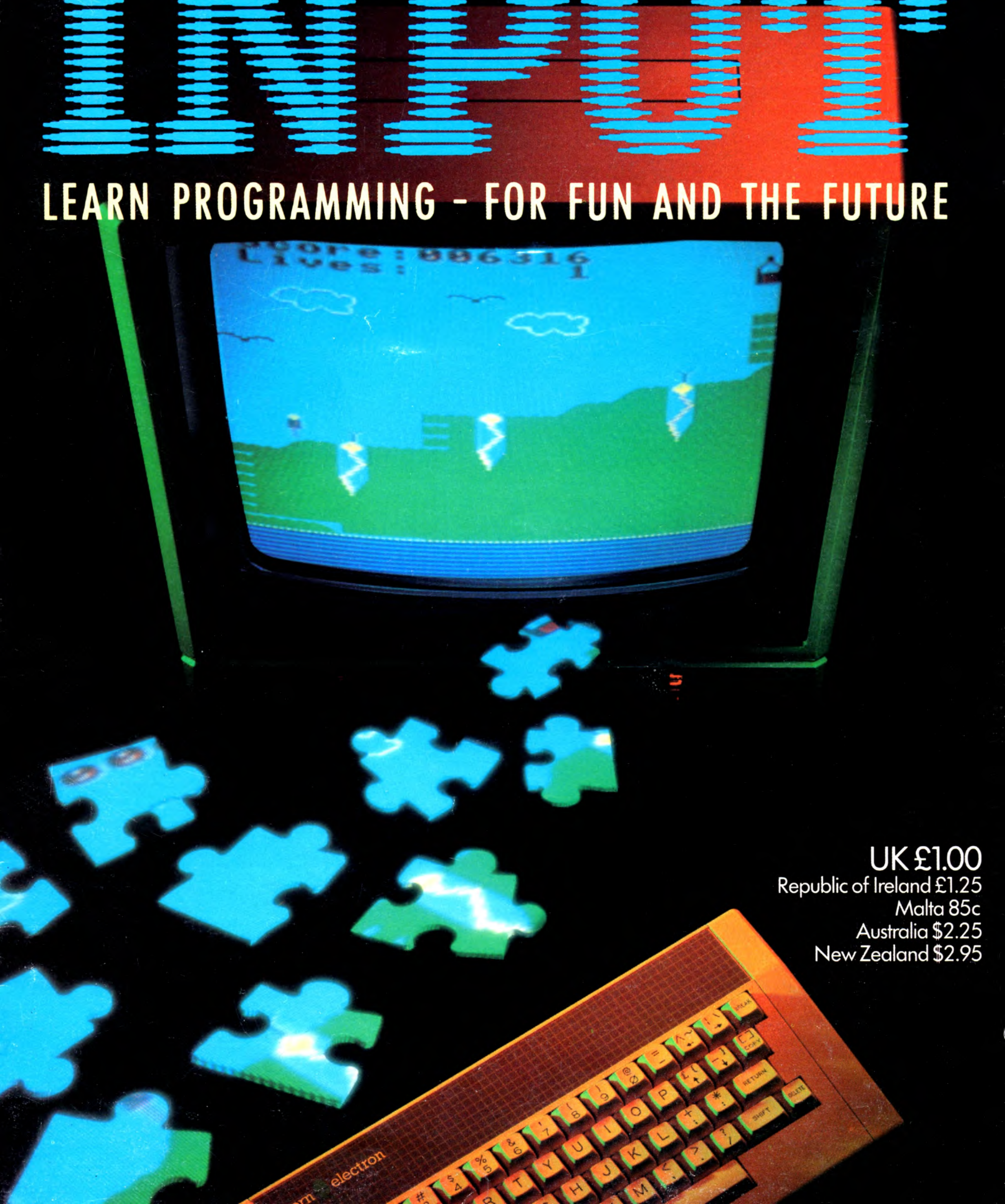
A MARSHALL CAVENDISH 50 COMPUTER COURSE IN WEEKLY PARTS

INPUT

LEARN PROGRAMMING - FOR FUN AND THE FUTURE



UK £1.00
Republic of Ireland £1.25
Malta 85c
Australia $2.25
New Zealand $2.95

INPUT

Vol. 4 No 50

INDEX

The last part of INPUT, Part 52, will contain a complete, cross-referenced index. For easy access to your growing collection, a cumulative index to the contents of each issue is contained on the inside back cover.

PICTURE CREDITS

Front cover, Graeme Harris. Pages 1557, 1558, Digital Arts. Pages 1562, 1564, Tony Lodge. Pages 1564, 1565, M. Furminger. Pages 1566, 1567, Pete Beard. Page 1567, Rick Blakely. Pages 1568, 1572, Graeme Harris. Pages 1577, 1578, 1579, Joe Wright. Pages 1581, 1582, 1584, Graeme Harris. Colour monitor by courtesy of Fidelity PLC.

Published by Marshall Cavendish Partworks Ltd, 58 Old Compton Street, London W1V 5PA, England. Typeset by MS Filmsetting Limited, Frome, Somerset. Printed by Cooper Clegg Web Offset Ltd, Gloucester and Howard Hunt Litho, London.

There are four binders each holding 13 issues.

HOW TO ORDER YOUR BINDERS

UK and Republic of Ireland:
Send £4.95 (inc p & p) (IR£5.95) for each binder to the address below:

Marshall Cavendish Services Ltd,
Department 980, Newtown Road,
Hove, Sussex BN3 7DN

Australia: See inserts for details, or write to INPUT, Times Consultants, PO Box 213, Alexandria, NSW 2015

New Zealand: See inserts for details, or write to INPUT, Gordon and Gotch (NZ) Ltd, PO Box 1595, Wellington

Malta: Binders are available from local newsagents.

BACK NUMBERS

Back numbers are supplied at the regular cover price (subject to availability).

UK and Republic of Ireland:
INPUT, Dept AN, Marshall Cavendish Services,
Newtown Road, Hove BN3 7DN

Australia, New Zealand and Malta:
Back numbers are available through your local newsagent.

COPIES BY POST

Our Subscription Department can supply copies to any UK address regularly at £1.00 each. For example the cost of 26 issues is £26.00; for any other quantity simply multiply the number of issues required by £1.00. Send your order, with payment to:

Subscription Department, Marshall Cavendish Services Ltd,
Newtown Road, Hove, Sussex BN3 7DN

Please state the title of the publication and the part from which you wish to start.

HOW TO PAY: Readers in UK and Republic of Ireland: All cheques or postal orders for binders, back numbers and copies by post should be made payable to:
Marshall Cavendish Partworks Ltd.

QUERIES: When writing in, please give the make and model of your computer, as well as the Part No., page and line where the program is rejected or where it does not work. We can only answer specific queries – and please do not telephone. Send your queries to INPUT Queries, Marshall Cavendish Partworks Ltd, 58 Old Compton Street, London W1V 5PA.

INPUT IS SPECIALLY DESIGNED FOR:

The SINCLAIR ZX SPECTRUM (16K, 48K, 128 and +), COMMODORE 64 and 128, ACORN ELECTRON, BBC B and B+, and the DRAGON 32 and 64.

In addition, many of the programs and explanations are also suitable for the SINCLAIR ZX81, COMMODORE VIC 20, and TANDY COLOUR COMPUTER in 32K with extended BASIC. Programs and text which are specifically for particular machines are indicated by the following symbols:

SPECTRUM 16K, 48K, 128, and + **COMMODORE 64 and 128**

ACORN ELECTRON, BBC B and B+ **DRAGON 32 and 64**

ZX81 **VIC 20** **TANDY TRS80 COLOUR COMPUTER**

PUZZLES, COMPUTERS AND MATHEMATICS

- TYPES OF PUZZLES
- USING THE COMPUTER
- SIMULTANEOUS EQUATIONS
- TRIAL AND ERROR
- MATHEMATICAL TRICKS

Sharpen your wits and test your programming skills with these mind-bending puzzles. You'll also learn a few new techniques for solving sets of equations

By this stage of *INPUT*'s course you should have a good grounding in most of the BASIC programming techniques. Given a problem, you should be able to write a program to solve it (assuming it can actually be solved). But unless you have a particular hobby or project that makes use of the computer, it's often difficult to think up things to try out.

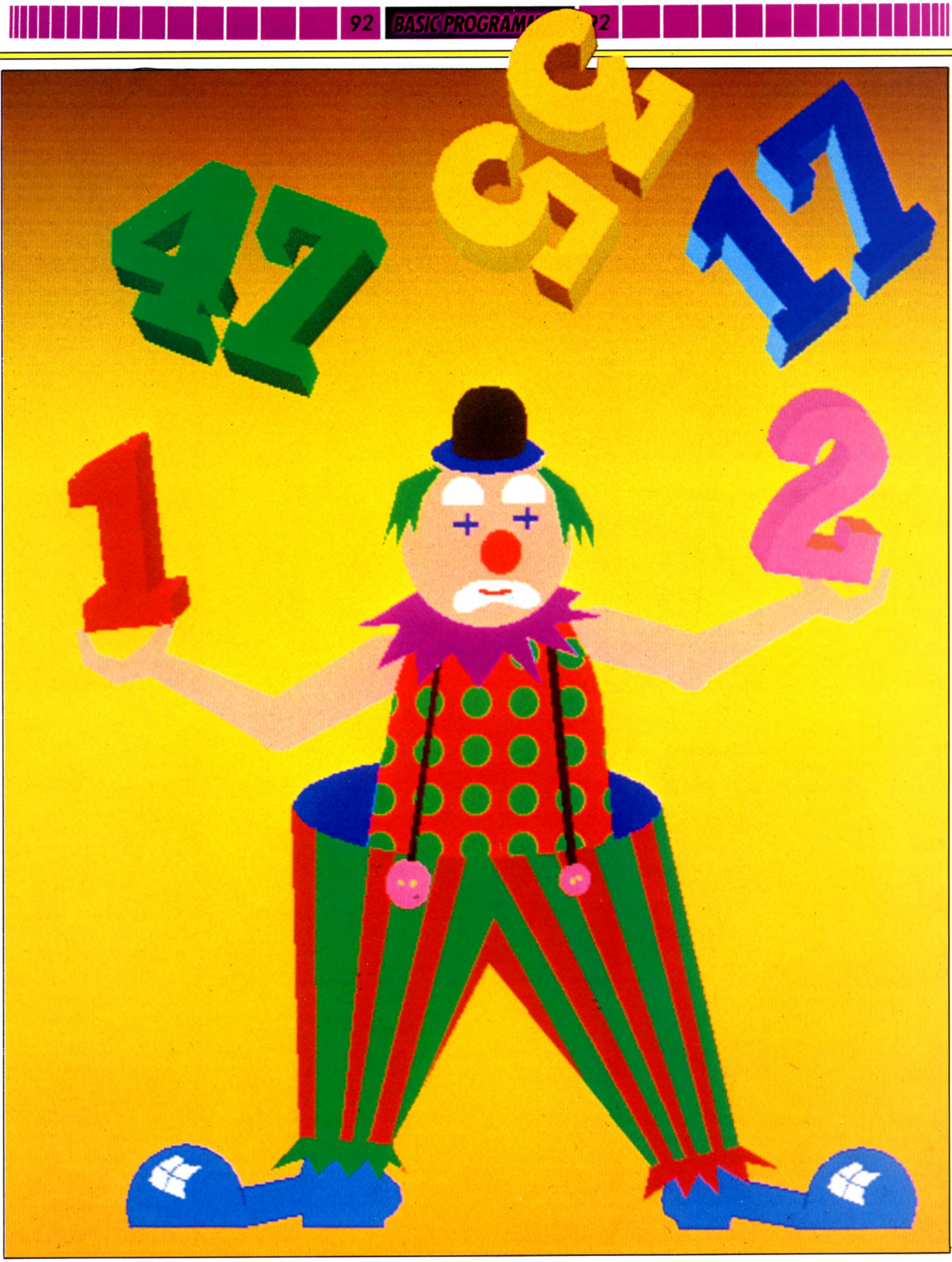
Puzzles offer a welcome challenge as they are an enjoyable way to practise programming and keep your skills up to scratch. They have been popular for thousands of years. The Egyptians were known to enjoy solving puzzles and the Ancient Greeks were extremely fond of mathematical and logical puzzles and paradoxes. In fact, many problems that started out as recreations have led on to important discoveries in maths and science.

Solving a problem successfully gives you a feeling of satisfaction and, if you enter some of the growing number of competitions in computer magazines you could even win yourself some prizes. But solving puzzles can also teach you more about programming than you can learn from simply reading about it because it forces you to work out your own methods and try out your own ideas.

COMPUTER PUZZLES

There are many different types of puzzles, not all of which are suitable for solving on a computer. Many simply require a flash of insight or some logical thinking. A classic example of the logical thinking type is this one. A man has three possessions: a wolf, a sheep and a cabbage which he has to ferry across a river. But the boat is very small and he can only take one thing at a time. And if left alone together, the wolf will kill the sheep, or the sheep will eat the cabbage. So how can he take everything across the river intact?

This *can* be solved on a computer (you can try and write a program if you like) but it is


47
17
1
2

much quicker to do it with paper and pencil or in your head.

The types best solved by computer are those where you are given a certain number of facts and figures and are asked to provide the value of some unknown. Others that work on a computer ask you to perform complicated tasks in arithmetic or geometry. In these cases it is quicker to write a program to solve the problem rather than to do it yourself.

This article will show you how to solve three of the most common types of puzzles. Before you look at the solutions, it is worth spending some time trying to solve them on your own. Only then turn to the programs and explanations of how they work.

SIMPLIFYING THE PROBLEM

If you are unused to solving puzzles then they may all seem rather difficult at first, as it takes a bit of practice to sort out the essential information from the mass of confusing words. In fact, the first type of puzzle relies on the words to mask what is often a quite simple problem. When stripped of the words, the puzzle usually amounts to solving a number of simultaneous equations. Here's a simple problem that you can solve without a computer so you understand the principles involved.

A group of friends visit a cafe and buy three coffees and two teas and the bill comes to 176 pence. The next day someone else joins them and they have twice as many teas but one fewer coffee. This time the bill is 16 pence more. How much is a cup of tea?

If you remove all the inessential information and use letters instead of words the problem reduces to $3c+2t=176$ and $2c+4t=192$.

These are called *simultaneous* equations because they are both true at the same time. In order to solve sets of simultaneous equations you need as many equations as there are unknowns. In this case there are two unknowns—c and t—and two equations, so they can be solved. The equations are also *linear* since none of the variables are raised to any power—$3c^2+2t=176$, for example, is a non-linear equation and is solved by a different method.

To solve the puzzle, rearrange the second equation to give you $c=(192-4t)/2$ then substitute this into the first one. You'll get $3(192-4t)/2+2t=176$. Rearranging this again gives $t=28$, so a cup of tea costs 28 pence.

Equations with two unknowns are easy to solve like this and three unknowns are not too difficult. But beyond this you definitely need the computer to do the hard work for you.

THE STAMP DEALER

Here's the first real puzzle for you to solve.

A stamp dealer has a box of foreign stamps that he plans to sell in packs of six different values. The price of the packs are coded from A to F. Six young members of a stamp club go along and spend all of their pocket money on the following:

	A	B	C	D	E	F	Cost
Sheila	6	2	3	1	1	2	341p
Trevor	14	11	0	1	2	1	469p
Anita	0	1	3	6	4	3	598p
Maggi	5	3	5	2	1	1	376p
Phil	12	1	4	4	3	2	587p
Sue	8	0	1	1	0	3	293p

How much did each pack cost?

Trying to solve these equations by eliminating one variable at a time is possible but it would take a very long time and would be prone to arithmetical errors. The following program will solve this puzzle and any other sets of linear simultaneous equations, many of which crop up in all sorts of problems, not only puzzles.

There are several ways of solving sets of equations like these and you may be able to find a different method, but the program below is quick and relatively short:

TROUBLESHOOTER

Singular equations

There are a few special—usually silly—situations in which simultaneous equations have either ambiguous solutions or no solutions at all. The equations $x=2$ and $x=3$ obviously cannot be simultaneously true, and even if you have as many equations as unknowns, the equations $x+y=3$ and $x+y=1$ are insoluble simultaneously. Also, the equations $x+y=2$ and $2x+2y=4$ are not *independent*, they are essentially the same equation and have many possible solutions.

When you have the right number of equations and yet they are inconsistent or not independent the equations are called *singular*.

Non-independent equations can be solved using a method of trial and error as used in, the Christmas presents puzzle. But several solutions are possible and you need extra information to pick out the correct one.

```
10 INPUT "ENTER NUMBER OF ROWS□";R
15 LET C=R+1
20 DIM A(R,C): DIM B(R,C): DIM A$(C-1,20)
30 FOR K=1 TO C-1
40 INPUT "ENTER NAMES FOR
   COLUMNS□",A$(K)
50 NEXT K
60 FOR J=1 TO R
70 PRINT : PRINT "ENTER VALUES FOR
   ROW□";J
80 FOR K=1 TO C
90 INPUT A(J,K)
95 PRINT A(J,K);"□";
100 LET B(J,K)=A(J,K)
110 NEXT K: NEXT J
120 FOR L=1 TO R
130 GOSUB 230
140 GOSUB 280
150 NEXT L
160 CLS
170 FOR K=1 TO C-1: PRINT AT
   1,4*K-4;A$(K): NEXT K
180 FOR J=1 TO R: FOR K=1 TO C: PRINT
   AT 1+J, K*4-4;B(J,K)
190 NEXT K: NEXT J
200 PRINT "ANSWERS:—"
210 FOR K=1 TO C-1: PRINT AT
   4+R,K*4-4;A(K,C): NEXT K
220 STOP
240 LET D=A(L,L)
250 FOR K=1 TO C
260 LET A(L,K)=A(L,K)/D
270 NEXT K: RETURN
290 FOR J=1 TO R
300 IF J=L THEN NEXT J: RETURN
310 LET F=A(J,L)
320 FOR K=1 TO C
330 LET A(J,K)=A(J,K)-F*A(L,K)
340 NEXT K: NEXT J: RETURN
```

```
10 INPUT"ENTER NUMBERS OF ROWS AND
   COLUMNS";R,C
20 DIM A(R,C),B(R,C),A$(C-1)
30 FORK=1TOC-1
40 INPUT"NAMES FOR COLUMNS";A$(K)
50 NEXT
60 FORJ=1TOR
70 PRINT"ENTER VALUES FOR ROW□";J
80 FORK=1TOC
90 INPUTA(J,K)
100 B(J,K)=A(J,K)
110 NEXT:NEXT
120 FORL=1TOR
130 GOSUB230
140 GOSUB280
150 NEXT
160 PRINT"♥"
170 FORK=1TOC-1:PRINTTAB(8*K-8);
```

```
    A$(K);:NEXT:PRINT
180 FORJ=1TOR:FORK=1TOC:PRINTTAB
    (K*8-8);B(J,K);
190 NEXT:PRINT:NEXT
200 PRINT:PRINT"ANSWERS:-"
210 FORK=1TOC-1:PRINTTAB(8*K-8);
    A(K,C);:NEXT
220 END
240 D=A(L,L)
250 FOR K=1 TO C
260 A(L,K)=A(L,K)/D
270 NEXT:RETURN
290 FORJ=1TOR
300 IF J=LTHENNEXT:RETURN
310 F=A(J,L)
320 FORK=1TOC
330 A(J,K)=A(J,K)-F*A(L,K)
340 NEXT:NEXT:RETURN
```

```
10 INPUT"ENTER NUMBER OF ROWS AND
    COLUMNS",R,C
20 DIM A(R,C),B(R,C),A$(C-1)
30 FORK=1TOC-1
40 INPUT"NAMES FOR COLUMNS",A$(K)
50 NEXT
60 FORJ=1TOR
70 PRINT"ENTER VALUES FOR ROW□";J
80 FORK=1TOC
90 INPUTA(J,K)
100 B(J,K)=A(J,K)
110 NEXT:NEXT
120 FOR L=1TOR
130 PROCdiag
140 PROCcalc
150 NEXT
160 MODEØ
170 FORK=1TOC-1:PRINTAB(1Ø*K-1Ø,1)
    A$(K):NEXT
180 FORJ=1TOR:FORK=1TOC:PRINTTAB
    (K*1Ø-1Ø,1+J);B(J,K)
190 NEXT:NEXT
200 PRINT"ANSWERS:-":@%=&2Ø2ØA
210 FORK=1TOC-1:PRINTTAB(K*1Ø-1Ø,
    3+R);A(K,C):NEXT
220 @%=&9ØA:END
230 DEF PROCdiag
240 D=A(L,L)
250 FORK=1TOC
260 A(L,K)=A(L,K)/D
270 NEXT:ENDPROC
280 DEF PROCcalc
290 FORJ=1TOR
300 IFJ=L□THENNEXT:ENDPROC
310 F=A(J,L)
320 FOR K=1TOC
330 A(J,K)=A(J,K)-F*A(L,K)
340 NEXT:NEXT:ENDPROC
```

```
10 CLS: INPUT"ENTER NUMBER OF ROWS
    AND COLS□";R,C
20 DIM A(R,C),B(R,C),A$(C-1)
30 FORK=1TOC-1
40 INPUT"NAMES FOR COLUMNS□";A$(K)
50 NEXT
60 FORJ=1TOR
70 PRINT"ENTER VALUES FOR ROW□";J
80 FORK=1TOC
90 INPUTA(J,K)
100 B(J,K)=A(J,K)
110 NEXTK,J
120 FORL=1TOR
130 GOSUB23Ø
140 GOSUB28Ø
150 NEXT
160 CLS
170 FORK=1TOC-1:PRINT@4*K-3,A$(K):
    NEXT
180 FORJ=1TOR:FORK=1TOC:PRINT@4*
    K-5+32*J,B(J,K)
190 NEXTK,J
200 PRINT"ANSWERS:-"
210 FORK=1TOC-1:PRINTA$(K);
    "□=□";:PRINTUSING"#####
    #.##";A(K,C):NEXT
220 END
230 D=A(L,L)
250 FORK=1TOC
260 A(L,K)=A(L,K)/D
270 NEXT:RETURN
280 FORJ=1TOR
300 IFJ=L□THENNEXT:RETURN
310 F=A(J,L)
320 FORK=1TOC
330 A(J,K)=A(J,K)-F*A(L,K)
340 NEXT:NEXT:RETURN
```

The first part of the program from Lines 1Ø to 11Ø lets you input the data, the second part from Lines 12Ø to 15Ø calls on two subroutines (or procedures) to work out the answer, and this is then printed out by Lines 16Ø to 22Ø.

