Parsers, 112ff

Parsers in Basic, 113-16, 161-2

Parsers in M/C, 117-46

Pick verb, 184-5
Problem designing, 102, 103

Quit verb, 164-5

Raster interrupts, 35-47
Read verb, 184-5
Redescribe verb, 166
Renummer program, 81-98
Role-playing games, 311ff

Save verb, 189
Saving memory, 61-3
Score verb, 166
Screen layouts, 100
Search verb, 191
Sequential files, 64-7
Sound, 48ff
Split screens, 27, 28
Sprite designer, 21-6

Sprites, 20ff

Take verb, 184-5
Talk verb, 190
Tape decks, 60ff, 99, 100, 158, 189
Text compression, 147ff
Text compression in Basic, 148-50
Text compression in M/C, 150-7
Tokenising, 148-50

Vocabulary designing, 103, 104, 105

Wear verb, 180-1
Woods, Don, 9

Zork, 9, 13, 76, 111, 112, 323

DUCKWORTH HOME COMPUTING

THE ODYSSEY by Mike & Peter Gerrard

'This is a big, interesting and well-planned adventure which should ensure you get your money's worth.' *Your Computer*

'Fat enough, and tricky enough, to give many, many hours of arcane entertainment.' *Commodore User*

Homer's epic tale, *The Odyssey*, composed nearly 3000 years ago, is one the first adventures ever written. It relates the heroic story of Odysseus, a story adventure players can now relive in a mammoth 80K text adventure for the 64.

The ten-year siege of Troy by the Greeks is at an end, and as Odysseus you must make the perilous journey from Asia Minor to reclaim your throne in Ithaca. Only a full crew and regular supplies of food and water will see you through your battles against the Sirens, the Lotus Eaters, the race of giants, the one-eyed Cyclops, the six-headed monster Scylla and the whirlpool Charybdis. Can you escape the nymph Calypso, survive the wrath of the god Poseidon, and finally defeat your challengers in the Battle in the Hall?

The game is divided into two parts on separate cassettes, each of which employs a turboload system which considerably reduces loading time. The parts may be played in either order and you are allowed to save your progress at any point. Cassette £9.95

THE COMMODORE 64/128 ADVENTURE PACK

by Peter Gerrard

This pack contains four superb text adventures on disk as well as a copy of *The Adventurer's Notebook* by Mike Gerrard, which will guide you into the world of adventures. The adventures included are: Colossal Cave, the very first adventure, originally written on a mainframe computer; True Spit, set in the wild west where you are a bounty hunter out for blood; Spaced Out, in which you are in charge of a secret formula in the depths of space; Underground Adventure, in which you find yourself exploring a vast network of caverns. Each adventure allows you to save your progress on disk at any time. £14.95