The values are put into array A(J,K) a row at a time. In this program J relates to rows and K relates to columns, and you should keep this in mind as you read through the rest of the program. The array A(J,K) is copied into an identical array B(J,K) so the original values can be printed with the answer in Line 18Ø.

The calculation proceeds a row at a time controlled by the loop in Lines 12Ø and 15Ø. First of all, the routine at Lines 23Ø to 27Ø picks out the *diagonal* element in the row (that is A(1,1), A(2,2) etc.) and divides each element in the row by this value. The diagonal element is now equal to one, that is, it has been *normalized.*

The next routine does the major work of the program. It consists of only six lines but it is very difficult to see what's going on. The best way to understand it is to work through a real (but short) example following through each step the computer takes in turn. In essence it works like this. Assume you have so far normalized row one, so L is one. Line 29Ø then takes each row in turn *apart* from row one (Line 3ØØ), so in this case it starts with row 2. Line 31Ø takes the first element in row two, and calls it F. Still on row two, Line 32Ø loops through each column, then Line 33Ø multiplies F by the element in this column in row one and subtracts it from the element in the same column in row two. And so on for rows 3, 4, 5 and 6. Then it starts again with L equal to 2.

At the end of this, the diagonal elements in each row are still equal to one, and all the other elements are zero. So the values for A, B, C etc. can simply be read off from the right-hand column. Except on the Spectrum and Commodores, the program prints out the answer to two decimal places to avoid rounding errors. Without this, an answer of 10 may appear as 9.999998 or 10.000001.

THE CHRISTMAS PRESENTS

The last program will solve any sets of linear simultaneous equations where there are as many equations as unknowns. But many puzzles arrange things so you have too few equations and these cannot be solved in this way—indeed they have no unique solution and the trick is to pick out the most likely answer from the solutions available. Usually there will be some clue in the puzzle to help you do this. Alternatively, the equations may turn out to be non-linear.

Try out this puzzle. At Christmas, Uncle Albert, a somewhat eccentric mathematician, explained to his two young nephews that he would give each boy as many packages as the boy's age (counting only whole years). He also explained that each package contained the same number of envelopes as the boy's age and each envelope contained the same number of pennies as the boy's age. Having handed over the presents he murmured to himself that next year it would cost him £5 more. How old were the boys?

Stripping this of words and putting the boys ages as A and B and the amount of money spent this year as M, the puzzle reduces to these equations:

$A^3+B^3=M \quad (A+1)^3+(B+1)^3=M+500$

These have three unknowns—A, B and M—but only two equations. Without a computer you would have to use a method of trial and error trying out different values of A and B and seeing if the equations agree. The computer works in a similar way, except that it can try out hundreds or thousands of values in a very short time and without making a mistake.

The first step in solving the problem is to look for some extra clue in the puzzle to help you limit the range of values you have to try. Since it refers to 'young' nephews, the boys are unlikely to be over 14, and must surely be at least three. The program to solve the problem is very short:

```
10 FOR A=3 TO 14
20 FOR B=A TO 14
30 LET M=A∧3+B∧3
40 LET N=(A+1)∧3+(B+1)∧3
50 IF ABS (M+500-N)<.01 THEN PRINT
   “A=”;A,“B=”;B
60 NEXT B
70 NEXT A
```

```
10 FOR A=3 TO 14
20 FOR B=A TO 14
30 M=A↑3+B↑3
40 N=(A+1)↑3+(B+1)↑3
50 IF ABS (M+500-N)<.01 THEN PRINT
   “A=”;A,“B=”;B
60 NEXT B
70 NEXT A
```

Running this program will give you only one answer because of the restraints put in Lines 10 and 20. If the puzzle had not specified young nephews, the FOR ... NEXT loop would have to be extended. By the way, the ABS condition in Line 50 avoids problems with rounding errors, what it is really checking for is IF M+500=N.

This method should work for all puzzles with more unknowns than equations. For other puzzles you may need to guess the values of more variables, or possibly have more IF ... THEN conditions of the sort in Line 50. With a large number of unknowns or conditions the program may take many minutes or even hours to RUN, but it will still be solvable.

MATHEMATICAL MAGIC

The third group of puzzles often presented in computer magazines are the number puzzles. These ask seemingly simple arithmetical questions that turn out to be very tricky to solve—unless you use the computer.

Here are a couple of examples.

There is one four digit number which, when reversed and multiplied by an integer, returns to its original value: 8712=4*2178. The problem is to find if there are any other such numbers and if so, what they are?

Another interesting number is 987654321. This is exactly divisible by 17, and it uses all digits from 1 to 9. Find the next highest number with the same properties.

The solutions to both of these types of problem rely on you treating the number as a collection of digits rather than a numeric value. You can either divide the number up into units, tens, hundreds and so on, or treat it as a string and separate the digits using RIGHT$, MID$ and LEFT$ or the Spectrum's equivalent.

The first problem is solved by the first method. Writing the number as ABCD produces this equation:

$$1000{*}A+100{*}B+10{*}C+D=X{*}(1000{*}D+100{*}C+10{*}B+A)$$

and you have to find the five unknowns. As usual, the main difficulty is deciding on the values to try out. A can vary from 1 to 9—it cannot be zero or you would have a three digit number. B and C can vary from 0 to 9. The multiplying factor X must be at least two and cannot be more than 10/D or the second half of the equation will be a five digit number. For the same reason, D cannot be more than 4. So now you can write out the program to solve the problem:

```
10 FOR A=1 TO 9
20 FOR B=0 TO 9
30 FOR C=0 TO 9
40 FOR D=1 TO 4
50 FOR X=2 TO INT (9.9/D)
60 LET J=1000*A+100*B+10*C+D
70 LET K=1000*D+100*C+10*B+A
80 IF X*K=J THEN PRINT J;“=”;
   X;“*”;K
90 NEXT X: NEXT D: NEXT C: NEXT B: NEXT A
```

```
10 FOR A=1 TO 9
20 FOR B=0 TO 9
30 FOR C=0 TO 9
40 FOR D=1 TO 4
50 FOR X=2 TO INT(9.9/D)
60 J=1000*A+100*B+10*C+D
70 K=1000*D+100*C+10*B+A
80 IF X*K=J THEN PRINT J;“=”;
   X;“*”;K
90 NEXT X,D,C,B,A
```

Type in the program, RUN it then sit back and wait—it will take some time to check through all the possible combinations.

The solution to the second puzzle treats the number as a string. The Spectrum program sets a slightly different problem to the others since it can only cope with 8 digits at a time, but the principles are the same.

```
10 LET M=98765432
20 LET M=M-27
30 LET M$=STR$ M
40 LET F=0
50 FOR P=2 TO 9
60 LET P$=STR$ P
70 LET X=0: FOR K=1 TO 8: IF P$=M$(K
   TO K) THEN LET X=K
75 NEXT K
80 IF X=0 THEN LET F=F+1
90 NEXT P: IF F=0 THEN PRINT M$:
   STOP
100 GOTO 20
```

```
10 M=987654321
20 M=M-17
30 M$=STR$(M)
40 F=0
50 FOR P=1 TO 9
60 P$=CHR$(48+P)
65 FOR Z=1 TO LEN(M$)
70 IF MID$(M$,Z,1)=P$ THEN 90
80 NEXT Z:F=F+1
90 NEXT P: IF F=0 THEN PRINT M$:END
100 GOTO 20
```

```
10 M=987654321
20 M=M-17
30 M$=STR$M
40 F=0
50 FOR P=1 TO 9
60 P$=STR$P
70 X=INSTR(M$,P$)
80 IF X=0 F=F+1
90 NEXT: IF F=0 PRINT M$:END
100 GOTO 20
```

```
10 M=987654321
20 M=M-17
30 M$=STR$(M)
40 F=0
50 FOR P=1 TO 9
60 P$=CHR$(48+P)
70 X=INSTR(M$,P$)
80 IF X=0 THEN F=F+1
90 NEXT
100 IF F=0 THEN PRINT M$:END
110 GOTO 20
```

Counting down from 987654321 the program converts every multiple of 17 into a string, M$. Lines 50 and 60 convert every digit from 1 to 9 into a string then Line 70 checks if this is in M$. If any digit is *not* in M$ then a flag F is incremented by one. M$ is only printed out if it contains all the digits from 1 to 9.

TUNE IN TO THE WORLD

Discover how you can use your computer to unravel the coded messages of the airwaves—including the stream of information that is beamed to Earth from satellites

They say that nothing is new. The information revolution didn't start with the coming of the home micro, but when Guglielmo Marconi made his first radio transmissions across the Atlantic way back at the very start of the twentieth century.

The real revolution is not the existence of the technology, but its sophistication—and this is what the information revolution owes to the micro. Today, microelectronic control technology doesn't only regulate the washing cycle in your automatic washing machine, it has also begun to change the face of worldwide radio communications and this is something in which the home user can easily participate.

Short-wave radio has the ability to send and receive signals at extremely long distances, and this is a technology which is widely accessible. In place of the gigantic World War Two radios you used to be able to buy in government surplus stores, tiny compact portable short-wave receivers now have the kind of pulling power that brings in stations from all over the world.

RADIO CODES

When you listen to one of these sets, many of the sounds are those of hundreds of foreign voices. But mixed up with them is a strange collection of blips, blops and chirping sounds. It's these normally unintelligible signals that are potentially the most interesting, because they are actually coded messages. Using a micro, a short wave radio and a special interface you can begin to unscramble some of these codes—a classic data reduction application.

Before looking at how to receive and decode these signals, let's look at their history. Radio telegraphy started in 1901 when Marconi transmitted the first transatlantic radio signal. The technology that he used could not have coped with sending a spoken transmission, so Marconi used a dot-and-dash code (invented earlier by Samuel Morse and named after him) to transmit the letter 's' from Poldhu in Cornwall to Signal Hill in Newfoundland.

Radio technology soon advanced to the point where it was possible to transmit voice signals. But in spite of this, it turned out that there were good reasons for continuing to use codes for many applications. The trouble is that transmitting voices (radio telephony) uses up a lot of radio space, and there is only a very limited amount. Transmitting coded messages by radio telegraphy and radio teletype makes more efficient use of the radio frequencies and over long distances is more reliable. Often, interference makes voice reception very difficult while telegraphic and teletype signals can still be decoded successfully.

Morse was naturally adopted for these uses, but soon people wanted to transmit messages faster and use machines for automatic sending and receiving. The Morse code wasn't very suitable for this because it was difficult for a machine to recognise where one letter ended and the next started. So Morse was replaced by Radio Teletype (RTTY) codes which get over this problem by using a sort of binary word. In the Baudot code, each 'word' is made up of a number of equal length units using seven bits. As these are all the same length, the machine does not need to worry about where they start and stop. Five of the bits are data units representing alphabetic information, and the other two mark the start and stop of each word. Each of the five data units can have a value of 1 or 0—in radio terms this is called a Mark or Space. Looking at the five data units, an 'a' is 00111 and a 'c' is 10001. RTTY signals are normally transmitted at a number of different speeds, these are 45, 50, 57, 75, 100 and 110 Baud (bits per second).

RECEPTION

Such codes are nowadays in continuous use and form a major part of the long distance traffic of the airwaves, carrying everything from messages to foreign embassies, through to satellite transmissions. Between them, all these codes make up the noises you hear on a short wave set. So what do you need to turn them into something readable?

The first thing, besides a microcomputer of course, is a short-wave radio. You don't have to have the latest thing but your set must be able to receive Single Side Band (SSB) signals, SSB signals are a way to make better use of the available frequencies but they can't be received without a special radio.

All of the new generation short-wave sets have this facility. It is also better to get one that will receive a wide range of frequencies, most good short wave radios can receive signals of frequencies between 100 Hz and 30,000 kHz.

The ICOM IC R70 is one of the top range sets. It costs about as much as a BBC B and disk drive, but has almost every facility that a short wave radio fan could wish for. There are a lot of receivers from this price down to around the price of a Commodore 64. A visit to the local amateur radio shop to see what they have to offer or a glance through some of the radio magazines will reveal most of them.

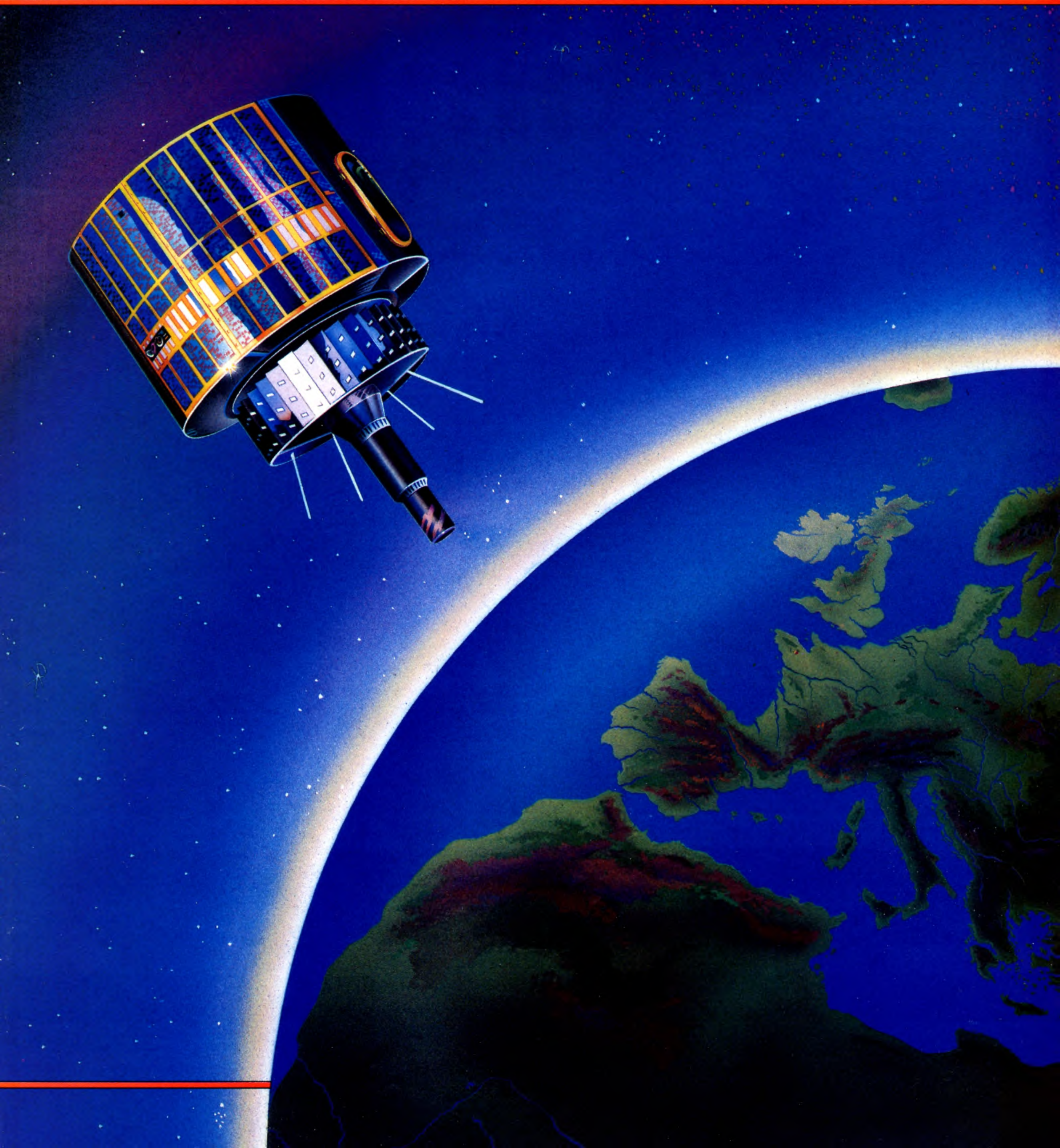
One portable radio is worthwhile considering. The Uniden Communication Receiver 2021 is a full facility shortwave radio and offers good reception on the FM band for ordinary radio programmes as well.

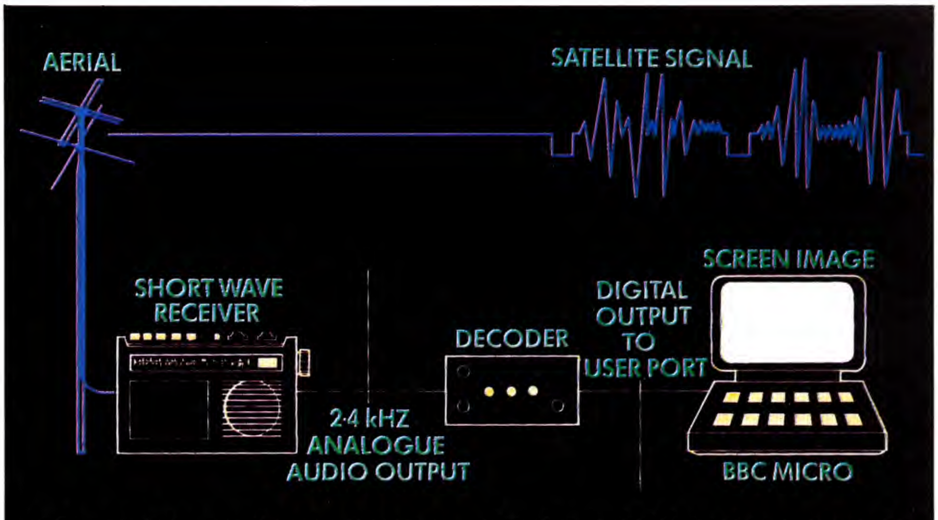
The next thing you need is an aerial. The two portables come with telescopic aerials but these aren't good enough to pull in the really distant stations. The simplest kind is a long piece of wire, the longer the better—either stretched from a convenient window to a nearby support in the garden or hung down the side of the house. The alternative is a purpose built aerial. Many of these are for outdoor use, but you can also get neat indoor ones such as the Datong Active Receiving Antenna.

This is only three metres long but because it uses active components has a great deal of power. There are two models, one for indoors and the other for outdoors, both with their own power supplies.

DECODING

A set-up like this will enable you to receive the signals, but not to decode them. This is the job of the computer, so next comes the all important link between the radio and the computer, the interface. When you tune into an RTTY signal, it is unmistakable, RTTY really sounds like continuous chirping, because the signal is made up of two tones





The receiving station in diagrammatic form. The radio receiver tunes in the signal which is digitized by the decoder and fed to the computer's user port

rapidly switching between each other. The lower tone represents the Space (logic Ø) and the higher tone is the Mark (logic 1). This is called Frequency Shift Keying (FSK). The difference between the two frequencies is usually quite small. On the ordinary AM short wave band (between 150 to 30,000 kHz) the two frequencies are 85 and 170 Hz. On the FM band they are 30 and 220 Hz. The interface operates as a decoder by changing these two tones into zero and five volts to represent the two logic levels of the code and feeds the resulting signals into the user port of the micro.

Radio magazines list a number of small firms making decoders for most of the popular micros and supplying the necessary software to work with them. One system is made by Scarab Systems, whose decoder and software costs about half the price of a Spectrum. It can accommodate the two different RTTY frequencies for HF and FM short wave transmissions. On the front panel are two light emitting diodes labelled Mark and Space, which flicker on and off in rhythm with the signal when it is tuned in properly. When you are tuning in to a signal you turn the knob until the lights start pulsing and then logic signals are being sent to the computer.

At this point, the software takes over. Its job is quite straightforward. All it has to do is detect the start and stop bits of each word and then use a look-up table to find out what character the five data bits represent and then print it on the screen or printer—but it has to do this quite quickly, of course.

It usually takes a little time and trouble to get the system set up properly. A common problem is interference from the computer itself. The first thing is to make sure that the aerial is as far away from the computer as possible. If this doesn't cure the problem then you will have to shield the computer electronically by lining the case with aluminium foil and wrapping all the cables coming out of it in the same material. The foil shield can then be connected to the ground to stop the interference signals from escaping. Another technique is to give the inside of the computer's case a coat of zinc paint and earth it in the same way.

TUNING IN

That's the system—what do you listen to? It can be fun just tuning around the dial randomly, but beware. It is actually *illegal* to receive some transmissions, with a very heavy fine if you are caught. You *are* allowed to receive the amateur transmissions which are found on these bands:

- 3.58 to 3.62 MHz
- 7.Ø35 to 7.Ø45 MHz
- 1Ø.14Ø to 1Ø.15Ø MHz
- 14.Ø8Ø to 14.1ØØ MHz

Books that give comprehensive lists of people using RTTY are the *World Press Services Frequencies Confidential Frequency List* and *Guide to RTTY Frequencies.*

READING OUT

When everything is tuned in and set properly, words suddenly start to appear on the screen, a very exciting moment. If you get gobbledegook instead, it may be because the message is in a secret code, but always try changing the baud rate first.

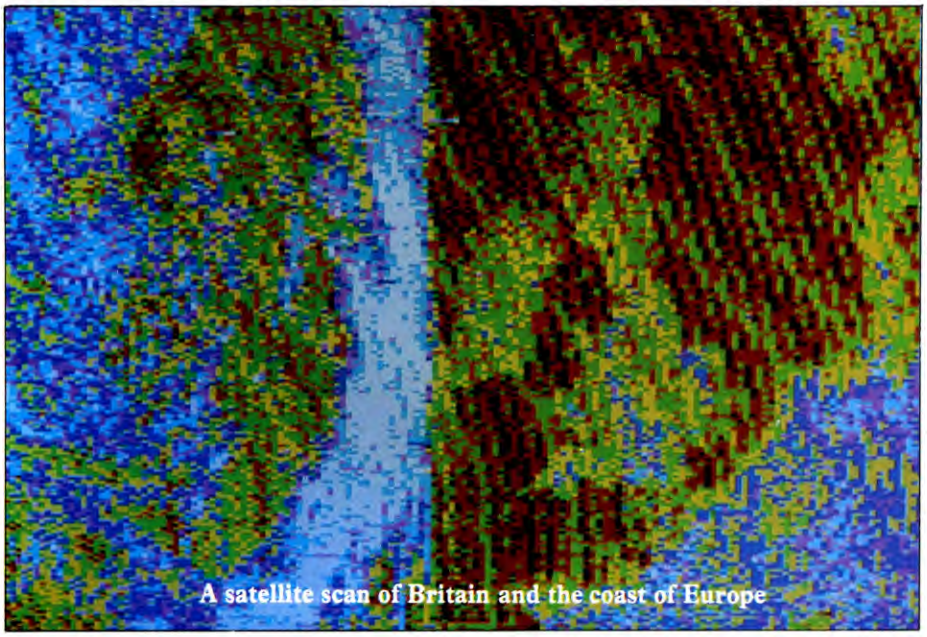
Sometimes, you may not be able to resolve a signal into words at all. This could just be that it's at a non-standard baud rate or frequency shift or it could be that it is on one of the newer Teletype systems. There are two main contenders—TOR and ASCII.

The ASCII is obvious, as this just uses the same transmission system but uses the ASCII code. TOR (Telex Over Radio) is a bit more complicated. There are two types of this—ARQ and FEC—in both of which each data word has a check sum included so that the receiving equipment can tell if an error has crept into the transmission. ARQ stands for Auto Request, meaning that if an error has been detected, the receiving station transmits a signal to the sender asking for the last character to be transmitted. This is quite complex and requires the user to have a transmitter and a receiver. The other system, FEC (Forward Error Correction) is simpler. It transmits everything twice and if the two versions disagree, the receiver only uses the correct character. Software and decoders for both ASCII and TOR systems are beginning to appear on the market.

CAPTURING A SATELLITE

With receivers that work at much higher frequencies it is even possible to receive and display pictures taken by the American NOAA and Russian Meteor satellites—although at present interfaces and software only exist for the BBC B. If you want to receive weather satellite pictures you must get a letter of permission from the Radio Regulatory department of the Home Office.

These satellites continuously transmit television pictures as they orbit the earth. Their orbit almost passes over the North and South poles of the earth and takes about one hundred and two minutes for each revolution. As they go round, the earth rotates steadily below them by about twenty five and a half degrees and so each time the satellite comes around it is looking at another part of the earth's surface. In this way a complete picture of the earth is built up in strips, each one slightly overlapping the last.



A satellite scan of Britain and the coast of Europe

The satellite takes two pictures side by side, one with visible light, the other with infra-red light, using Automatic Picture Transmission equipment (APT). As the satellite spins rapidly, special equipment on board scans the earth's surface and sends a TV picture at 120 lines a minute—one line in half a second. This is a slow scan television picture—a domestic TV scans six hundred and twenty five lines in one twenty-fifth of a second. Each scanned line is split into two, so that one represents the infra-red and the other the visible light picture.

TRANSMISSION

The signals that are sent are transmitted as radio FM radio codes on 137.5 MHz or 137.62 MHz. Receiving these signals is reasonably straightforward. You can use either specially designed shortwave receivers or a converter that connects to an ordinary domestic FM radio. You also need a crossed dipole aerial which can be bought, or made from an old BBC 1 television aerial.

The picture information is carried by a 2.4 kHz audible tone which is modulated by the brightness of the image so when you listen to it it gets louder and softer depending on how bright or dark bits of the picture are. Normally this varying-loudness tone is used to drive a facsimile (FACS) recorder. In these machines a piece of light or heat sensitive paper is wrapped around a rotating drum and the image is exposed or burnt onto it by a scanning light or heating diode.

Receiving facsimile pictures itself isn't new because the BBC used to transmit them in 1934. A kit was available at the time to build a fascimile recorder but they were expensive even then. Today a facsimile recorder can cost many thousand pounds.

ENTER THE COMPUTER . . .

This is where the micro comes in. With suitable software the micro can display a very good picture, although this and the instructions on how to make the necessary interfaces between the receiver and the computer are only available for the BBC micro at the moment. The only disadvantage in this product is the fact that the construction details for the clock and high speed analogue to digital converter are limited to a schematic circuit diagram. You will have to make a hard wired version or design an appropriate circuit board, but it is very cheap. The hardware consists of an accurate clock, and a fast analogue-to-digital converter.

The clock is needed to synchronise the start of each line. Looking at the diagram of the waveform received from the satellite, you will see two large pulses, one at the beginning and the other separating the two pictures. These are synchronising pulses, of which the first is the most important. The software has to align all of these accurately one under the other to produce an undistorted picture.

The synchronizing clock is crystal controlled and produces accurate 2Hz pulses which the software uses to position the lines. It is possible to use the line pulses themselves but the trouble is that if the signal fades the pulses are also lost and the system comes to a grinding halt. It is much easier to record the satellite signals on a stereo cassette or reel to reel tape recorder with the clock pulses on the other track so that they can be passed through the BBC micro at a more leisurely rate after the satellite has passed.

Before the computer can display a picture, the 2.4 kHz analogue signal has to be turned into a digital one. Each line takes only half a second so the information rate is too high for the BBC's internal A to D converter to deal with. An external ADC with a very fast conversion time is used and the signal is passed on to the computer's user port along with the 2 Hz pulses from the other tape track.

THE SOFTWARE

The software presents you with a menu allowing you to select the resolution and colour range of the picture—there are two. The first uses colours that display a good black and white image on a black and white monitor and the second uses a colour range designed to show different temperature bands in the infra red spectrum. A picture is slowly built up on the screen as data is converted.

In its very basic form the software contains two loops. The first produces a horizontal scan pixel by pixel, each time plotting a colour according to the value of the ADC. The second produces a vertical scan synchronized by the crystal 2 Hz clock and line synchronizing pulses. The image-producing routines run in machine code, because BASIC is not fast enough to process the data from the high speed ADC.

Of course, the software does a lot more than this—most importantly it corrects the geometric distortion found in the raw, unprocessed pictures. These look like a tall thin strip, with all the land features being highly squashed. The software plots every other line and stretches the line length to overcome this natural distortion. Other programs in the package allow you to store pictures and view them at will.

As the satellite rotates from north to south and the earth rotates underneath it, it follows that it will only be in range of the receiving aerials two or three times a day and these times will be different every day. Proper prediction tables can be purchased from the NASA information bureau.

See how to apply computer control by programming a micro—in this case a BBC—to run a stepper motor. This is the starting point for all kinds of robotics applications

Following on from the article on pages 1552 to 1556, in which you saw the general principles of using a computer to control a mechanical system, it's time to look at a specific example. The system explored here applies only to the BBC micro, because it is extremely easy to interface these machines with the minimum of electronics. But the principles hold for the other computers, too, although the hardware connections are likely to be more complicated.

The programming given here shows what is involved in controlling a stepper motor. As explained on page 1555, these are particularly suited to computer control, because they can be made to turn in very precise increments. Stepper motors are available with a wide range of torque ratings to suit many load requirements. The simplicity with which they can be interfaced and controlled by computers make them a very attractive proposition in many control applications.

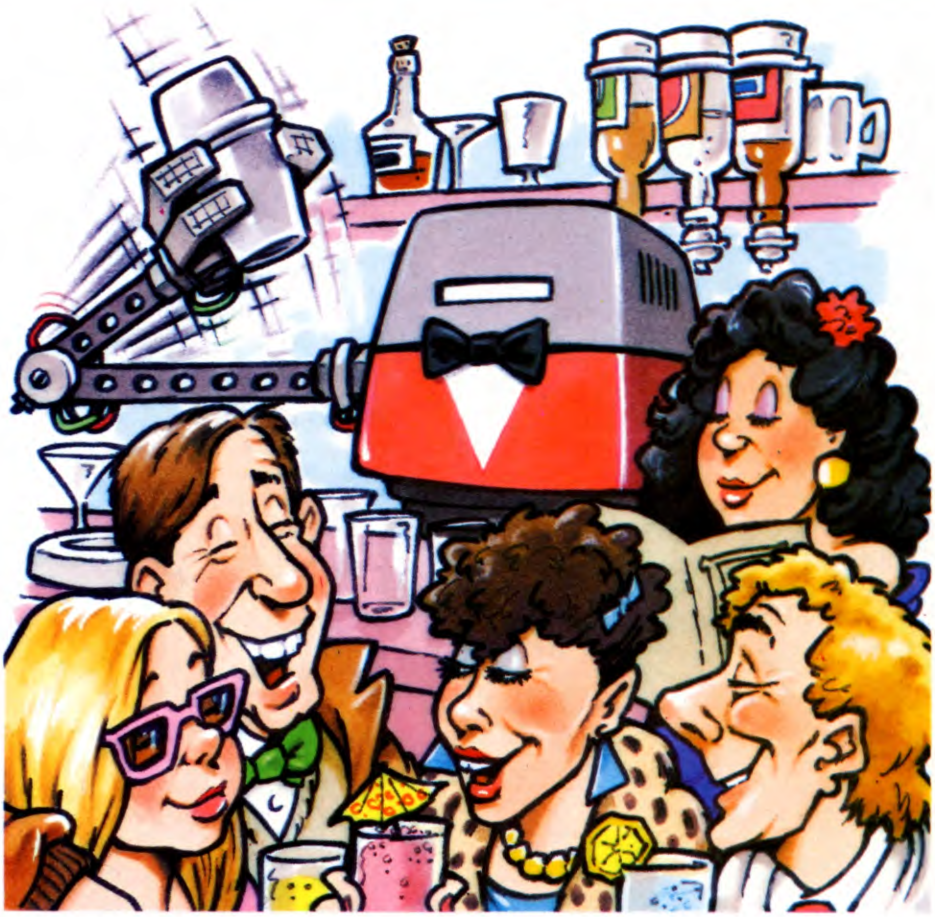
INTERFACING

If you have a stepper motor you would like to drive from a BBC micro, the interfacing is not complicated. The illustration explains the electronics you need to control a stepper motor for the BBC Buggy (there are actually two motors in every buggy).

In this example, two bits from the computer's output at the user port, bits Ø and 1, are used to control four stages of a Darlington driver chip. Two of these stages are driven directly, and the remaining two are driven by the inverse (or *complements*) of bits Ø and 1, obtained by putting them through a pair of inverters. The outputs of the driver chip then directly switch the drives to the stepper motor coils.

PROGRAMMING

This short program shows how little BASIC is needed to drive a four-pole stepper motor. This will drive the motor at a constant speed in one direction:



```
1ØØ ?&FE62=&FF
11Ø REPEAT
12Ø ?&FE6Ø=Ø
13Ø PROCdelay
14Ø ?&FE6Ø=1
15Ø PROCdelay
16Ø ?&FE6Ø=3
17Ø PROCdelay
18Ø ?&FE6Ø=2
19Ø PROCdelay
2ØØ UNTIL Ø
21Ø END
22Ø DEFPROCdelay
23Ø TIME=Ø
24Ø REPEAT
25Ø UNTIL TIME>1Ø
26Ø ENDPROC
```

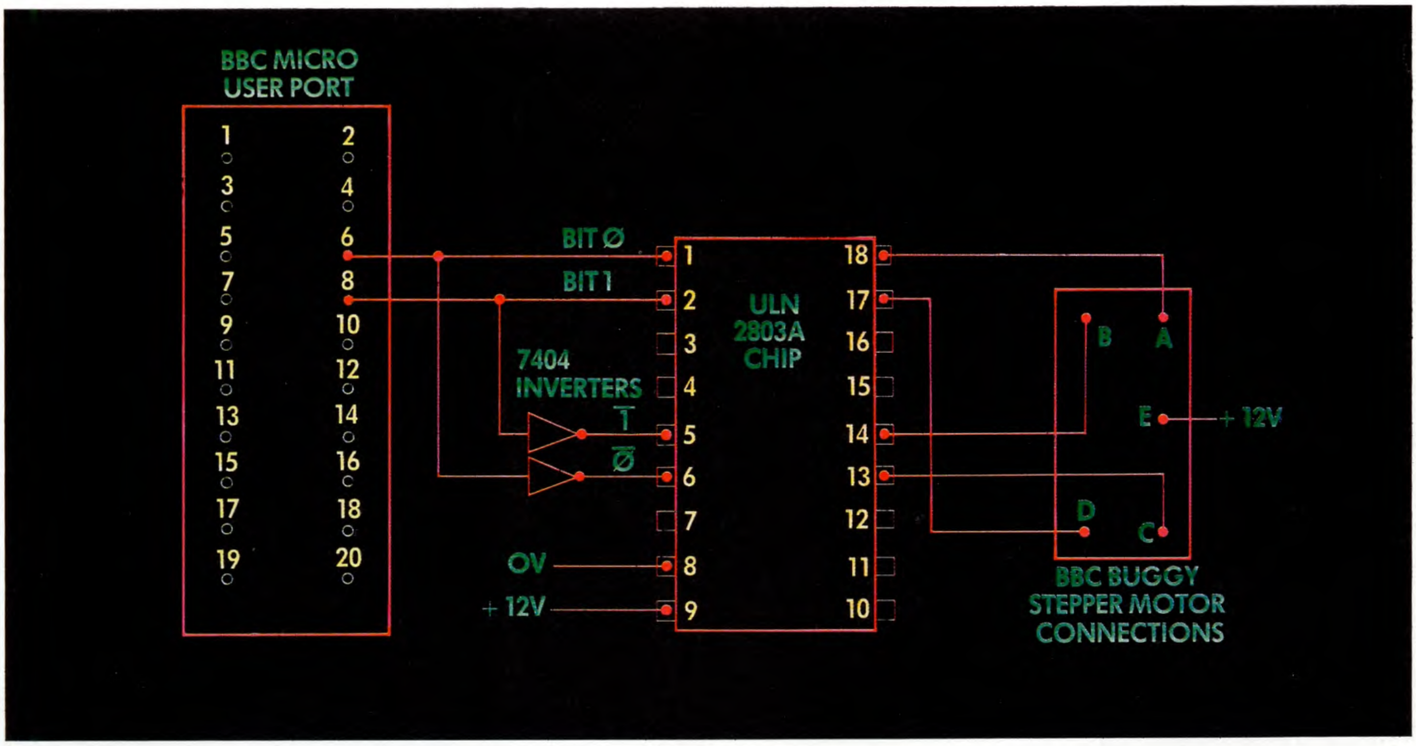
The speed of rotation is decided by the time delay between steps, set in Line 25Ø. If PROCdelay were removed altogether, or if this program were written in machine code you could find the stepper unable to keep up with the driving software. Hardware does take time to react to change and PROCdelay is necessary!

Line 1ØØ sets up the BBC's user port to act as an output. Only bits Ø and 1 are needed to drive the stepper motor. These bits are driven in a sequence (Lines 11Ø to 2ØØ). The sequence—Ø, 1, 3, 2, Ø, 1 and so on—at first sight may look a little odd. What it is doing is counting in *Gray code*. Moving through a Gray code sequence, only one bit in the word changes from one step to the next.

DECIMAL	BIT1	BITØ
Ø	Ø	Ø
1	Ø	1
3	1	1
2	1	Ø
Ø	Ø	Ø

To see why Gray code is used, imagine a very simplified version of a practical four pole

- PRINCIPLES OF CONTROL
- STEPPER MOTORS
- INTERFACING
- ELECTRONIC CONNECTIONS
- PROGRAMMING
- HOW IT WORKS
- TIMING
- COUNTING
- GRAY CODE
- PRECISION



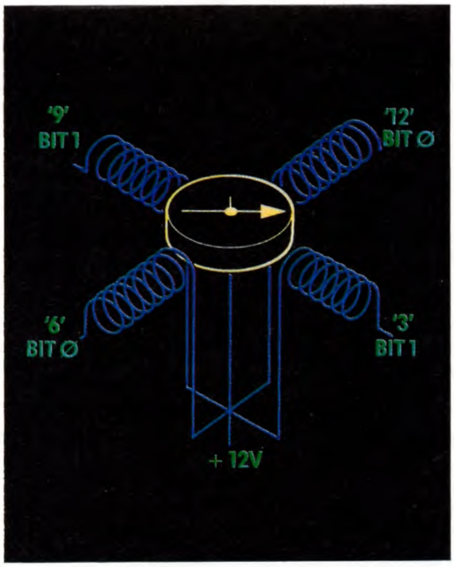
stepper motor—one with only four windings like the one shown on page 1554. These windings are driven from just two bits, bit Ø and 1 of the micro. Bit Ø drives the windings at '12 O'clock' (and '6 O'clock') while bit 1 drives the windings at '3' and '9 O'clock'. The sense of the drive to 12 and 3 is always opposite to that for 6 and 9. The moving rotor of the motor can be imagined as a permanent magnet pivoted about its centre point.

FIELD COILS

As the fields in the coils are switched by the Ø, 1, 3, 2 sequence the rotor always attempts to position itself in line with the strongest field. For example, when coils '9' and '12' are powered (a Gray count of 1), the rotor will move to a point midway between these two coils, where the fields add to be strongest. To step the motor on, the Gray count is increased to 3. In other words, it changes the sense of the field in coil '9' (and '3'). The rotor has to move clockwise, passing through '12' to its new position between '12' and '3'.

By using Gray coding, the direction of rotation of the rotor is never ambiguous. If the motor needs to be driven in the reverse direction the count sequence is simply reversed. (ie. 2, 3, 1, Ø, 2, 3, 1, Ø and so on.)

Fig. 1 (above) shows how to interface a stepper motor to the user port of a BBC computer. Fig. 2 (below) shows how a four-pole motor can be controlled by just two bits of the output



In practice, real stepper motors have many more poles than the simple version used to illustrate the principles of stepper operation. With more poles, the size of step can be reduced, but the principles stay the same. For instance, a 100 pole motor with 1.8 degree steps would still be driven by that same Gray count.

UNDER CONTROL

Once a stepper motor is under computer control, the system can not only be used to control its speed but its position, also. As each single step, forward or backwards, can easily be counted from software the position of a motor can be fixed exactly. Two hundred steps clockwise followed by two hundred steps anticlockwise will return the motor's shaft to its exact starting point. This principle applies as long as an equal number of steps in each direction are made.

ESCAPE: COMPLETE THE ADVENTURE

Here's where the evil king's realm comes together. RUN the game, and find yourself in the bowels of his forbidding castle. Now search for the amulet and Escape. . . .

Add the program lines below to your existing text program and you will have the complete text. RUNning the program will generate the text in array Z or Z%, and the pointers in array A or A%—Dragon and Tandy owners will find two separate programs which will generate the arrays.