THE COMMODORE PLUS/4 ADVENTURE PACK

by Peter Gerrard

As above but for the Plus/4. £14.95

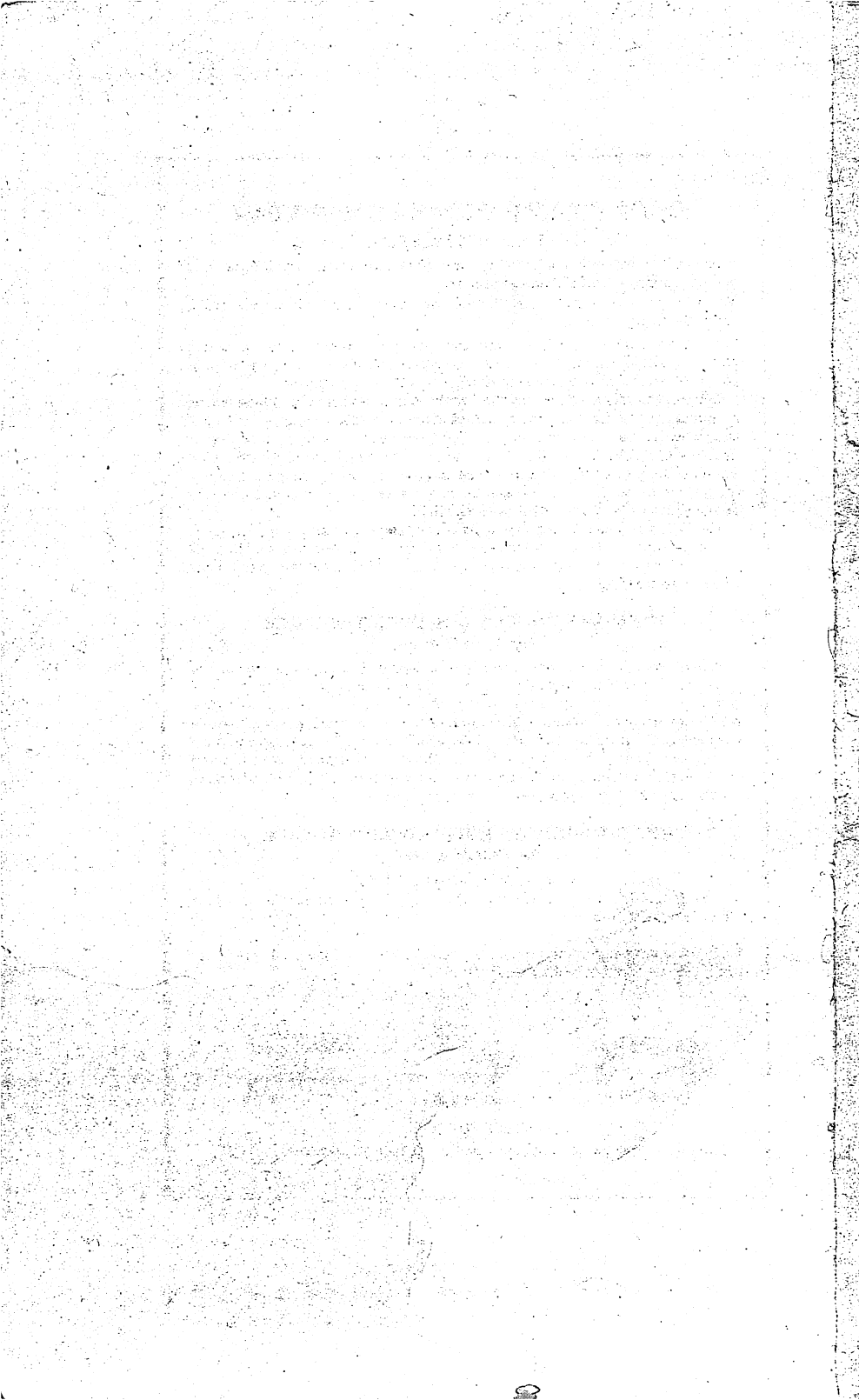
'Worth the £14.95 it costs just to get a copy of the classic Colossal Cave for your Plus/4.' *Your Commodore*

Many other adventures and books are available. Write in for a catalogue of our Commodore products, or telephone 01-485 3484.



DUCKWORTH

The Old Piano Factory, 43 Gloucester Crescent, London NW1 7DY



FURTHER ADVENTURES ON THE COMMODORE 64/128

by Peter Gerrard

Peter Gerrard's *Exploring Adventures on the Commodore 64* explained clearly how to write text-based adventures in Basic. This, his latest book, will transform your programming techniques. Assuming an average knowledge of Basic, it covers text compression, tape-based, disk-based and role-playing games, incorporating graphics and sound, improving program response times and creating sophisticated parsers in both Basic and machine code to give your adventures professional polish.

Enormous Cave, a full-blown disk-based adventure and sequel to *Colossal Cave Adventure*, is included to demonstrate how to create mammoth games with Infocom-like complexity. Each part of the listing is fully explained, showing how to incorporate hundreds of locations, an extended vocabulary, lengthy room descriptions and numerous messages. A tape-based adventure, a demonstration of role-playing games, and a complete BASIC COMPILER – which will make your programs run at lightning speed – all combine to make this book a comprehensive guide to the intricacies of writing adventures.

Peter Gerrard, former editor of *Commodore Computing International*, is the author of many Duckworth books and adventure games including *Using the 64* and *Advanced Basic and Machine Code for the Commodore 64*.

ISBN 0-7156-1859-8



9 780715 618592

Duckworth
The Old Piano Factory
43 Gloucester Crescent
London NW1 7DY

ISBN 0 7156 1859 8

IN UK ONLY £9.95 NET