If the program shows a checksum error when RUN, rectify the mistake and RUN the program again. Repeat the process until the program RUNs without errors.

```
6652 DATA "596B96D95B2329E57CAD97ØDB6DB6CA62F"
6654 DATA "465E72A64651CBDE464B343Ø16464B24BC"
6656 DATA "CF792CA39739F6A6CF194B232BB164647D"
6658 DATA "31654CAB9632F8CC6472F79ØC591953293"
666Ø DATA "32864B2BE5CE5162CA4CC65ØCAD9598B47"
6662 DATA "23219DE5A65ØCA9978CA39CØ6Ø16464359"
6664 DATA "2D9432F19719EAAB919EAB7C8CAB92C911"
6666 DATA "6465DB3DE4B28B3CE5ØCAD95B232E5B3ED"
6668 DATA "CE4B2A338C965A6632397BC866F9572BE7"
667Ø DATA "65667394B29B3DE55C8C8C8C86DB38C9A3"
6672 DATA "65A32B9ØC5919ØCF9C8CAF94723343ØEB"
6674 DATA "CF7919432F597ØCEF29656CBC3319532CE"
6676 DATA "596F972CAE5165B2E32AE465DB259ØD85F"
6678 DATA "B23219D6464B343Ø2ØCFBCAF96F95F2B3B"
668Ø DATA "656C8C932196B96995B2329B2B94592C21"
6682 DATA "862C8C86DB678CA99194728E55C8CBBCØ4"
6684 DATA "F553CA197592C8CDØC2ØCB64B2EF25947C"
6686 DATA "F2BE4996CA19192CAF91972CAE5167BCDC"
6688 DATA "96465DE55C8CBB6DB3CE54C965AB2ØCECA"
669Ø DATA "B2A65ØCB197C6465364B218B23219BE4C1"
6692 DATA "656CAD9A182ØC8E5AE4B2B65666D97BCAC"
6694 DATA "AB9432B32D9432EB232B656C8C936DB663"
6696 DATA "DB79F6A99197AC964B28656C9651CAB9ØB"
6698 DATA "1976CDB2329E4B296467DB2B94592C86C2"
67ØØ DATA "2C8C862CA197592C8657219E72A64B2349"
67Ø2 DATA "24C96499C64B2D33432ØCB65ØCA99779B3"
67Ø4 DATA "198CA65A64B232E19F19197ØCAE5164317"
67Ø6 DATA "24CE325969951B679CAF95F29E686Ø2Ø55"
67Ø8 DATA "CA6465ØCA394B232865D64B2E191C96452"
671Ø DATA "65EB259632BE5ØCBAC96432397BC8CAB48"
6712 DATA "95B2FØCAE5165B23286464990CBBCA19AD"
6714 DATA "572328E4316464337C8CAD95B34316464E"
6716 DATA "4646432B94590647232965A653CC66F9Ø9"
6728 DATA "4B2965E3232E591C8CA1979CBCB65721CC"
672Ø DATA "8B232198C9654CB9CAF919A18Ø2ØCF39BF"
6722 DATA "532328E5EF2B65C67AAEB2865F191926DF"
6724 DATA "5B2965ØCBCB259267595323286 5E195CA9"
6726 DATA "A394B218B23218B2865D64B2334316467A"
6728 DATA "4646432B9459Ø653232C657CB9C9656CC3"
673Ø DATA "B8CF558CB4CA6CEB2D3232C6432592CB59"
6732 DATA "CE56CA596F95F2328B3AC8CA39572F199E"
6734 DATA "ØC5919ØCDB2864B233432ØCF9C8CAB9772"
6736 DATA "AC8C864B24C8654CA992CB7CA2CAE5168F"
6738 DATA "472328646464B232EDB219779432AE468B"
674Ø DATA "51C862C8C866F9195B2B668616464646DD"
6742 DATA "432B9459Ø65B296465DE5D652CB767BC54"
6744 DATA "96465DE55C8CBB679CA992CB56419EF2F1"
6746 DATA "9652CBE192C862C8C866F9195B2B6686CA"
6748 DATA "2ØCC64B2AE5664B24CA64B2DB24CAE5183"
675Ø DATA "67BC96465DE55C8CBB64B218B23232DFØ7"
6752 DATA "232B656CDØCØ2ØCDB2AE4B2C64646433D2"
6754 DATA "CE5A656CACC96499BE55C8CAECAE51645F"
6756 DATA "B218B23232F395B29652CA8CEB232E72Ø6"
6758 DATA "BE5ØC9643259195D92C93164643319431F"
676Ø DATA "28E5F19473431646464643 2B9459Ø675Ø6"
6762 DATA "953259432F19472DF29654CBC32D9194DB"
6764 DATA "32324DB6C865DE5ØCAB91947218B232195"
6766 DATA "9BE4656CAD9A1816464646432B9459Ø6F8"
6768 DATA "759432ED92C933BCA595B2FØCC64B2AEDE"
677Ø DATA "4651657218B232198C8C8CA3919A182Ø2D"
6772 DATA "CF591919 7ØCB65ØCA99779ØCEB29652C7Ø"
6774 DATA "BE197CB9C964B2595B2324F3ED4F232159"
6776 DATA "98C965AE5B656C8CA19266D97BCB8CA313"
6778 DATA "9572C65F19F6CAE5164B2DB25972C96422"
678Ø DATA "316464653CA197592C8CDØCØ1646433C4E"
6782 DATA "652CA595328B21953219D652CAD94F2BCE"
6784 DATA "E499ØC8C86759432A654C8C96697192CA3"
6786 DATA "B466F9572B65667395F2BE499D652CA736"
6788 DATA "9ØCF792C8C8CA2C594992CB6C965CE4BD9"
679Ø DATA "23232D734316464334CBBCAD97ØCF1965A"
6792 DATA "F919532F2CAE516759432A654CAB91977Ø"
6794 DATA "6CE325969951B37C8CB8CEØ31646433B12"
6796 DATA "CA99194324CDF2BE51C8678CA594B2A6FD"
6798 DATA "516432A6432596D92CB9CDØC38C965A32B"
68ØØ DATA "219532196CB4CB197C65C6631646433BE9"
68Ø2 DATA "CB4CA1978CA2CF194B233ED4DB2392CA3F"
68Ø4 DATA "F96AC59499EF2597AC8653232C64B2EFD9"
68Ø6 DATA "2E72AE5F9919267192CB4CDØCØ38C96595"
68Ø8 DATA "A33DE4B2F59ØC5919ØCF8C8CB8CDØC237C"
681Ø DATA "96D95B28651CBDE6CØ2B92CFB5162CA5FE"
6812 DATA "9499E328654CBE18B2933BCA4C86465C7D"
6814 DATA "36DB6DB3CE5A654CA79191919A1836CA16"
6816 DATA "99194728B219197ØCF8C8CB862CA4CA632"
6818 DATA "4653CB4CA99739A18Ø38C965A67DAF59DB"
682Ø DATA "ØCF392CA997794B29E53CAF972C5919ØBD"
6822 DATA "C865AE5A656C8CA79CØ638C965A33194A9"
6824 DATA "3233ED4D9E72AE5DE4B29B2195B2B3161C"
6826 DATA "46466C34CBBCAD97ØCF196F919532F3957"
6828 DATA "BØ2ØCEF2A6465C67AAEB2328654CBC64FØ"
```

- FINISH THE TEXT DATA
- CHECKSUMS
- DRAGON/TANDY DIFFERENCES
- GENERATE THE POINTERS AND CODED TEXT
- THE DECODER
- PUTTING IT ALL TOGETHER
- LOADING THE PROGRAM
- TAPE OR DISK?
- ORDERING THE PROGRAMS

```
6830 DATA “6499AE4B2EF2AE58CAB94321B6DB6D9Ø91”
6832 DATA “CB6CB6C965ØCA99473433DE433DE4B28BF”
6834 DATA “B39CA594F232AE465DB35CA5953218B242”
6836 DATA “933BCAB97AC86DB67192CB4CDØCØ38C961”
6838 DATA “65A6546472DB2595B2B67AADF29654CBØØ”
684Ø DATA “C33CE4B2A3164655CA2C8E5B64B2B656EA”
6842 DATA “6CAE51CFFØD46Ø219ØCBE5EB2595B3D5EC”
6844 DATA “5B29652CBE32591976CF196F919532F28E”
6846 DATA “DB6DB6DB35C964B2BE4655CA195B25943D”
6848 DATA “72594734D9EF219D64B29658CBE328B359”
685Ø DATA “8C965A6546D9B65ØC964668638C965A62B”
6852 DATA “54CFB5767394B29B3192C8CB195F2E721Ø”
6854 DATA “59532864B25919779A1838C965A33DE4AF”
6856 DATA “B2F5919739F6A6CEF29653643232E3383Ø”
6858 DATA “ØC3BCA595B2FØCC654C965BE5CE6CØ384Ø”
686Ø DATA “C965A33DE4B2F5919739F6A6CEF296533F”
6862 DATA “657259CØ6Ø38C965A67DAA6433EF2597C5”
6864 DATA “C64B232EF38337C964B24CF19499566F1D”
6866 DATA “9191926D9BE55C964338C965A65479BØ5F”
6868 DATA “3DE52CB766B92C8CB86472965EB2323274”
687Ø DATA “32AE5DE5CE5166F9572B65667192CB46Ø1”
6872 DATA “DB2592CBCE57C8CF318Ø38C965A33DE4F4”
6874 DATA “B2F5919739F6A6CEF2965362C8C964993D”
6876 DATA “AE5A658C8CEØ3Ø38C965A65464B25979DD”
6878 DATA “CAF9ØCAE5167591957232ED98C964651E1”
688Ø DATA “CAB978C8C8C8C9668638C965A65464B27E”
6882 DATA “5979CAF9ØCAE5167394B29B219632C6573”
6884 DATA “7CB6CA79432EB25919A18Ø21953219C6F2”
6886 DATA “4B2D19B654C8CB6CA19532324C594995B7”
6888 DATA “C8CB19532328651C936D9C64B2D32A32AC”
689Ø DATA “592CBCE57C879FB5C6725CE75331801612”
6892 DATA “46465CB38C965A335CB4C96433CE55CBA8”
6894 DATA “BC9653CDØC1646432D9432F197ØCF19429”
6896 DATA “B2BE51CB9CFB536432DB2DB2A64658CAD4”
6898 DATA “B9432595D9C64B2D32A64B25979CAF9188”
69ØØ DATA “9A6472F79ØCC65ØCAD95B28B1646433B8D”
69Ø2 DATA “CB4CA19532F194734338C965A33DE4B296”
69Ø4 DATA “F59ØCAE465EB2965F1919262C8C866D956”
69Ø6 DATA “196F95F2328B25926CF19432A65F197337”
69Ø8 DATA “95F28E51CDØC3195F2B656CA19533637A6”
691Ø DATA “CAB9195D98CAF95B2B65ØCA99B3E325958”
6912 DATA “19779F6A2CA64B2965AE6C3E32591977B7”
6914 DATA “9F6A2CEB23259532594B2D736Ø2196D963”
6916 DATA “194F23232C65ØCA99719F6A2CA64B296B2”
6918 DATA “5AE6CØ3E7232BE57CB9CFB516532594BE5”
692Ø DATA “2D736Ø3E7232BE57CB9CFB516759192CAF”
6922 DATA “A992CA596B9B219532D72965A654CB8CØ2”
6924 DATA “D83DE4B2B656CD8Ø239194B232BB3DE4ØC”
6926 DATA “B2B656CF55BE465C66CØ3ACA19197CE4FE”
6928 DATA “657CA792C8CBB67BC9656CAD9B1654C8F2”
693Ø DATA “CA19472D32A65C66CØ3E32592CB1919179”
6932 DATA “9739BØ3BCB4CA19532F19EAA992CA59673”
6934 DATA “B9BØ3BCB4CA19532FØCA64B2965AE6CØ89”
6936 DATA “1652CA994F2D32A6467AAC6464B2D72EØA”
6938 DATA “B232336Ø1652CA994F2D32A6433191922D”
694Ø DATA “CB5CBAC8C8CD8Ø3BCA19532A64653CD836”
6942 DATA “39CAB97792CA79EABBCB4CA19532FØCAE6”
6944 DATA “64B2965AE6CØ3BCA19532F1919739B37C3”
6946 DATA “CA594B2F191972CF792CAD95B36Ø36CAØA”
6948 DATA “1919432B656C8C933DE4B2B656CD8Ø36F2”
695Ø DATA “CA194728E5ØCBBC8CD8Ø38C965A32195D2”
```

```
6952 DATA “3219E72A6465762CA4CC64B232592C8C9Ø”
6954 DATA “B4CAECE325969951B67CE4657CA39198B5”
6956 DATA “C591919B67BC9651678C9651CA196D964C”
6958 DATA “D92CA195323259A18Ø38C965A331943219”
696Ø DATA “33ED4D9E32A655C8CBE18B2321E6CØ3859”
6962 DATA “C965A67DAA6433C652CAB91976CDF2BE39”
6964 DATA “56CACCBE5CE52CB4CBBC9669B6DB671962”
6966 DATA “2CB4678CA5919F6A6CE72BE57C9335CB84”
6968 DATA “867BC8CAD96D9CØ63BCA594B28E4B2E3C5”
697Ø DATA “319DE52CA59472597198CEF286464646Ø4”
6972 DATA “46684210863C652C8CFB536759ØC8E55DC”
6974 DATA “CAD95B2E3318B23233606078CA595F28E9B”
6976 DATA “5CE7DA9B3CE5ØC8CB197ØCE3259699A19D”
6978 DATA “8Ø38C965A33DE4B2F5919739F6A6CEF2ØE”
698Ø DATA “9653643232E338ØC1646433C654CA1969B”
6982 DATA “F9759532AE5E32EF2195CA2CD32DB36Ø21”
6984 DATA “3AC8CAD94B2DD9C64B2D3318B23219BE3A”
6986 DATA “464B2BE432D94B2965F194598C9652CAAØ”
6988 DATA “CC8646432B9197AC964B25919779A18ØA2”
699Ø DATA “2396F9572D736Ø3EF232D72D995C8CD8C8”
6992 DATA “2F9B2F94328E466C38C965A33194323332”
6994 DATA “ED4D9E329316464B24CF79191919A18Ø1C”
6996 DATA “319532596F9739BØCD8Ø2196B96995B2AD”
6998 DATA “329E6C2D9432F19719EAAB919EAB7C8C47”
7ØØØ DATA “AB92C96465DE6C2D943232595F232E73BA”
7ØØ2 DATA “6ØCD8ØCD8ØCD8ØCD8Ø3C654CB4C8CBE3AB”
7ØØ4 DATA “232E591CBE7232AE54C8CD8Ø3ACA594BE7”
7ØØ6 DATA “2F8CD8CD8Ø3E3232E336CD8Ø36C8CA7929”
7ØØ8 DATA “2CA5919BCD8Ø2397BCAB95F2B65E336Ø99”
7Ø1Ø DATA “CD8Ø23919432F3979CD8CD8Ø3ACA9919C8”
7Ø12 DATA “432F19BØ3192CA597C66CØ3ACB4C8CB151”
7Ø14 DATA “9ØC96499E72B652CB7CAF9194736CD8Ø56”
7Ø16 DATA “CD8ØCD8Ø3E7232AE5EB232192C932195FA”
7Ø18 DATA “32A64B2DF28E6C3BCB4CA19532F19473EE”
7Ø2Ø DATA “6Ø3ACA596F9B3E325919779F6A2CD7234F”
7Ø22 DATA “2E736ØCD8ØCD8Ø2992CB6C966CCD8ØCDA9”
7Ø24 DATA “8ØCD8Ø37CAB9195F36CD8Ø2396F94B2AA9”
7Ø26 DATA “65E3363C654CB4C8CBE3232E59DE5A65DC”
7Ø28 DATA “ØCA9978CD83BCA595B2F19BØCD8Ø3CE5CF”
7Ø3Ø DATA “5C8CAB9472F79B3ACA19532EF28655C85E”
7Ø32 DATA “CD8Ø3ACA99572EB2336Ø2396F9572D735D”
7Ø34 DATA “6Ø3E7232AE4B362B9197AC8CB9C9652CØF”
7Ø36 DATA “A99719BØ3E32595B2B66CØ319191963299”
7Ø38 DATA “F8CD8Ø3CE55CBBC9653CD83DE4656CB667”
7Ø4Ø DATA “CD8Ø3BC8CA79BØ165ØCBE323363192CB3E”
7Ø42 DATA “4CA994F36Ø3C654C965B66CØ2D91943264”
7Ø44 DATA “F5919B3BCA59BØ39CA595329E466CØ2334”
7Ø46 DATA “94B2D329E466CØ2F94328E466C37CAF97B”
7Ø48 DATA “47233634CB6CD83C652CB7CB9CD82F9469”
7Ø5Ø DATA “3259BØ3C654CAB9197C66C37CA594B36Ø8”
7Ø52 DATA “2392C965A658CAF9BØ39CD8Ø239B2F9B62”
7Ø54 DATA “37CD8Ø34CD8Ø3C66CØ3CE56CA596F95F87”
7Ø56 DATA “2328E6CØ3ACB4C8CB19ØC96499E72B654C”
7Ø58 DATA “2CB7CAF919473638C965A331943233ED5C”
7Ø6Ø DATA “4D97CA19262C8C879B2597C65ØCB879E89”
7Ø62 DATA “AC8656CACCEF2964657CEØ3Ø16464B2456”
7Ø64 DATA “CF395F2BE56CA2CEB2329E53CAF919A1A5”
7Ø68 DATA 11,253,4Ø8,515
7Ø7Ø DATA 7Ø6,828,998,12Ø6
7Ø72 DATA 1317,1388,145Ø,1534
7Ø74 DATA 1651,1835,189Ø,2Ø18
```

```
7Ø76 DATA 2227,24Ø1,2637,2723
7Ø78 DATA 2856,3Ø23,3136,3265
7Ø8Ø DATA 3365,3577,3654,3831
7Ø82 DATA 4Ø23,4Ø5Ø,4172,4248
7Ø84 DATA 4366,4592,4642,4713
7Ø86 DATA 483Ø,49Ø4,5Ø13,5122
7Ø88 DATA 5171,5247,5312,5394
7Ø9Ø DATA 5435,5484,5544,5628
7Ø92 DATA 5666,5792,5776,5825
7Ø94 DATA 5885,5935,5987,6Ø25
7Ø96 DATA 6Ø89,6138,6178,6255
7Ø98 DATA 6328,6367,6399,6464
71ØØ  DATA 6482,6492,6529,6556
71Ø2 DATA 6589,6614,6628,6669
71Ø4 DATA 67Ø5,6746,6815,6845
71Ø6 DATA 6868,6882,69Ø4,6947
71Ø8 DATA 6989,7Ø18,7Ø52,7Ø86
711Ø DATA 7138,716Ø,7228,7273
7112 DATA 7282,7296,731Ø,7328
7114 DATA 735Ø,7366,7385,7396
7116 DATA 74Ø3,7421,744Ø,7452
7118 DATA 7462,7477,7491,7511
712Ø DATA 753Ø,7539,7561,757Ø
7122 DATA 7585,76Ø2,7613,7655
7124 DATA 7676,7698,7755,7784
7126 DATA 78Ø5,7828,7851,7875
7128 DATA 7923,793Ø,7939,7941
713Ø DATA 7947,7971,7979,7981
7132 DATA 799Ø,8Ø1Ø,8Ø2Ø,8Ø22
7134 DATA 8Ø24,8Ø26,8Ø28,8Ø47
7136 DATA 8Ø54,8Ø56,8Ø61,8Ø63
7138 DATA 8Ø71,8Ø73,8Ø83,8Ø85
714Ø DATA 8Ø93,8Ø95,81Ø3,811Ø
7142 DATA 8129,8131,8133,8135
7144 DATA 8154,8164,8169,8182
7146 DATA 8184,8186,8192,8192
7148 DATA 8196,8198,82Ø4,82Ø6
715Ø DATA 8214,8232,8239,8241
7152 DATA 825Ø,8261,8269,8276
7154 DATA 8282,8295,83Ø2,831Ø
7156 DATA 8318,8325,833Ø,8336
7158 DATA 8344,8351,8358,8362
716Ø DATA 837Ø,8378,8384,839Ø
7162 DATA 8394,84Ø1,84Ø6,8414
7164 DATA 8419,8428,8431,8433
7166 DATA 8435,8438,8441,8444
7168 DATA 8455,8474,8492,8511
```

```
362 DATA ØDØ2792E9333BØ1F7D8F75C26E557ØFØC5BE
364 DATA AEBØF6E1276DC33F7A15E85FAØFD6FBC6D6Ø
366 DATA B86A25883AC448679F89918CCEF255B521D1
368 DATA B342AØEB3B523A75ØØADF2EA97EC9DD36DF8
37Ø DATA 22D441B55F53Ø63D89D88DØ25CB43342AØE2
372 DATA 2AB41EDD6BØABB4222D593CE3542524F32FB
374 DATA C76C366BCD474FB6D8B88DØ2DØE88EA27343
376 DATA A7237859E685AØEC8A75E639E88D9FCØ43AB
378 DATA DC126168C7E5779DE85FAØE7C231A1D161C4
38Ø DATA 9E95AFØD1ØE26EB63786D8B55B75AØ268EEØ
382 DATA D222CØF6AD6B5ØBF9ØØ22C5B9C48E93B47BB
384 DATA 25DF65C3Ø79EAEA2CØDA42A6B732C31F7CCF
```

```
386 DATA ØFBAF19D3D426Ø4D528617237AAØ8FØD3171
388 DATA 9AØ52ADAB3429ØØ22C5B9DØ8DF94EE325719
39Ø DATA CDF4218Ø493EB69577225119781987B39199
392 DATA 981F7E497AØ6A4EØCA2C62EA97118Ø8D5C3A
394 DATA Ø547ØD9FFB953667EØØF83ABDC1C247654B6
396 DATA FEDØ5D4D5F99AD589FB8CBEØ39ØC83DB35B2
398 DATA C31F7ØAAE72368B5CEBØFEDØ5D4D5C85Ø1ED
4ØØ DATA DAD826F2EØ1F76A94ØØE4E953391FBD33BAB
4Ø2 DATA DC85Ø1F834A4C6F9B9BB9ØE3DDC8B4ØE9Ø8E
4Ø4 DATA 4472C3ØF83ABDC1ED44F711C654E8EC384EØ
4Ø6 DATA 5C29C4679EE42D9998ØF83ABDC1EDØEC493D
4Ø8 DATA 656E8FØ9245CC68Ø36DD8225E85FAØFC87ØD
41Ø DATA 5ØA6E3E9D2F7F7E4A56ØBCFAD3Ø5E16A3587
412 DATA ACB577C3EB8EØCFAF58A16CA172Ø2Ø699D22
414 DATA B42F9Ø3E9294A2EØC1F4AB4D69BF9F4F25Ø4
416 DATA 1A6EØ5A2EF795ACE1BDCBEA4B1FAC11524E4
418 DATA 32E42ØD6B42F9Ø3CBBB68FC5653B3AC475Ø9
42Ø DATA D2A6EF84251D67184Ø5F9Ø3D684ØBDEE91F7
422 DATA 14A897Ø98E5AAE72C337DA37DØ75531E9C65
424 DATA B4ØE3D2ØCF1ØFE23E7DB5C547569CFD97439
426 DATA 985C735E413D9698217FB859BF6A5ØCFBF82
428 DATA ØCCF4Ø2235AØ9F783Ø2933EDAD49CFC3A9EF
43Ø DATA 549EF26FB87DF4A34ØCC8C352117381BBF82
432 DATA Ø6A3684ECA721A17B859E7DE21F8D4EDD39C
434 DATA 79679C3452C3737F4Ø37DA3BCØCE39BEB5EC
436 DATA CDC156679E6BB4ED5C3246E4BCE54Ø1F7BAØ
438 DATA 4679744AAEØ3755C5714ØØEØ36C9948DE85F
44Ø DATA BD8E5A8A6652AA75759486A37BAE21EF34CC
442 DATA 8C37DA5Ø4764EB662EF12786655Ø22BØF1D8
444 DATA DACD31CE7ØD3EØ1FBBF9D6CEDB49DFØ76DF7
446 DATA 5CA617A9D72A8B2728E51B3CØFA4AC7BC5EF
448 DATA 34A25AØ74DØBB859E7DC87CB26EFØC5E1547
45Ø DATA A4761A17B859BF6A51CDFDCC7728381BE7ØØ
452 DATA ØØ3A15BBB313DBF21B37DA3EEBDØ4E7C4DF6
454 DATA A9B92CDFBØ37DA665DØ676BD91DDFØ656926
456 DATA 624B2BDB93DD287BAD27735FBDCAF7A9B87Ø
458 DATA 179FABDEB17A56D5FC2C9C961B98662FB859
46Ø DATA BF6A51CDFDCC7728342375D2866F9837DA5Ø
462 DATA 497838EA97DØD3B6Ø4453DFFD9AØØØ37DA5Ø
466 DATA 497838EA97CB26EØØC5EE9C22465Ø62ØC1EF
468 DATA 347ADØD99412D274B141914EØ5A49C961B8E
47Ø DATA BF7BE72DF3A818BF9Ø717859BD73C3F2DED3
472 DATA 73429Ø3AC3C68FC32F164CA561B5543ØD82A
474 DATA 3E151471DE6DE69AA3EBF8842D15Ø1F62428
476 DATA 6238217FB859BF6A5ØE9FA964Ø9987BØ3DC7
478 DATA 3926621378E5718DE85F45ECE852B6DD25DD
48Ø DATA AF6AC2983ØF7C7787716C52CF35F7846F9F6
482 DATA 97CF28A42E6BØ6C343ØF27187716C52CF35F
484 DATA 792EFE726E2DØA5966BF792EFE726E2E215Ø
486 DATA CB5B5884B59514E2B6F52D6B5888A2EE5AB4
488 DATA 33D6F38B59E987C8F9A237ECDAD83Ø53A11C
49Ø DATA A9C5ECF7CBØØ4E6B6F47D3C5FADØCB5B58EE
492 DATA C85243A116B83Ø5124F3219D38153DØ5EC51
494 DATA 24F32ØEFAD798CD76E13213B58E53DA867AE
496 DATA FA411E1C94B44D7F6E137Ø721BDDD2EFF2FC
498 DATA 2B5AB6D985ECE13EEAD54D7F5A1ØØC3AB35F
5ØØ DATA B859B84FFE67DDA8BC2ØA34A3E15147EA5E1
5Ø2 DATA E63399AFFED3DF84986C328485999837DA3B
5Ø4 DATA CØCE39BE94A87ØCFE6BFB859E7DC87E1D53A
5Ø6 DATA FBDD56766659F6B2CB8423FØA33D66F83DD2
5Ø8 DATA 57866CB559AFBB9322B74C7C4A5ØDC65BEDF
51Ø DATA 81999Ø83BØ3BC8CE39BDØ7195ADBB4183Ø7D
512 DATA A577739EF37DEØ62Ø7C15335B859BF6A51CD
514 DATA FDCC7728381BE7ØØØØØFBF13Ø6FCD2BDBB1C
516 DATA E2F35B35BA2CADF29AØ638ADBA8C83D5DØ4D
518 DATA 5296445D94AB6ØØD27A311DDEC8D75D6B6FA
52Ø DATA B5D854CCØØBDECBBA2654Ø37DA3BCØCE39BE
522 DATA 9267C9FA833545157CE54Ø783Ø853A571E6B
524 DATA 343BF19D35325771291C66BF34Ø4F39B5878
526 DATA 3Ø783Ø783Ø783ØFØCDØEØ3969F913432EØE9
528 DATA D2F74D7FF9AF78374D7FF9AF5899119AF9AF
53Ø DATA ØFB95EDDB35FF9AFØ9ØEBE72EØ783ØE9Ø43B
532 DATA E6BF4296466B6B428391FE6CADEDC735F9AF
534 DATA F9AFF9AF792A681B1384A8B6ØE6B6F47D3C3
536 DATA F35F6A5A66BF7846F9F697A84E6BF9AFF9AF
538 DATA 22D8E6BFF9AFF9AFF9AF5EA4735FF9AFØEF1
54Ø DATA 278B58FØCDØEØ3965E8F278B58EDD6FØ1B78
542 DATA 3ØF2D2BØ779A6A136E14B35F6A9468CCØØ8D
544 DATA 75D6B6F8AAB2EØA868E4CA9BE6BF7855366B
546 DATA 4Ø626335735D54F2EØF5DB6B58EB1E6BDØF7
548 DATA B35F42D113CCØØFØC5B5ECBØ8F45ECEDD9AF
55Ø DATA 6653199AØCB3199A3C21E6BF5EFØB35F536B
552 DATA 58FØADF84D7F3C15ECFØCA9E99AF5E514D7F
554 DATA ØA59E2F83ØE6B68DECBDECDEB6D2B6FØ1BF2
556 DATA E96EF2383ØEAC3Ø2127DED2C6E46B637DA3B
558 DATA CØCE39BA413399AFA5FØ8E1DFAB7DACE3916
56Ø DATA 4EØ29Ø92737A5A2E2272399997FF7FFFXXXX
999 X=1:DIM A%(2Ø4),Z%(2315):PRINT " PLEASE
    WAIT..."
1ØØØ READ Z$:FOR Z=Ø TO 35 STEP 4
1Ø1Ø ZZ$=MID$(Z$,Z+1,4):FOR ZZ=1 TO 4:Z(ZZ)=
    ASC(MID$(ZZ$,ZZ,Z))-48
1Ø2Ø IF Z(ZZ)>9 THEN Z(ZZ)=Z(ZZ)-7
1Ø22 NEXT ZZ
1Ø24 IF S=Ø THEN A%(X)=(Z(1)*16+Z(2))*256+
    (Z(3)*16+Z(4))-32767:T=T+A%(X)
1Ø26 IF S=1 THEN Z%(X)=(Z(1)*16+Z(2))*256+(Z(3)*
    16+Z(4))-32767:T=T+Z%(X)
1Ø3Ø IF S=Ø AND X=2Ø4 THEN PRINT1:X=Ø:S=1:
    TT=T:T=Ø:IF TT<>-391783 THEN 1Ø6Ø
1Ø32 IF S=1 AND X=54Ø THEN PRINT2:TT=T:T=Ø:IF
    TT<>86611Ø THEN1Ø7Ø
1Ø34 IF S=1 AND X=1Ø8Ø THEN PRINT3:TT=T:T=Ø:
    IF TT<>-23866 THEN 1Ø72
1Ø36 IF S=1 AND X=162Ø THEN PRINT4:TT=T:T=Ø:
    IF TT<>3ØØ681 THEN 1Ø74
1Ø37 IF S=1 AND X=216Ø THEN PRINT5:TT=T:T=Ø:
    IF TT<>248654 THEN 1Ø76
1Ø38 IF(S=1 AND X=2315)AND T<>3256Ø3 THEN
    PRINT6:GOTO 1Ø78
1Ø4Ø IF S=1 AND X=2315 THEN 2ØØØ
1Ø5Ø X=X+1:NEXT Z:GOTO 1ØØØ
1Ø6Ø PRINT"CHECK LINES Ø-44":END
1Ø7Ø PRINT"CHECK LINES 44-164":END
1Ø72 PRINT"CHECK LINES 164-284":END
1Ø74 PRINT"CHECK LINES 284-4Ø4":END
1Ø76 PRINT"CHECK LINES 4Ø4-524":END
1Ø78 PRINT"CHECK LINES 524-56Ø":END
2ØØØ Z$="":FOR Z=1 TO 255:Z$=Z$+" ":
    NEXT Z
2Ø1Ø SYS 52976:FOR L=1 TO 2Ø4:Z%(Ø)=A%(L):SYS
    53ØØ8:PRINT "MESSAGE";L:PRINT Z$
2Ø2Ø NEXT L:PRINT "SAVE DATA (Y/N)?"
2Ø3Ø GET A$:IF A$<>"Y" AND A$<>"N" THEN
    2Ø3Ø
2Ø4Ø IF A$="Y" THEN INPUT "NAME";N$:GOTO 2Ø6Ø
2Ø5Ø GOTO 2ØØØ
2Ø6Ø OPEN 1,8,1,N$+",S,W"
2Ø7Ø PRINT#1,2Ø4:FORL=1TO2Ø4:PRINT#1,MID$
    (STR$(A%(L)),2+(A%(L)<Ø)):NEXT L
2Ø8Ø PRINT#1,2315:FORL=1TO2315:PRINT#1,MID$
    (STR$(Z%(L),2+(Z%(L)<Ø)):NEXT L
2Ø9Ø CLOSE 1:END
```

```
259Ø DATA 2447849786665AF86D48351Ø2A25BE9E
26ØØ DATA 58DAC53A6Ø6816F85Ø385A3837Ø34D7Ø
261Ø DATA 477E47BC23EDEØ24ED6DDBB6991DC6Ø8
262Ø DATA ØAØ72ØA648A5C4ØF6F475AB8B76D4CA9
263Ø DATA EØ22F81AA7DØØEØE228963664C2B24EE
264Ø DATA 87D82942FC96F8885A3886E61A755Ø38
265Ø DATA 787CADØ7Ø79AE4BØDBB61D693Ø46686F
266Ø DATA ØØ31CDEC257E5Ø387DC8452A7C8D634A
267Ø DATA 38ØCCD117Ø5Ø385AØ82FØ34D498EFC8E
268Ø DATA 6D47DAC111DADBB64C333B8CA4D58E83
269Ø DATA B442E22CF59E22C4DF12BFØ8ØØØDEØ96
27ØØ DATA 257E5Ø38D2C2452A634A7D3BCD117C8D
271Ø DATA 385A38ØCØ71A755Ø4CB3AØAC3DA9F2DØ
272Ø DATA 86DB2BØ84C6CB881FD7ØØA31F28BBEA8
273Ø DATA BA22F4EFF776A49CACD81BAA1FCØ4564
274Ø DATA A5B117B52CD6FDØ463696BF6ACFB31E3
```

TROUBLESHOOTER

- Escape is rather a complicated program, consisting of three pieces of software, and for it to RUN properly, all three parts should be bug-free.
- Debugging the compressed text is easy because of the checksums, but you will have to be more careful with the BASIC control program and the text decoder if you are entering it from scratch. Inaccuracies in either will make the program crash, or messages appear in the wrong place or point during the game.
- Commodore owners should make sure that the text compression/decoding software is located in the correct part of memory. The listings given in the three text compressor articles—starting on pages 628, 648 and 684—are for the Commodore 64.
- If the BBC runs out of memory it is because there are too many spaces in the control program. The solution is easy—just use the packer starting on page 546.
- If you are using tape, you should make sure that the separate pieces of software are in the correct order on the tape.

```
2750 DATA 2195FE5D4326B9899A415554560AB1E8
2760 DATA D6FDØØØE8BCEB62D73241E5AAØ4951Ø7
2770 DATA 34ØBCA7A148E16CEE6B76D1B7E3BB2F7
2780 DATA 171B878384DØ8CDB5A388Ø211C4AE938
2790 DATA 14ØEEØ8ØCBC1871A6AF818EA31A5854F
2800 DATA 3Ø345B7BØ785Ø31ØBEF99DCCBØ5FBCB5
2810 DATA 523783335AF8247EE153D136A17ØB4FØ
2820 DATA 1912CF9853EA57F23D8C79BEDCA177CB
2830 DATA FØØ9682D9CA17ØB44BD38118B64D7211
2840 DATA 2DD457FØ4E951F4EAFF9E1ABEAED686A
2850 DATA 2EE6EØAB387ØDBØØC731E3DC6A719FD9
2860 DATA D21Ø7D52CE96CE6BDC931ØØ37Ø53E68B
2870 DATA BD87837E15839F62F53C648DF172AC68
2880 DATA B12E4FB9B4D42E72D7723FED6ØCØ19AA
2890 DATA CD4B8E3D8D16BEC8D6CA8EØ3A5A74E72
2900 DATA 255BF51A3A8966C693FBA4A67B387Ø52
2910 DATA 4D7293425822C4Ø35AA27A6FED54F44D
2920 DATA 4E7C27F83D3Ø34EBC8CD4B8EØ38D16BE
2930 DATA 72D6CA8E62A4A74E71C956BDEFA9AE9E
2940 DATA D6C2595BA385EF9637ØF7E5277387Ø82
2950 DATA F7FØFAE6DFA425AA8EA6F69A6B2143F3
2960 DATA 385FE43EF3A3853387DØD97B5Ø931A8E
2970 DATA 81861CF8C65B2AEFED34187Ø27210667
2980 DATA 3EE778E15D22D4E629F749A937697D78
2990 DATA BDF9D1C2DAB5FØEC1D6EEFC4EC2334ØF
3000 DATA 1Ø8ØA159CEA3F69391F8CD7D57AA8176
3010 DATA 3E8E191FB66DDBC6E6571Ø8A16E2492D
3020 DATA D2FB4ØB3EØ1CF114433B1Ø3Ø53EB19C7
3030 DATA 912A773DFCA2F6DB7E39D6C24CØB1F55
3040 DATA DE631FC753E6B3B6F8ØØEA7EØ38D3A5A
3050 DATA 34132FF99ØFD2Ø3Ø56EEC3F5BF45F1FØ
3060 DATA E276B12E4Ø436EA76Ø6816FABDEFFE2Ø
3070 DATA 6B3861EDC5BA89A56D1B68C8F86DDBB6
3080 DATA EFD418A191ØE673DEC2Ø189A76BA53BB
3090 DATA EB72AE8ØD41DED17D5A2F9ED54DFB6C1
3100 DATA B9Ø93E86B5DCØ3ØDE32Ø43B3DE9EB5AA
3110 DATA 433BØBEBCF13D6A25ØD243A5DB4E8DB2
3120 DATA 919AD86C72B14E9FD2A1Ø61A88E6471D
3130 DATA B4FØ494EED2ØØCCD3A6676ØAC11F8E68
3140 DATA 135CACC3E64769E16Ø689EF73242E82Ø
3150 DATA C5E1D221ØE2F961ED67DDC9Ø16EBFØF6
3160 DATA 272Ø76DB2352E1ØE6C2DF74ØØ31ØCØDØ
3170 DATA 491C5CACBCC73C69C4E5EØA5A32E9F87
3180 DATA A7C2DB4Ø2Ø433337BB8FDØFC43BD9E71
3190 DATA 87D24EEØA1FA249772B1ØEØFD2A1Ø61A
3200 DATA Ø31Ø4333ØC1D5CAC336E955FF54D1AD7
3210 DATA 3FC981A123F796361AF81AD19A11B4Ø7
3220 DATA 4AFE2Ø18E124Ø7FAEBE22D4A8EØØ1217
3230 DATA 48853BDC967BAØ8D1Ø6Ø68B61D5CACØ3
3240 DATA B7D477A44EDDD17E5C2D9ADFE14BB91F
3250 DATA DCØ3ØDB92Ø43B3B52467ABFØB14EB6E8
3260 DATA 4EDDD17EEE8186DC8ØA1D95ACØAAF62Ø
3270 DATA B396CEØFBBD47B9281865CAC46A5B4F9
3280 DATA 1ØBC39FA34C95DE4C48F1ØØFØ31Ø4373
3290 DATA 541F5CACE51DF15ØØ4C4ØE4F442ADCE1
3300 DATA ADE51E68Ø31Ø189A5Ø1F5CACE53EC9ED
3310 DATA A4ØAØF6FB681465D6826Ø2DEØ7FD2Ø6Ø
3320 DATA 63A7DØØE77F852EA3C6125E561Ø653FA
3330 DATA 2C88B56B6BC4F7B675FA8C8F97CB968B
3340 DATA 1D6AAØ211Ø3Ø34232295123F2BF541E1
3350 DATA 1FCØE94E9AØ5A55ØDB46ØB67E6B3B6DE
```

```
3360 DATA 4CA94FF7A945D07E68B96C39100CCD35
3370 DATA C3ED4E332E4FB9F194427DB169E1DBA9
3380 DATA 70C6B947E83E1018C47B3341452A257E
3390 DATA CB9683C238C0D09C7650385A661C1FD3
3400 DATA 21BD0F34247E114F55DCDC67DA4F6AF5
3410 DATA 5D183AF43EC15FF380A199166B3F5A38
3420 DATA 833FD0D023C0D08C285BD996E6BD1C65
3430 DATA EEB32AC0C44F4A2D9FD4F0297E387072
3440 DATA CDC0A47418A1360C833F1CB84FE8A486
3450 DATA 189A734ADF675A38EE54F9A168F99D53
3460 DATA C4D2351C38C080F3CF403C5AED35BFB9
3470 DATA 68D6C24DEE346C1EE5C633DC20C0E63C
3480 DATA 7AC6A1FB7A8242F4C08AEBAE0A65DB70
3490 DATA 3D303447E68BDA8FF5762A53FBA40695
3500 DATA B4F0A1AF5A380CCD6B65C751A7F2AE67
3510 DATA A251E5875AD971B1F0CFB1CE3B2060D4
3520 DATA 5BCF56FAED085F4A97A7DCF80B2B57AA
3530 DATA 9BE6A828D2C79E6DF8223E9604ED515A
3540 DATA 5A388666CC07DD675F8CF0A677244895
3550 DATA 5A38189ACED16B3F29F7CD7D01671CB8
3560 DATA F056EE8039BF45F13F5A38B07DCED16B
3570 DATA BA29F7CD3830E0AC07DD675ADE11BEF6
3580 DATA 27E96670DCAB4CE27CA8DE1360F3282D
3590 DATA AA77CB3DA09771387B31DF2B2D7CD6D6
3600 DATA 8F9B971C5A38C63CCED16B3F29F7CD7D
3610 DATA D3F524B43803CE0379C9515AD117EBB8
3620 DATA 4684B753A15900BE515A3880EBB879C9
3630 DATA E1A6CC17C3695F8C218666A47BB4F041
3640 DATA 1314DA5042317552A5854F1125B76D1B
3650 DATA DFCFE3E6EA9CCB191060CCF33D5AF872
3660 DATA 5EF343741043F3D40FC7433BCC17AFC4
3670 DATA D4B6E1A6BE2B58315E7294167D94666E
3680 DATA A2B5108F85F43ED03034473C6B3F5A38
3690 DATA 977AEAD0B1079AC027B9C8BDE6F814E2
3700 DATA 60688EF15368EDC5DEA5DE3699426B2F
3710 DATA 7947F8B02D4517F747F86073D017F779
3720 DATA 6CAEA5A810C3C48617F719A760732D45
3730 DATA 737E2FF95A8A2EEE2FF9C0E62FEE737E
3740 DATA 5C4B51A1963585D8F636E39489D86CAD
3750 DATA B5DAEF228C73D7B3C907EAD8EDB7A379
3760 DATA 54B0D95AC6291D21014BF86C48EF6CCE
3770 DATA D17AC653EFD85C4BA2C3534852B0B996
3780 DATA 9EA1F4A406BD16B8F4A4526C7A2DF0A0
3790 DATA 14EED80CE6D83CA1AFE7A9BD1D9E427A
3800 DATA 80CDB51473F014EEF052DE9B5BABFD72
3810 DATA ED05DA36D66A3F6111DA80CD60333B8C
3820 DATA 50385A38A95D687E4B23213C7F9416BE
3830 DATA 3666E2257E9B07DA18855FD40585B26D
3840 DATA 5A38189AB9CF403CF0A914BF38D83CD0
3850 DATA 07DD675A7B3B55E2E677D6DE4DB3765A
3860 DATA B66DDBB63F5A38DB6FEE3312783DDD9B
3870 DATA 9C6DCB7EA2943B0651CA7DB801E03E5C
3880 DATA 3C86109ABEB9CF48DCDA1A877E360F34
3890 DATA 9FF3782563607E7336D3C2876B3F5A38
3900 DATA CD7DCED11CB829F710800167FD86143F
3910 DATA 1D3BBE5234DBF4622D2D3A3081460BDF
3920 DATA A38E2BEE1394F54016F1A5B4E0AC14DE
3930 DATA 91A4A70E8E6068DEFB36D7F5CDD4D935
3940 DATA BC6CBE8038C06622CF403C5A6812BFB9
3950 DATA 3403FB49FC16C530B079C0E69E58BA86
3960 DATA 713CB46CD733B59EE61DA972730 5B4C0
```

```
3970 DATA BØ79D89CBØ79BØ794DF1BØ791F9783ØF
3980 DATA 6Ø33B492CDF852EAF8BØ798Ø798ØCD38
3990 DATA 919AD8BØ8FBØ799B33DEDEBA89BØ796Ø
4000 DATA 6Ø733EØF84EABØ79C2CØ663CEB6CC697
4Ø1Ø DATA 7E92Ø34347EE2D6D79BØ7936F9BØ79BØ
4Ø2Ø DATA 931CE82B8EB728855348EF6CEA6Ø73C4
4Ø3Ø DATA F8CØE65B17F77947796CCEA9A2BØ79BØ
4Ø4Ø DATA 79CØ66D979BØ79BØF3A5DEBØ8EBØ796Ø
4Ø5Ø DATA D88CA7F283ØF4DF1A79ØDE9756EED88C
4Ø6Ø DATA BØ799BF1F7B152F3EE14EA9BEA6Ø3315
4Ø7Ø DATA 8ØCDE89536D7F58E6ØB32AF94AE5E8A9
4Ø8Ø DATA F8CØ669CCØ6CB656F336E3636ØF3D45E
4Ø9Ø DATA D86C5BF65Ø6C9EECC26Ø33F88ØCD93D2
41ØØ DATA 6CB645F16C46ØFB1E6BØ59EE8C9B9954
411Ø DATA BC9B99B4DECØ6622D36Ø33F12DF1D86C
412Ø DATA BC8ØCDF94AF16C16DEBØ199F8A8ØCD52
413Ø DATA BØF9625A6C8E36E736DF6CBE9BF136D3
414Ø DATA 726F69F343EBBØ396D7E92Ø33647EE2D
415Ø DATA 4Ø3C5A38C1BBB9CF2536ØF347A1EØEF1
416Ø DATA B9CF5AB86ØCØ79177BF3931Ø73A22FDA
417Ø DATA 79189AB9
```

```
27ØØ DATA 6DC491771215A62114EC43C45F,1421
271Ø DATA 896FCE686ØE25A385Ø751AØ7AD,1429
272Ø DATA 7C78BØE49AØ6DBB48EDEDØ8C61,2Ø16
273Ø DATA D99A68Ø1C283F129522DB9ØFA9,1579
274Ø DATA 4C71AF8239A18ØE25A385Ø7Ø4D,1481
275Ø DATA Ø3BC223BF23927Ø6DBB48EDEDØ,1599
276Ø DATA 8C61D99A7Ø71DAB4859B6DB6DB,2Ø29
277Ø DATA 89ØAD31Ø8A7BD422FC4B7E5CØ1,1427
278Ø DATA AØØ6E141F894A917ØB48EDF529,165Ø
279Ø DATA 8E35FØ47343ØE25A385Ø751AØ7,12Ø8
28ØØ DATA ACAØB34DB6DB6DB7A1E6549E84,2Ø46
281Ø DATA 15EDC34ØE1B13ØC429C3F7Ø8F4,1898
282Ø DATA 5F977FA115D4E668EDEF543D8A,186Ø
283Ø DATA C6445CØ1FB845EC69413F66B69,1659
284Ø DATA 63E331FB516DB77FA9D2189B92,183Ø
285Ø DATA 7ØD5155Ø7AE8B1ØA56ØEØØFDA8,1488
286Ø DATA BE8B5A1E2473Ø75272681EB282,1245
287Ø DATA CD36F8968E141F9BDEC8EDFAØE,1928
288Ø DATA 1C6C5F6E3343E25A38E94A1C8Ø,1294
289Ø DATA EØØD43EØE5F5ØC7C3527C4B4A3,1769
29ØØ DATA C5ED6CDØCØ4ØØE141F29DF9BEB,1725
291Ø DATA 5BC5FBØ3383DFDC9BF127C4B46,1591
292Ø DATA 847C3E25A385ØCC7E2436DB6FE,17ØØ
293Ø DATA 4AFD4A77CF318ØF72DDE877ØB5,1846
294Ø DATA AØ4DF12D1C2867Ø6DØ36DBA696,1497
295Ø DATA 22E49EØ5EA16A7ØFCCA9D57C36,1627
296Ø DATA DB6DF95DFB9A3B7AAAF8398B8Ø,1998
297Ø DATA 381C6E7198DB71F667DC5A949F,1757
298Ø DATA 44349AF3A5B38ØCØ4413DC8BE5,1856
299Ø DATA 4DC1FAØE1EF58A6DB6DBFØ62B1,1972
3ØØØ DATA AC879EADØA2FC6E53CBAC5C8BB,1952
3Ø1Ø DATA 6DA969DFB7EE5AF54338ØCF639,18ØØ
3Ø2Ø DATA 2F3722F8968DØ38ECAD6724DB6,16Ø9
3Ø3Ø DATA DD4F558D7AAD92E331E893AA6A,1898
3Ø4Ø DATA 4FB93526DCØE1EDØA4DC934ØF1,1663
3Ø5Ø DATA Ø8961EF544B49B7D153B7EØ9DF,1399
3Ø6Ø DATA 13BACDØCF6392F3722F8968DØ3,14Ø3
3Ø7Ø DATA 8ECAD6724DB6DD4F55315EAB64,173Ø
3Ø8Ø DATA B879EAEA9EF5B59C2D696EDBE2,2218
3Ø9Ø DATA 5A3527EØF37827Ø3877E6FAFØF,1373
31ØØ DATA 7AA25A4DF95DFB9A3BCDØC85AC,1779
311Ø DATA FB917CEØDF12D1F9BDECE843C7,2366
312Ø DATA ØD49B6D43EØ721AØ7BCA96F19B,1613
313Ø DATA D3BF79C1B4413F15F1CE7DCDA8,199Ø
314Ø DATA 44BB5293EE52F1D6937C4B47E6,19Ø6
315Ø DATA F7B3C2D76B13BDB86DB6DB743C,2Ø2Ø
316Ø DATA DØ8FB166864413F7Ø8F39F737E,1749
317Ø DATA 245DAØ6A95C7C6638FB1A28415,1675
318Ø DATA F98B527885ACDØ3EF4853C4738,1729
319Ø DATA ØC4ØEDØF1C67AD4CF5DCAA476F,1525
32ØØ DATA DC22FCC2D6396FCCA8F896998E,2147
321Ø DATA 3EC7BD6D67CA9BF68Ø1B6F8968,1772
322Ø DATA EA34ØFE4BC4CDØCØ83F643D7ØF,1867
323Ø DATA B95BC3C516FCBAC5DB8A9DB9ØD,2Ø37
324Ø DATA AØ7DØB343Ø83FBBEEAD61386BA,1755
325Ø DATA 589BAC5C8681B6DF8A118D4EF3,1792
326Ø DATA D67ØEAC6686Ø83B2EFDCEE9DAØ,2281
327Ø DATA 4EE72EB17EEØ7753B6DBF345AB,1968
328Ø DATA 83EA9ØC7C13721AØ7B96B6686Ø,18Ø4
329Ø DATA 838EAAD67B7B5ØB3B43A2D6136,1596
33ØØ DATA DB6DB79F743D5542CA353B7B13,1454
331Ø DATA 5233EDD62E434ØD43ABØEA3F34,1556
332Ø DATA 44DC93E25A668683B429D998E9,1941
333Ø DATA A2387FØ7ØEB17Ø4F896946DF9B,1424
334Ø DATA DE79A18Ø83A1Ø8C88756E1C51E,18Ø5
335Ø DATA AB8BC3ABØE3ED5EDE1D62EC4Ø9,1892
336Ø DATA B877ØA911AØ7B96B66864ØØEB1,1274
337Ø DATA 7Ø7125A4F31EF2978397121E7C,1546
338Ø DATA B751AØ7854E6E6686Ø83F3423E,179Ø
339Ø DATA EDC67AF5ØF813B4A1E5C93EA84,1714
34ØØ DATA 3C3AC6434ØD43AB19A184ØØEB1,1327
341Ø DATA 7Ø74317E676E33D71A4DF5A181,152Ø
342Ø DATA C979B4B7445F446BEØ681EDØ46,1659
343Ø DATA 686Ø83F92BE81C938528B78BAC,1697
344Ø DATA 5C48Ø54EDB877ØA911AØ7B96B6,1514
345Ø DATA 686Ø4ØØEB17Ø7691DF52DDFB81,1736
346Ø DATA DD4EDF9A2D5C1FB94BE1C8681E,1663
347Ø DATA E5AD9A1883C2AD9C93A2D93AC5,2Ø15
348Ø DATA FB81DD4EDC868372D6CDØC83DA,2Ø58
349Ø DATA ABØØ3F3A5ACE49EF52EEB1721A,1537
35ØØ DATA ØF3694A8FA39BC1ØE45DC934ØF,1485
351Ø DATA 1Ø8FC473434ØØEB17Ø7D5143C4,1373
352Ø DATA 77953C3B1Ø26DB6E1DC2A44681,1356
353Ø DATA EE5AD9A18Ø4ØØEB17Ø7D43B724,1612
354Ø DATA FB95BC3C2A91751AØ78Ø899A18,1428
355Ø DATA 83F41C3B429D8FA94BE1DF99A5,1838
356Ø DATA 613CFB983C2D76B1Ø4F6F7C46B,176Ø
357Ø DATA 8F8CFB758B96CB9721AØ9C3AB1,1878
358Ø DATA 9A184ØFC4A548B85Ø7D4AD3BA7,1542
359Ø DATA ØØ7D4294166B8968F7ADACF9BD,1739
36ØØ DATA D3EA531FB8Ø45A9396CB9986B9,18Ø9
361Ø DATA A18Ø4ØF4ED7ØFC6E53CBAC5F5Ø,1941
362Ø DATA A52A76F896946F738CEØ3Ø4ØFB,1824
363Ø DATA A1Ø4F7511F894A917ØAØE5B2E7,179Ø
364Ø DATA 343ØE25A385Ø76D31F1C6634ØF,11Ø9
365Ø DATA BD29E22FC48CFB9B7157D5A93F,189Ø
366Ø DATA 7ØF43A185DF35FC13E25A6686Ø,1527
367Ø DATA E25A3F7ØF434ØFEØE3343Ø8ED6,17Ø9
368Ø DATA 847DE6CØAACFB8B5293F1ØA7C3,19Ø3
369Ø DATA 527DC9CØE1B6DB6DF9D293Ø334,1996
37ØØ DATA 3ØDA8462EØ7ØFEØE1A93A13D29,1536
371Ø DATA CE686ØE25A67DFA1F9953B94E7,2Ø45
372Ø DATA 7E5AØ7ØD74B13CEØ3ØE25A3C4Ø,13Ø1
373Ø DATA CFB9BF2BBF737ØB59AØCD8D3B5,1999
374Ø DATA C3F1B94F39BØ83EE8719EBD1ØA,1916
375Ø DATA 7813D5775C56Ø36DB6DB86DB28,1555
376Ø DATA 5239A18ØF63F7Ø8BE54C476F59,1564
377Ø DATA 5Ø6A4FBAFA1B6DF12D3343E25A,1557
378Ø DATA 51C7656B67AEF2A787E6546851,18Ø8
379Ø DATA 71D95ACDD639FE1A8C8ØEFEAB6,2Ø99
38ØØ DATA CF5B4A5F153B7E3729E5B6DB6D,15Ø8
381Ø DATA B7AA572BØB28A8E69FB1F56963,1717
382Ø DATA E22F896946DBDAØ8CDØCE25A67,1666
383Ø DATA DDØ7CA9BC2317E674824779A18,1462
384Ø DATA E25A3F7ØF4739F737DCA6EØ719,1593
385Ø DATA CØ6ØEE56FØF145BF39BØE25A3F,1965
386Ø DATA 7ØF4739F737DCA6EAB38ØCE25A,1737
387Ø DATA 67DDØ7F72F8A9DE7Ø7BFB827E2,1798
388Ø DATA 4EAF7Ø9B6F7AA1F12D28F36ØF7,1826
389Ø DATA 2DDEA8E1C65E8ØAF7CC5EF5B59,1995
39ØØ DATA F12D1B6E4BCDCCF318E25A3F7Ø,1665
391Ø DATA F4739F737DCA6D1Ø4F5D3Ø67Ø1,14Ø9
392Ø DATA 8ØE25A51C979B8EB17D153B784,1896
393Ø DATA 46BEØØ59A18ØE25A51C979B8EB,1776
394Ø DATA 17CA9B86317DA7ØEAC668685Ø7,1417
395Ø DATA C4B47B5ØDA141352753142114D,1244
396Ø DATA B7C4B4A392F3733CC641CBE25A,2Ø68
397Ø DATA 3D7443F2BBF73CD84ØEDØF1C3F,16Ø3
398Ø DATA 12BC79CFB9B86DB5ØC56ØAEF89,1677
399Ø DATA 6944BCDCCD38FB1E216B45AØ7D,1617
4ØØØ DATA E9ØA788E686ØE25A3F7ØF43A9E,1656
4Ø1Ø DATA 97CØ9AØ7B1BA22E49BF1ØA7C73,1774
4Ø2Ø DATA 78C7343ØC5ED685336DEA5DE2F.175Ø
4Ø3Ø DATA 6B4299BØF8A9DE7DC5D14B5CD8,2Ø55
4Ø4Ø DATA F8A9DE7DC5F42A296B9B86C4C3,2Ø75
4Ø5Ø DATA 1ØA719F717452D736ØF92F7E73,134Ø
4Ø6Ø DATA EE2E8A5AE6CØF92F7E73EE2FA1,1917
4Ø7Ø DATA 514B5CD885359694E336F7ØB5B,1578
4Ø8Ø DATA 368922EFDC2D6CF5DCE336EAØ7,1824
4Ø9Ø DATA C979A3BCF5F7ØB5B3654211D29,15Ø8
41ØØ DATA C66CF8AACØ739BEF4853C67AD1,21Ø9
411Ø DATA 4B5CD8EF4853C3A296B9BØ52A4,1891
412Ø DATA F4A19EB821AF419B52A4F4AØFØ,2Ø65
413Ø DATA 435E8336EE14A13CD8E6BDFB9E,1869
414Ø DATA 7AF7A429E1D14B5CD8EE14FØ73,2ØØ4
415Ø DATA 9BDE52FØ72FDC2D6CD8ØDAØ5ED,2267
416Ø DATA 613F7ØB5B36ØDA118C3B336ØE2,1535
417Ø DATA 5A385Ø7E685DA93C2268D2EF89,15Ø2
418Ø DATA 6946DFC978998DØ66C7EE117E2,1727
419Ø DATA B186DB285Ø59A18ØE25A3C4ØCF,1675
42ØØ DATA B9BF14A9FØDØ3CD8E25A67DDØ7,1936
421Ø DATA E2553B7BAB59DF95A77EF34DB6,192Ø
422Ø DATA DBE25A3F1233EE6F9BDD3D787E,1699
423Ø DATA CB6D9CØ6EE528AE331F7294571,1678
424Ø DATA 98FB8ØØ668421Ø86F1233EE6FA,1675
425Ø DATA 1C6B6B7198DØ3CD87E2578F39F,1676
426Ø DATA 737E6Ø6387C4B4CDØCE25A3F7Ø,1655
427Ø DATA F4739F737DCA6EØ719CØ6Ø4ØFC,17Ø6
428Ø DATA 521BF54AF8EC758BD36CD8E8B4,2115
429Ø DATA B77C4B4CC681EE2B8ED297C45E,1859
```

```
4300 DATA 14ACEØØEA7AAA4779A188EF5D7,183Ø
431Ø DATA 36FB35D9D4CD8ØBE6CBC2266CØ,1934
432Ø DATA E25A3C4ØCFB9BF1268827ECØCD,1798
433Ø DATA ØCC516FCE6CØCD8Ø86BA589E6C,1912
434Ø DATA B43C719EB533D772A91DE6CØB4,1872
435Ø DATA Ø9DCE736CD8ØCD8ØCD8ØCD8ØF1,2Ø87
436Ø DATA 4DØF83971F92B4336ØEA52F8CD,1647
437Ø DATA 8ØCD8ØF838CD8ØCD8ØD89A919B,21Ø1
438Ø DATA CD8Ø8FB57B78CD8ØCD8Ø89ØF3E,178Ø
439Ø DATA 736ØCD8ØEA843C66CØC297C66C,1915
44ØØ DATA EB43Ø3927E6D2DD11CD8CD8ØCD,1722
441Ø DATA 8ØCD8ØF92BE81C938528B78E6C,1766
442Ø DATA EF4853C4736ØEA5BE6CØF8A9DE,2187
443Ø DATA 7DC5EA739BCD8ØCD8ØA2D966CØ,2165
444Ø DATA CD8ØCD8ØCD8ØDEA5F36ØCD8Ø8E,22ØØ
445Ø DATA F2A78CD8F14DØF8397DE9ØA78C,2Ø53
446Ø DATA D8EE56F19BCD8ØF28AC7DE6CEA,2412
447Ø DATA 14EE15336ØEA95E8CD8Ø8EF5D7,1976
448Ø DATA 36F92AB36ØA9E8E672A719BØF8,1981
449Ø DATA ADAD9BCØ63E336F2BBF73CD8F6,2271
45ØØ DATA 5B6CD8EC9E6C5ØF8336ØC2D293,1943
451Ø DATA CD8ØF145B66CB1ØF466CEE59BØ,18Ø6
452Ø DATA E552666C8CB4999BBC2266CØDD,1886
453Ø DATA 44CD8ØD36CD8F12DF9CD8ØBC16,2Ø14
454Ø DATA 6CF14A9F19BØDE52CD8Ø8A5A62,1746
455Ø DATA F9BØE7368E6CBE6CDF36D336F1,2Ø41
456Ø DATA 9BF3696E88E6CØEB43Ø3927E6D,1857
457Ø DATA 2DD11CD8E25A3C4ØCFB9BBC134,1762
458Ø DATA ØF3697C2E1E7AB85ACFB917CEØ,2Ø9Ø
459Ø DATA 3Ø4413F37BDA2FA273B99A18ØØ,14Ø6
```

RUN the above program, and the coded text array (Z%) will be generated. It will be SAVEd if no checksum errors are detected.

The shorter program, printed below, will generate the pointers when RUN. Type it in and RUN it. Again, if there are no checksum errors, the array will be SAVEd.

The BASIC program will LOAD the two separate arrays when you RUN the game.

```
1Ø CLS:FORK=ØTO16:T=Ø:FORJ=ØTO11
2Ø READB:T=T+B
3Ø NEXT:READA:IF T< >A THENPRINT"CHECKSUM
   ERROR INLINE";1ØØØ+K*1Ø:END
4Ø NEXT
5Ø PRINT" SAVING TO TAPE"
6Ø OPEN"O",#-1,"":RESTORE
7Ø FORK=1TO2Ø4:READA:PRINT#-1,A:NEXT:END
1ØØØ DATA Ø,129,212,269,365,428,521,63Ø,691,731,
   766,81Ø,5552
1Ø1Ø DATA 873,969,1ØØ1,1Ø69,1181,1272,1396,1442,
   1514,16Ø1,1663,1733,15714
1Ø2Ø DATA 1789,19Ø5,1948,2Ø41,2144,216Ø,2225,2266,
   233Ø,2447,2472,2513,2624Ø
1Ø3Ø DATA 2579,2621,2679,2738,2766,281Ø,2845,2891,
   2914,2941,2974,3Ø2Ø,33778
1Ø4Ø DATA 3Ø43,3Ø75,31ØØ,3127,3158,3184,3212,3232,
   3265,3291,3312,3353,38352
1Ø5Ø DATA 3392,3414,3431,3468,3479,3485,35Ø8,3523,
   3542,3557,3565,3589,41953
1Ø6Ø DATA 36Ø9,3633,3673,3689,37Ø4,3712,3726,3753,
   3776,3794,3812,383Ø,44711
1Ø7Ø DATA 3854,3866,39Ø3,3927,3932,394Ø,3949,3959,
   3971,3981,3992,3998,47272
1Ø8Ø DATA 4ØØ2,4Ø12,4Ø23,4Ø29,4Ø34,4Ø43,4Ø51,
   4Ø61,4Ø7Ø,4Ø75,4Ø88,4Ø93,48581
1Ø9Ø DATA 41Ø2,4111,4117,414Ø,4152,4165,42ØØ,4217,
   423Ø,4244,4259,4272,5Ø2Ø9
11ØØ DATA 4297,43Ø1,43Ø7,43Ø9,4313,4327,4332,4334,
   4339,4351,4356,4358,51924
111Ø DATA 436Ø,4362,4364,4374,4379,4381,4385,4387,
   4391,4393,4399,44Ø1,52576
112Ø DATA 44Ø6,44Ø8,4413,4417,4427,4429,4431,4433,
   4443,4449,4453,4461,5317Ø
113Ø DATA 4463,4465,4469,4471,4473,4475,4479,4481,
   4486,4496,45ØØ,45Ø2,5376Ø
114Ø DATA 45Ø7,4513,4518,4522,4526,4533,4537,4541,
   4546,455Ø,4553,4557,544Ø3
115Ø DATA 4562,4566,457Ø,4573,4577,4581,4585,4589,
   4592,4597,46ØØ,46Ø5,54997
116Ø DATA 46Ø9,4614,4616,4618,462Ø,4622,4624,4626,
   4632,4642,4653,4665,55541
```

THE DECODER

The Spectrum, Acorn and Dragon/Tandy machines have separate halves to the text compression program described in the three articles starting on 628. These articles contain all the information you'll need about the text compressor. You will only need to use the decode section with this game, so if you haven't typed in the compressor earlier, you only need enter the decompressor.

On the Commodore, the text compressor is in one piece, so you will need to enter the complete compressor to RUN the game.

When using the coder/decoder program from Parts 20 to 22 to produce the DECODE file for Escape, it is important to change the following two lines:

page 633, col. 2, Line 20: should be 2Ø HIMEM=&79ØØ
page 653, col. 2, Line 1900: should be 19ØØ *SAVE DECODE 79ØØ 7A8Ø

In addition, the coding section from Part 20 *must* be typed in. Users of Basic I must add the following line to the game program from Parts 44 to 48:

```
5ØØØ DEFFNINSTR(A$,B$):IFLENA$<LENB$:=Ø
   ELSE:=INSTR(A$,B$)
```

and change INSTR in Lines 55Ø, 66Ø, 199Ø and 332Ø to FNINSTR.

LOADING THE GAME

The order in which the BASIC control program, the text decoder, and the compressed text (and separate pointers, in the case of the Dragon and Tandy) are arranged is important. If you are using tape, rather than disk, be sure to have the separate parts of the game in the correct order on one tape, because it will save a lot of effort when loading up the game. Disk owners should ensure that they have all three parts on the same disk.

There are some differences between the machines in how they handle the mixture between the BASIC and machine code programs, so read the sections below:

To play the game, first LOAD the BASIC control program. When you RUN the program, the decoder will be asked for first, then the coded text. If you are using tape, the programs should be SAVEd in precisely that order—first the BASIC control program, then the decoder, then the text file.

The Commodore 64 needs the text compressor to be LOADed in first of all. The program should be LOADed as machine code—the LOAD command needs ,1 to be added after the device number (1 for tape and 8 for disk). Next type NEW and tap RETURN.

Now the BASIC program should be LOADed. RUNning the program will cause the coded text file to be LOADed.

If you are using tape rather than disk, you will need to change all the device numbers from 8 to 1 throughout the program. Tape users should SAVE the programs in this order—text compressor, BASIC control program, and text file.

The Dragon/Tandy adventure game consists of four separate pieces of software—in this case, the pointers in array A% have been SAVEd separately in array Z%.

First you should LOAD the BASIC control program, then RUN it. It then takes in the decoder, the coded text, and the pointers. Tape users should store the programs in this order—BASIC control program, decoder, coded text, and pointers.

Q+A

How does the adventure game work?

As you have seen, the game consists of three parts. The text code consists of two parts—the coded text held in array Z, or Z%, and the pointers held in array A or A%. The pointers are necessary to select the correct message from the Z array.

THE TOWER OF BABEL

- LANGUAGE DEVELOPMENT
- THE ROOTS OF LANGUAGE
- MOVING ON
- IMPERATIVE LANGUAGES
- FUNCTIONAL LANGUAGES

At present, BASIC is the most popular language for home computers, but recent advances in technology have meant that easier languages could take over

So far, this series on languages has examined LISP, FORTH, Pascal and LOGO. Of course, the INPUT course has concentrated on BASIC, and in particular the article on structured programming (pages 173–178 and 217–219) dealt with the theory of structures in BASIC.

But these are not the only programming languages by any means and more will, no doubt, be developed in the near future. So this article takes a look at some of the other popular languages in existence and in a subsequent article you will see how computing languages are—and may be—developing. However, because some of these languages are such new developments, at this time very few of them will be available on your own machine. Instead, their importance is for the future, rather than the present, of computing.

One of the most important aspects of recent microcomputer development is the rapid increase in available memory. This means that high level languages, which make programming simpler, can be used as native languages in microcomputers. At the end of the 1970's the only languages commonly available on micros were BASIC and machine code. But as available memory gets larger, machines get cheaper and so the number of users increases. As a result, there is a greater demand for different applications—and different languages—so there are already implementations of FORTH, Pascal, LOGO and LISP, which you have seen, plus C, Fortran, Pilot and PROLOG for many popular micros.

An even wider range is already available for business micros, such as the IBM, including APL, Cobol, Occam, PL/1, Snobol and ADA. It seems likely that such languages will become more popular and more widely available as users become more familiar with their advantages. This is likely to increase even more rapidly as big computers become cheap and cheap computers are expanded.

Other languages, like the authoring language Microtext or Acornsoft's new music control language Ample, may be specifically developed for microcompter applications, and languages which are less well known may also be added to the list of available microcomputer variations. The future may, for example, soon see micro implementations of COMAL, Modula-2 or Smalltalk.

LANGUAGE DEVELOPMENT

It is unlikely that all computing languages will eventually be available on micros, if only because there are so many of them. In the 1960s alone it has been estimated that over two hundred new languages were developed, though only about thirteen of these achieved popular success. In fact most of the languages now widely used in both mainframe and micro computing are among these thirteen. However this does not mean that programming languages have remained static since 1969.

All computing languages, like normal human languages such as English, are undergoing constant development, revision and alteration. Thus the FORTRAN in use today has many differences from the original version of 1954, in some ways it is a totally different language. Not only are languages constantly being improved themselves but, because of implementation on different systems and by different organisations, many dialects of the same language may develop.

It is one of the banes of the micro user's life that a program written in one BASIC, say Commodore BASIC, will usually not run on a machine produced by another manufacturer, such as the Spectrum. This is despite the fact that there have been several attempts to create a *standard* BASIC. A standard specification does exist, called the ANSII standard, but the only small micro that actually comes close to adopting it is the ill-fated Newbrain.

Microsoft BASIC has been regarded as a standard by several firms (especially Microsoft!) but has always been adapted by manufacturers using it. One could argue that BBC BASIC and the QL's SuperBASIC are really enhanced versions of Microsoft BASIC but in practice it is the enhancements that are used by programmers using these languages, rather than the standard features.

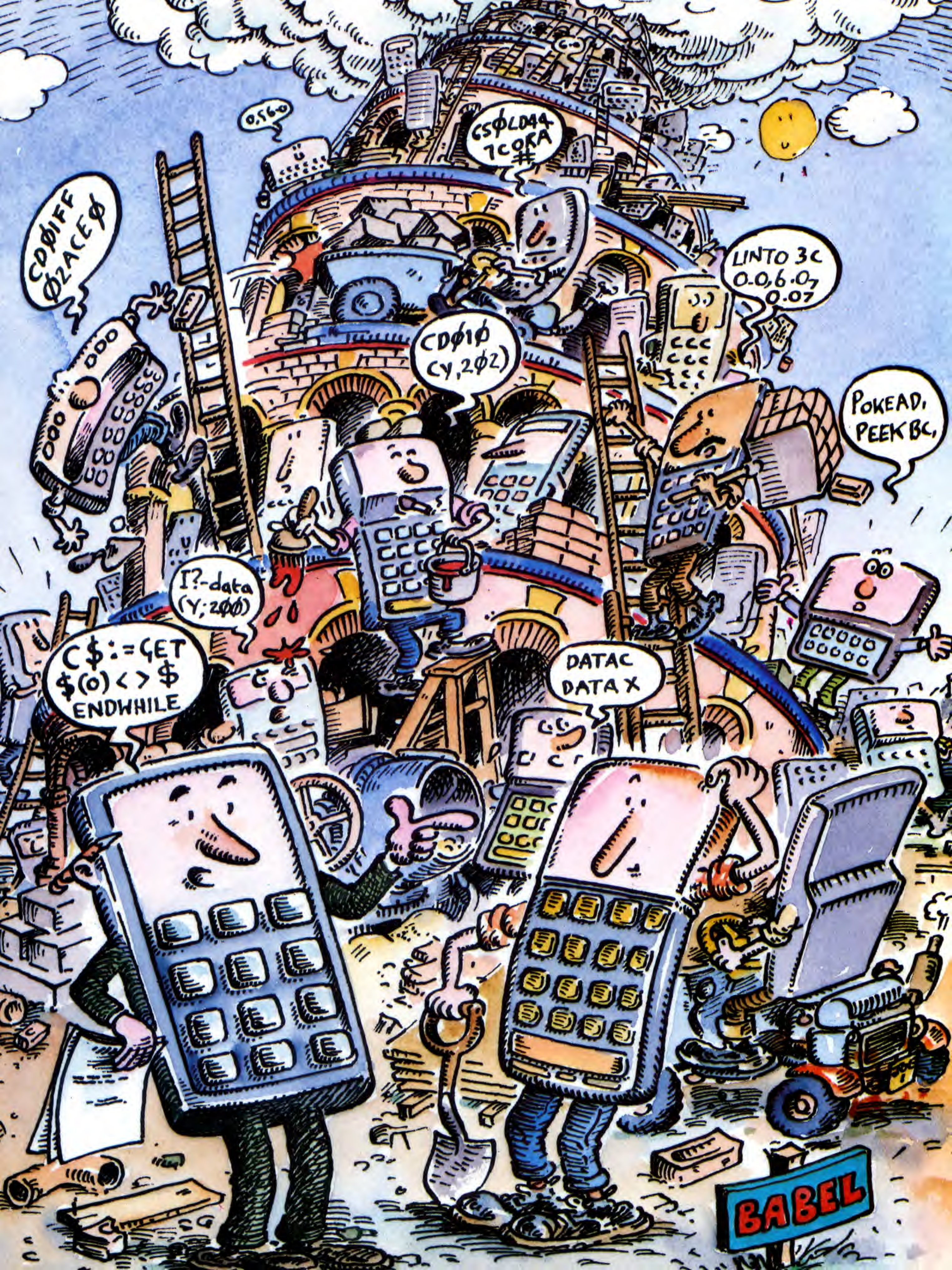
The latest attempt at standardisation is the Japanese MSX standard, which is another implementation of Microsoft BASIC, but which scarcely varies from machine to machine. The only practical variations are that you cannot always use all of the facilities available in MSX because of differences in the hardware of different MSX machines.

The story has not been much different for other languages. Where standards have been more or less successful it has usually been because one firm cornered the market on a particular language. FORTRAN (an name derived from 'FORmula TRANslation') was first developed by IBM and, because it was distributed as free software with IBM machines a standard version was widely adopted. Most developments of FORTRAN were made by IBM who were in a good position to distribute their upgraded versions to the majority of users because they were users of IBM systems. But a language like Pascal, which is probably the second most popular language on microcomputers after BASIC, exists in at least seven different forms—each of which is more-or-less standard for a particular operating system or microprocessor.

THE ROOTS OF LANGUAGE

We can see why this Tower of Babel is an unavoidable consequence of having programming languages by looking at the way programming languages have developed. It is fair to assume that the earliest programming language was FORTRAN. This was developed to be two things – a language for mathematical and scientific formula, and a language which efficiently used computing resources. In both respects it was very successful but at the same time it was rather inadequate from other points of view which had not been thought important for the designers. FORTRAN is excellent for mathematical and scientific purposes, but extremely clumsy (even useless) for processes like text handling or compiling databases. It is also not very structured and often restricts the programmer because of this.

ALGOL60 was developed in the early


0,56-0
CDØIFF Ø2ACEØ
C5ØLD44 7COKA #
LINTO 3C 0-0,6.07 0.07
CDØ1Ø (Y,2Ø2)
POKEAD, PEEK BC,
I?-data (Y;2ØØ)
C$:=GET $(0)<>$ ENDWHILE
DATAC DATA X
BABEL

sixties with a different design philosophy and so with strengths and weaknesses which differed from those of FORTRAN. In particular it was the first of the structured languages. It was intended to be algorithmic in nature, to correspond not to mathematical formulae but to the processes involved in carrying out a task or solving a problem. So it had many of the features now regarded as synonymous with structured programming. In particular its basic unit was the procedure (a self contained block of code with a clearly defined task) and was designed to act upon data and variables of predefined types. Both notions will be familiar to Pascal programmers as Pascal is a direct descendent of ALGOL.

ALGOL60 itself has not been widely used but was probably the most influential of the sixties languages in terms of the way other languages have developed. Compared with this the third granddaddy of computing, COBOL, has been very widely used but has not created many direct descendents (a notable exception being PL/1). COBOL was the result of the realization by the U.S. Department of Defense that all its military installations and research centres needed a common language if the department as a whole was to operate as effectively as possible. It therefore commisioned COBOL, the COmmon Business Oriented Language. Because it was developed by cooperation between government and business teams it was rapidly and widely accepted and is still probably the language most widely used for business applications on mainframes. One probable reason for its success was its attempt to approximate to natural English.

MOVING ON

From these three languages most of the languages of the sixties were developed, including many of the languages we use today. BASIC is best seen first as a development from FORTRAN with the added need to be as easy to use as possible, following the COBOL idea of being friendly to programmers. From ALGOL60 came ALOGOL68 (an improvement which amounts almost to a different language), Pascal and SIMULA. ALGOL and COBOL combined to give PL/1. From Pascal and SIMULA came a group of other languages, such as Modula, Mesa and Euclid which all influenced the production of a language called Ada. SIMULA also gave birth to a language, which currently seems to be enjoying a vogue, called Smalltalk.

IMPERATIVE LANGUAGES

All of these are what is called *imperative* languages. This means that they operate by retrieving data from memory and acting upon it, that is, by assigning data to variables. Most people are so used to this idea that it seems computing could not be carried out in any other way. But there is another family of languages to which FORTH, LISP and LOGO belong. This is the functional (or *applicative*) family of languages which operates primarily through the recursive application of functions.

FUNCTIONAL LANGUAGES

Instead of getting data from memory, performing a computation and then returning the result to a different memory location, a functional language will carry out a series of embedded functions irrespective of the actual addresses involved. Functions act upon parameters, without knowing what those parameters represent. In contrast imperative languages act upon actual addresses, actual memory locations, and therefore have to be rewritten to do different jobs. If you want a function to do a different job you simply give it different parameters.

Functional languages tend to be modular. Imperative languages tend to be linear and sequential in structure. If you understand the way that a word can be defined in FORTH or that a list can be used in LISP you will know that a single word or list can be, in effect, a program. This is because it is defined in terms of other words or lists which are themselves defined in terms of other words and lists—and so on down the line until you reach definitions in terms of the basic units of the program. These are the functions which carry out so-called primitive operations.

In a language like Microsoft BASIC, on the other hand, a program consists of a series of statements which have to be carried out one after another in an exact order for the program to work successfully. Microsoft BASIC is an imperative language. If the order of its statements is altered then certain memory locations (variables) will not hold the values required by later statements.

In a functional program the order of execution is usually unimportant providing the correct functions are used. If a function is used correctly it will always give the desired result irrespective of the actual values held in any memory locations at any point during execution of the program.

Functional languages thus give an approach to programming which is very different from that of imperative programming. All such languages, which ultimately derive from LISP (also developed in the sixties) enable

new languages to be developed from the basic set of functions defined in the language. As a result they are infinitely extendable. In a sense any program in LISP is an extension of the language, created by providing new functions. The programmer has much more control over what he or she is doing with a functional language than with an imperative language, but needs to have a good, clearly defined approach to the task in hand.

Functional languages require the same kind of programming discipline as the structured procedural languages. For this reason an area which is likely to develop is in languages which combine both structural, procedural approaches to a task and functional extensions of the given language.

EXTENSIONS

An obvious development along these lines can be seen in BASIC. ANSII standard BASIC provides a usable imperative language. Microsoft BASIC extended that standard. BBC BASIC added a degree of structure to that standard and, by implementing recursion in procedures and functions, allowed some functional elements to be added to the language. Sinclair's QL SuperBASIC goes a little further by adding more in the way of structure (such as the SELECT statement which is similar to Pascal's CASE) and also allows functions and procedures to be called by name alone so that one could write a LISP interpreter using BASIC. Future developments are likely to induce attempts to develop microcomputer languages which move away from the ANSII minimum towards, on the one hand, structure as typified by Pascal and, on the other, functional flexibility as found in LISP and FORTH.

LISP, as you have already seen, was developed as a language for Artificial Intelligence and it has remained the major language used in that field. One important characteristic of Artificial Intelligence work has been the úse of character strings and LISP showed that list processing was a useful way to approach string handling.

SNOBOL (StriNg Oriented symBOlic Language) was a language developed for string handling following the philosphy of LISP. It has been used primarily in processing ordinary natural language, as in deciding the authorship of manuscripts or performing computerised stylistic analysis of texts.

The other area that functional programming has successfully developed is in systems programming, such as in writing compilers for other languages. BCPL is one such language which Acornsoft have made available for the BBC. However, a much more popular descendant of BCPL is C. C was used to write the Unix operating system and together they provide a powerful set of programming tools for systems development.

Unfortunately C is very much a 'programmer's' language being much more unfriendly than languages such as LISP and Pascal. It owes this largely to its origins and the fact that it exists at a level of application and versatility somewhere between the high level languages like BASIC and the distinctly unfriendly but more versatile assemblers used to program individual microprocessors. But it is increasing in popularity especially as Unix becomes more widespread and organisations and individuals want to develop their own systems which are not just passively running other people's software. There are now implementations of C for several micros, including the Spectrum.

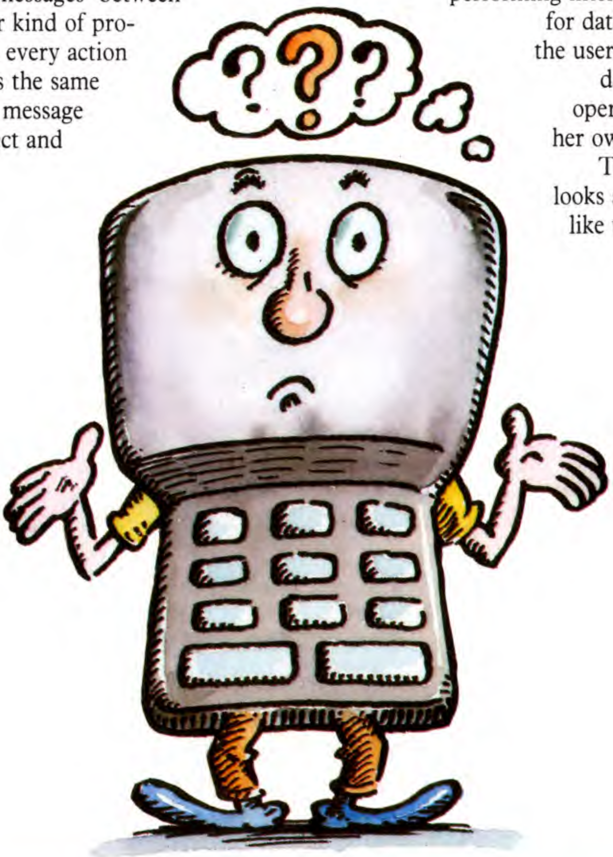
Smalltalk has already been mentioned as an imperative language but it also draws quite heavily on the LISP tradition. It is similar to C because it provides a complete programming environment, not just a language, but it is very different from C in its aim to simplify the task of information handling as much as possible. It does this by simplifying the concept of the program. Every item in a Smalltalk program is regarded as an object and all processing is done by passing what are called 'messages' between objects. No other kind of processing exists, so every action in the program is the same kind of thing—a message is sent to an object and the object will either send a message back, or send a message or messages to other objects.

Objects may do some internal processing according to the messages they receive. So an object in Smalltalk is something like a LISP function and also like a Pascal procedure. But rather than being like building up procedures in a tree structure or linking functions into a recursive nest, objects in Smalltalk can be thought of as independent entities each of which can pass messages to any other. It is something like possessing a database in which each record of data can interrogate and perform actions on other records. Each record is itself an object, but it can also interact with other objects. So an object can be seen both as data and program.

This idea is rather hard to grasp until you see it working and, Smalltalk is not yet widely available on micros. But it does represent one of the major trends in computer languages which is likely to be important in the near future. Developments like Dbase III seem to be converging on similar ground from a different direction, while COBOL data structures created the idea of the database management system.

Database systems have now developed to such an extent that they begin to resemble programming languages, allowing sequences of operations on data to be held as 'programs', performing intelligent searches for data and allowing the user to modify the database and its operation to his or her own satisfaction.

The next article looks at more trends like these in detail.



CLIFFHANGER: CHECKING FOR FIT

The game is now complete. Willie is ready to scale the cliff and challenge the sea, the boulders, the snakes and the potholes. But will he be able to overcome the bugs?

Cliffhanger is now complete, but is it working? It is unlikely that you could have keyed in such a long program without introducing errors. In addition, there are a few amendments which need to be made to the listings—these are being published in the errata sections that appear in the index.

Once you have ironed these out, it is time to tackle any problems that have been introduced by typing or copying errors. These can be discovered by checking your assembly listings carefully and reassembling if necessary.

When you have got the individual routines working, load them all into memory, SAVE all of it in one chunk (see pages 992 to 997) and test the whole lot by calling it with the machine code execution command for your computer (see pages 1537 to 1544).

Because Cliffhanger is modular, it is easy to debug. If you have any trouble getting the game to run you should be able to track down the part responsible and concentrate your error-trapping efforts on that.

Even if all the individual routines work independently, it is possible that the whole program together may not. Routines call other routines and if one of the calls is wrong, or one of the routines is assembled in the wrong place, the chances are the program won't work.

So, as a final check *INPUT* is publishing a complete hex dump of the game so that you can double-check your final assembled version.

When you have assembled Cliffhanger, it is lodged in memory as a series of hex numbers in successive memory locations. Here we are giving you the complete game in hex. The four digit number on the left-hand side of the column is the address of the memory location occupied by the first two-digit byte of the program in that line. The subsequent pairs of digits occupy the subsequent locations. So you can check each byte of Cliffhanger against the contents of the corresponding memory location. That way you can make sure you have assembled each instruction in the game right.

Unlike other games, you need never be bored by Cliffhanger. Anytime you feel a twinge of tedium coming on, you can change its innermost workings. Its modular construction also means that there is plenty of scope for you to make modifications. You can easily customize Cliffhanger to suit your personal preferences. Be bold. Experiment. After all, Cliffhanger is now your game. And it can be anything you want it to be.

The game is written in the 50,000 area so that it can be assembled by the *INPUT* assembler.

Remember to check that the routine that plays the tune has been shifted so that it does not overwrite other routines (see page 1145).

If you are not using the *INPUT* assembler you may have to shift the game in memory because the space is itself occupied by the assembler. The best place to relocate it is to the 20,000 area. To do that just change all the addresses that begin with a 5 so they start with a 2.

Use your machine code monitor to PEEK your Spectrum's memory and check that it tallies with the following hex dump. If you have relocated it, the location addresses will of course be different, but the hex should be the same:

```
DEA8 18 3C 3C 18 3C 3C 3C 3C 3C 3C 18 18 18 18 18 1E
DEB8 Ø1 Ø3 Ø3 Ø1 ØØ Ø1 Ø1 Ø1 8Ø CØ 6Ø 8Ø ØØ ØØ ØØ EØ
DEC8 ØE ØØ Ø1 Ø2 Ø4 Ø8 Ø4 ØØ ØØ ØØ 8Ø 4Ø 2Ø 2Ø 3Ø ØØ
DED8 ØØ ØØ ØØ ØØ 18 3C 3C 18 ØØ 1Ø 1Ø 1E EØ ØØ ØC 24
DEE8 42 82 43 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø1 Ø3 Ø3 Ø1
DEF8 ØØ ØØ ØØ ØØ 8Ø CØ CØ 8Ø ØØ Ø1 Ø1 Ø1 ØE ØØ Ø1 Ø2
DFØ8 ØØ ØØ ØØ EØ ØØ ØØ 8Ø 4Ø Ø4 Ø8 Ø4 ØØ ØØ ØØ ØØ ØØ
DF18 2Ø 2Ø 3Ø ØØ ØØ ØØ ØØ ØØ 1C 3E 7F FF FF FE FC 38
DF28 Ø3 Ø7 ØF ØF ØF Ø7 Ø3 Ø1 8Ø CØ EØ FØ FØ FØ EØ CØ
DF38 ØØ ØØ Ø7 18 2Ø 4Ø 4Ø 8Ø ØØ ØØ 1F AØ CØ ØØ ØØ ØØ
DF48 ØØ ØØ 8Ø 4Ø 5C 22 Ø2 Ø2 8Ø 4Ø 7C Ø2 Ø2 Ø1 ØØ ØØ
DF58 ØØ ØØ ØØ Ø4 ØA 11 6Ø ØØ Ø2 Ø4 Ø8 Ø4 Ø4 Ø4 F8 ØØ
DF68 ØØ ØØ 78 86 Ø1 Ø1 ØØ ØØ ØØ ØØ 1E 61 8Ø 8Ø ØØ ØØ
DF78 ØØ ØØ ØØ ØØ 87 79 ØØ ØØ ØØ ØØ ØØ ØØ E1 9E ØØ ØØ
DF88 22 14 Ø8 Ø8 Ø8 Ø8 Ø8 Ø8 18 3C 36 3E 7E 3C 18 18
DF98 18 18 ØC ØC Ø6 Ø6 Ø3 Ø3 Ø6 Ø6 ØC ØC 18 18 3Ø 3Ø
DFA8 6Ø 6Ø C6 C3 66 6C 38 38 84 D6 FF FF FF FF FF FF
DFB8 ØØ Ø1 Ø3 Ø7 1F 3F 7F FF ØØ ØØ FF FF 3C 3C FF FF
DFC8 Ø6 Ø8 76 FF FF FF 3E 3C 1Ø 1Ø 1Ø 38 38 38 38 38
DFD8 1Ø 1C 7C 38 38 38 1Ø 1Ø ØØ ØØ ØØ 2Ø 51 8A Ø4 ØØ
DFE8 ØØ ØØ ØØ 82 45 28 1Ø ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
DFF8 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
EØØ8 ØØ ØØ ØØ ØØ ØØ ØØ ØØ 62 ØØ B4 Ø5 E9 ØØ C7 Ø4 83
EØ18 ØØ 3E Ø4 DC ØØ C4 Ø3 4E ØØ 8C Ø3 93 ØØ C4 Ø3 Ø5
EØ28 Ø1 3E Ø4 6E ØØ 11 Ø5 83 ØØ 6A Ø6 31 ØØ B4 Ø5 6E
EØ38 ØØ 11 Ø5 E9 ØØ C7 Ø4 62 ØØ B4 Ø5 93 ØØ B4 Ø5 2C
EØ48 ØØ 6A Ø6 62 ØØ B4 Ø5 DC ØØ 11 Ø5 5C ØØ ØC Ø6 DC
EØ58 ØØ A7 Ø7 43 4C 49 46 46 48 41 4E 47 45 52 43 52
EØ68 45 41 54 45 44 2Ø 42 59 2Ø 41 2E 44 4F 45 57 52
EØ78 49 54 54 45 4E 2Ø 42 59 2Ø 5Ø 2E 43 4C 41 52 4B
EØ88 2Ø 41 66 74 65 72 2Ø 61 2Ø 73 68 6F 72 74 2Ø 77
EØ98 61 6C 6B 2Ø 57 69 6C 6C 69 65 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
EØA8 72 65 74 75 72 6E 73 2Ø 74 6F 2Ø 66 69 6E 64 2Ø
EØB8 74 68 65 2Ø 67 6F 61 74 73 2Ø 68 61 76 65 2Ø 2Ø
EØC8 73 7Ø 72 65 61 64 2Ø 68 69 73 2Ø 7Ø 69 63 6E 69
EØD8 63 2Ø 67 6F 6F 64 69 65 73 2Ø 61 6C 6C 2Ø 2Ø 2Ø
EØE8 6F 76 65 72 2Ø 61 2Ø 72 6F 63 6B 79 2Ø 65 6D 62
EØF8 61 6E 6B 6D 65 6E 74 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
E1Ø8 57 69 6C 6C 69 65 2Ø 73 65 74 73 2Ø 6F 66 66 2Ø
E118 74 6F 2Ø 72 65 63 6C 61 69 6D 2Ø 68 69 73 2Ø 2Ø
E128 6C 6F 73 74 2Ø 7Ø 6F 73 73 65 73 73 69 6F 6E 73
E138 2C 62 75 74 2Ø 69 73 2Ø 68 61 6D 7Ø 65 72 65 64
E148 62 79 2Ø 66 61 6C 6C 69 6E 67 2Ø 62 6F 75 6C 64
E158 65 72 73 2C 7Ø 6F 74 2Ø 68 6F 6C 65 73 2Ø 2Ø 2Ø
E168 61 6E 64 2Ø 76 69 63 69 6F 75 73 2Ø 73 6E 61 6B
E178 65 73 2E 54 6F 2Ø 6D 61 6B 65 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
E188 6D 61 74 74 65 72 73 2Ø 77 6F 72 73 65 2Ø 74 68
E198 65 2Ø 74 69 64 65 2Ø 69 73 2Ø 72 69 73 69 6E 67
E1A8 61 6E 64 2Ø 68 65 2Ø 69 73 2Ø 69 6E 2Ø 64 61 6E
E1B8 67 65 72 2Ø 6F 66 2Ø 62 65 69 6E 67 2Ø 63 75 74
E1C8 6F 66 66 2E 54 6F 2Ø 68 65 6C 7Ø 2Ø 57 69 6C 6C
E1D8 69 65 2Ø 69 6E 2Ø 68 69 73 2Ø 71 75 65 73 74 2Ø
E1E8 72 65 61 64 2Ø 74 68 65 2Ø 66 6F 6C 6C 6F 77 69
E1F8 6E 67 2Ø 61 6E 64 2Ø 7Ø 72 65 73 73 2Ø 27 53 27
E2Ø8 74 6F 2Ø 73 74 61 72 74 2E 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
E218 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 4E 2Ø 2Ø 2Ø 2Ø 2D 2Ø
E228 52 75 6E 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
E238 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 4D 2Ø 2Ø 2Ø 2Ø 2D 2Ø
E248 56 65 72 74 69 63 61 6C 2Ø 6A 75 6D 7Ø 2Ø 2Ø 2Ø
E258 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 42 6F 74 68 2Ø 2D 2Ø
E268 44 69 61 67 6F 6E 61 6C 2Ø 6A 75 6D 7Ø 53 43 4F
E278 52 45 2D 3Ø 3Ø 3Ø 3Ø 3Ø 3Ø 4C 49 56 45 53 2D 35
E288 47 41 4D 45 2Ø 4F 56 45 52 2Ø 21 21 21 23 23 21
E298 23 23 23 21 23 23 21 23 21 23 23 23 23 21 23 21
E2A8 23 23 23 23 21 23 23 21 23 23 23 CD 5Ø E3 3E Ø2
E2B8 D3 FE 3E 1Ø 32 48 5C DD 21 5B EØ Ø6 Ø5 3E 46 21
E2C8 86 ØØ CD 2B E3 Ø6 Ø6 21 CC ØØ CD 2B E3 Ø6 1Ø 3E
E2D8 Ø7 21 62 Ø2 CD 2B E3 Ø6 12 21 A2 Ø2 CD 2B E3 Ø6
E2E8 Ø2 21 E8 FD 11 ØØ ØØ 2B E5 ED 52 E1 2Ø F9 1Ø F1
E2F8 CD 5Ø E3 DD 21 88 EØ 21 2Ø ØØ 3E Ø7 Ø6 FF CD 2B
E3Ø8 E3 Ø6 8A CD 2B E3 11 27 ØØ 19 Ø6 64 3E 46 CD 2B
E318 E3 3E FD DB FE CB 4F 2Ø F8 C9 3E FD DB FE CB 4F
E328 2Ø F8 C9 C5 F5 DD 7E ØØ CD 3E E3 F1 CD 69 E3 23
```

- TRACING THE BUGS
- TESTING THE ROUTINES
- ENTERING THE GAME
- SAVING CLIFFHANGER
- THE HEX DUMP

The 'CLIFFHANGER' listings published in this magazine and subsequent parts bear absolutely no resemblance to, and are in no way associated with, the computer game called 'CLIFF HANGER' released for the Commodore 64 and published by New Generation Software Limited.

```
E338 DD 23 C1 10 EE C9 E5 21 F8 3C 11 08 00 06 1F 90
E348 19 3D 20 FC E5 C1 E1 C9 DD 21 00 40 21 00 1B 3E
E358 00 DD 77 00 DD 23 2B E5 11 00 00 ED 52 E1 20 F1
E368 C9 F5 E5 C5 E5 D1 7A FE 01 38 11 D5 11 00 07 19
E378 D1 7A FE 01 28 06 D5 11 00 07 19 D1 D5 11 00 40
E388 19 D1 3E 08 C1 F5 0A 77 24 03 F1 3D 28 03 F5 18
E398 F5 E1 F1 D5 11 00 58 19 D1 77 D5 E1 C9 00 00 00
E3A8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E3B8 00 00 00 00 00 00 00 3E 10 32 F0 DF DD 21 B2 E2
E3C8 06 20 C5 CD 35 E4 3E 00 32 F1 DF DD 7E 00 DD 2B
E3D8 FE 21 20 0D 05 3A F0 DF 3D 32 F0 DF 3E 01 32 F1
E3E8 DF 3A F0 DF 47 21 1F 00 3E 2D CD 48 E4 01 B0 DF
E3F8 3A F1 DF FE 01 20 03 01 B8 DF 3E 2C CD 69 E3 3A
E408 F0 DF 47 3E 17 90 47 3E 20 11 20 00 19 CD 48 E4
E418 C1 10 AF 21 31 00 06 0C 3E 29 DD 21 75 E2 CD 2B
E428 E3 21 71 00 06 07 CD 2B E3 CD 57 E4 C9 21 00 40
E438 06 D8 0E 1F 23 7E 2B 77 23 0D 20 F8 23 10 F3 C9
E448 C5 01 00 3D CD 69 E3 11 20 00 19 C1 10 F2 C9 3A
E458 00 E0 21 B8 DF 47 04 11 08 00 19 10 FD E5 C1 21
E468 BF 00 3E 3A CD 69 E3 3A 00 E0 FE 00 28 0C F5 CD
E478 83 E4 F1 FE 01 28 03 CD A9 E4 C9 21 C9 01 CD 96
E488 E4 21 91 01 CD 96 E4 21 3A 01 CD 96 E4 C9 06 04
E498 C5 01 00 3D 3E 2D CD 69 E3 11 20 00 19 C1 10 F0
E4A8 C9 21 C9 01 CD BC E4 21 91 01 CD BC E4 21 3A 01
E4B8 CD BC E4 C9 3E 04 01 90 DF F5 3E 2B CD 69 E3 11
E4C8 20 00 19 F1 3D 20 F2 C9 CD B3 E2 3E 05 32 FF DF
E4D8 3E 00 32 00 E0 21 00 00 22 F9 DF 22 FB DF 22 FD
E4E8 DF 3E 13 32 65 EA 3E 06 32 09 E0 21 E0 02 22 0A
E4F8 E0 21 82 00 22 01 E0 3E 03 32 03 E0 3E 00 32 04
E508 E0 3E 02 32 05 E0 21 C1 01 22 F4 DF 21 00 00 22
E518 F6 DF 3E 00 32 F8 DF 21 DF 00 22 0C E0 3E 00 06
E528 05 32 06 E0 80 32 07 E0 80 32 08 E0 CD BF E3 CD
E538 3B E6 21 77 00 3A FF DF 06 30 80 CD 3E E3 3E 29
E548 CD 69 E3 CD 60 EA CD 11 E7 CD 71 E6 CD AF E9 CD
E558 02 E6 CD AB E5 CD 7F E5 3A F8 DF FE 01 CA 8C E9
E568 FE 02 CA 04 E9 06 32 3E FF 3D 20 FD 10 FB 3E FE
E578 DB FE CB 47 20 D0 C9 3A 05 E0 3C CB 9F 32 05 E0
E588 01 68 DF FE 04 38 03 01 78 DF 3E 2E 21 2A 00 C5
E598 CD 69 E3 23 CD 69 E3 C1 21 44 00 CD 69 E3 23 CD
E5A8 69 E3 C9 3A 03 E0 3D 32 03 E0 FE 00 28 01 C9 3E
E5B8 06 32 03 E0 3E 2D 01 00 40 2A 01 E0 16 03 1E 02
E5C8 CD 5A E6 3A 04 E0 FE 00 28 03 2B 18 01 23 22 01
E5D8 E0 01 38 DF 3E 2F 16 03 1E 02 CD 5A E6 11 81 00
E5E8 ED 52 20 06 3E 00 32 04 E0 C9 11 90 00 2A 01 E0
E5F8 ED 52 20 05 3E 01 32 04 E0 C9 01 E0 DF 3A 09 E0
E608 CB 57 28 03 01 E8 DF 2A 0A E0 3E 0F 16 20 D5 C5
E618 CD 69 E3 23 C1 D1 15 20 F5 3A 09 E0 3D 32 09 E0
E628 20 10 3E 0A 32 09 E0 2A 0A E0 11 20 00 ED 52 22
E638 0A E0 C9 21 37 00 DD 21 F9 DF 06 06 C5 DD 7E 00
E648 06 30 80 CD 3E E3 3E 29 CD 69 E3 23 DD 23 C1 10
E658 EB C9 E5 D5 E5 D5 CD 69 E3 23 D1 15 20 F7 E1 11
E668 20 00 19 D1 1D 20 EC E1 C9 3A 00 E0 FE 00 28 05
E678 FE 03 28 01 C9 3A 0E E0 FE 01 28 55 2A 0C E0 01
E688 20 DF 3E 2A CD 69 E3 23 3E 2D 01 00 3D CD 69 E3
E698 2A 0C E0 11 E0 01 ED 52 28 5D 2A 0C E0 11 20 58
E6A8 19 7E FE 0F 28 51 FE 2D 20 1A 2A 0C E0 01 00 3D
E6B8 3E 2D CD 69 E3 11 20 00 19 22 0C E0 01 20 DF 3E
E6C8 2A CD 69 E3 2A 0C E0 2B 22 0C E0 3E 01 32 0E E0
E6D8 C9 2A 0C E0 11 00 58 19 7E FE 28 20 05 3E 02 32
E6E8 F8 DF 2A 0C E0 3E 2A 01 28 DF CD 69 E3 23 CD 69
E6F8 E3 3E 00 32 0E E0 C9 2A 0C E0 01 00 3D 3E 2D CD
```

```
E708 69 E3 21 DF 00 22 0C E0 C9 3A F7 DF FE 00 C2 C8
E718 E7 3A F6 DF FE 01 28 5C 2A F4 DF 2B 01 00 40 3E
E728 2D 11 02 02 CD 5A E6 01 A8 DE 3E 28 23 11 02 01
E738 CD 5A E6 11 40 58 19 7E FE 2D CA C2 E7 FE 2B 28
E748 79 FE 0F 28 75 3E 00 DB FE CB 57 20 0E 06 01 CB
E758 5F 20 02 06 81 78 32 F7 DF 18 09 CB 5F 20 05 3E
E768 01 32 F6 DF 2A F4 DF 11 BF 00 ED 52 30 05 3E 01
E778 32 F8 DF C9 11 03 00 21 0C 06 CD B5 03 2A F4 DF
E788 11 21 58 19 7E FE 2B 28 31 FE 2C 28 27 FE 2A 28
E798 29 11 20 00 19 7E FE 0F 28 20 FE 2D 28 1C FE 2B
E7A8 28 18 2A F4 DF 3E 28 01 B8 DE 11 02 02 CD 5A E6
E7B8 23 22 F4 DF 3E 00 32 F6 DF C9 3E 02 32 F8 DF C9
E7C8 11 06 00 21 F7 02 CD B5 03 3A F7 DF FE 01 20 1B
E7D8 3C 32 F7 DF 2A F4 DF 11 20 00 ED 52 22 F4 DF 01
E7E8 D8 DE 3E 28 11 03 01 CD 5A E6 C9 FE 02 20 1F 3C
E7F8 32 F7 DF 2A F4 DF 01 A8 DE 3E 28 11 02 01 CD 5A
E808 E6 11 40 00 19 3E 2D 01 00 3D CD 69 E3 C9 FE 03
E818 20 1A 3C 32 F7 DF 2A F4 DF 01 D8 DE 3E 28 11 03
E828 01 CD 5A E6 11 20 00 19 22 F4 DF C9 FE 04 20 18
E838 3E 00 32 F7 DF 2A F4 DF 11 20 00 ED 52 01 00 3D
E848 3E 2D CD 69 E3 C3 11 E7 FE 81 20 33 3C 32 F7 DF
E858 2A F4 DF 11 21 58 19 7E FE 2B CA C2 E7 FE 2C 20
E868 07 2A F4 DF 2B 22 F4 DF 2A F4 DF 11 20 00 ED 52
E878 22 F4 DF 01 F0 DE 11 03 02 3E 28 CD 5A E6 C9 FE
E888 84 28 5B 3C 32 F7 DF 3A F6 DF FE 01 28 23 2A F4
E898 DF 01 00 40 3E 2D 11 03 02 CD 5A E6 23 22 F4 DF
E8A8 01 A8 DE 3E 28 11 02 01 CD 5A E6 3E 01 32 F6 DF
E8B8 C9 2A F4 DF 01 B8 DE 3E 28 11 02 02 CD 5A E6 23
E8C8 22 F4 DF 11 40 58 19 7E FE 2C 20 0C 3E 00 32 F7
E8D8 DF 3E 04 06 05 CD FC E9 3E 00 32 F6 DF C9 3E 00
E8E8 32 F7 DF 2A F4 DF 2B 01 00 40 3E 2D 11 02 02 CD
E8F8 5A E6 11 21 00 19 22 F4 DF C3 11 E7 11 C4 00 21
E908 3E 04 CD B5 03 11 83 00 21 6E 06 CD B5 03 2A F4
E918 DF 01 00 3D 3E 2D CD 69 E3 11 20 00 19 22 F4 DF
E928 01 A8 DE 3E 28 11 02 01 CD 5A E6 11 1E 00 21 6E
E938 03 ED 5F 6F CD B5 03 2A F4 DF 11 C0 02 ED 52 38
E948 CD 3E 09 32 65 EA 3A FF DF 3D 32 FF DF C2 EE E4
E958 21 4A 01 3E 8E 06 0B DD 21 88 E2 CD 2B E3 CD 3B
E968 E6 11 05 01 21 6E 06 CD B5 03 11 88 01 21 3E 04
E978 CD B5 03 11 05 01 21 6E 06 CD B5 03 3E 32 32 6E
E988 E5 C3 D0 E4 11 0B 02 21 26 03 CD B5 03 3A 00 E0
E998 3C CB 97 32 00 E0 3A 6E E5 3D 32 6E E5 3E 03 06
E9A8 05 CD FC E9 C3 E9 E4 3A 00 E0 FE 02 30 01 C9 DD
E9B8 21 06 E0 21 A9 01 CD CE E9 21 71 01 CD CE E9 21
E9C8 1A 01 CD CE E9 C9 E5 ED 5B 0A E0 ED 52 E1 38 01
E9D8 C9 DD 7E 00 3C CB A7 DD 77 00 DD 23 FE 07 30 01
E9E8 C9 01 88 DF 16 2B FE 0F 20 05 01 00 3D 16 2D 7A
E9F8 CD 69 E3 C9 DD 21 F9 DF 16 00 5F DD 19 DD E5 CD
EA08 12 EA DD E1 10 F7 CD 3B E6 C9 DD 7E 00 3C FE 0A
EA18 20 09 3E 00 DD 77 00 DD 2B 18 EF DD 77 00 C9 00
EA28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EA38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EA48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EA58 00 00 00 00 00 00 00 00 DD 21 0F E0 06 13 C5 DD
EA68 56 01 DD 5E 00 DD 66 03 DD 6E 02 DD E5 CD B5 03
EA78 DD E1 11 04 00 DD 19 C1 10 E4 C9 00 00 00 00 00
```

Watch out for addresses—apart from the origin—given in decimal. The *INPUT* Assembler will accept decimal and hex numbers.

Use your machine code monitor to PEEK your Commodore's memory and check that it tallies with the following hex dump.

```
3000 3C 42 99 91 91 99 42 3C 38 44 82 82 FE 82 82 00
3010 F8 84 82 FC 82 82 FC 00 7C 82 80 80 80 82 7C 00
3020 F8 84 82 82 82 84 F8 00 F8 84 80 F8 80 84 F8 00
3030 F8 84 80 F8 80 80 80 00 78 84 80 98 84 84 78 00
3040 84 84 84 FC 84 84 84 00 38 10 10 10 10 10 38 00
3050 38 10 10 10 10 90 70 00 84 88 90 E0 90 88 84 00
3060 80 80 80 80 80 80 FC 00 82 C6 AA 92 82 82 82 00
3070 82 C2 A2 92 8A 86 82 00 7C 82 82 82 82 82 7C 00
3080 7C 82 82 FC 80 80 80 00 7C 82 82 92 8A 86 7C 00
3090 7C 82 82 FC 88 84 82 00 78 84 80 34 0C 84 78 00
30A0 FE 92 10 10 10 10 10 00 82 82 82 82 82 82 7C 00
30B0 82 82 82 82 44 28 10 00 82 82 92 92 92 92 7C 00
30C0 82 44 28 10 28 44 82 00 82 44 28 10 10 10 10 00
30D0 FE 04 08 10 20 40 FE 00 38 38 10 FE 28 44 82 00
30E0 AA AA AA FF FF FF FF FF 03 07 0F 1F 1F 3F 3F 7F
30F0 DF EF DF EF CF B7 FF FF DF EF FF FF EF F7 FF FF
3100 00 00 00 00 00 00 00 00

3180 7C 82 82 82 82 82 7C 00 10 30 10 10 10 10 38 00
3190 7C 82 02 7C 80 80 FE 00 7C 82 02 7C 02 82 7C 00
31A0 90 90 90 FC 10 10 10 00 FC 80 80 F8 04 04 FC 00
```

```
31B0 3C 40 80 F8 84 84 FC 00 FC 04 04 08 10 20 40 00
31C0 78 84 84 78 84 84 78 00 78 84 84 7C 04 04 78 00
31D0 00 00 24 5A 99 00 00 00 00 00 E7 18 18 00 00 00
31E0 81 42 24 18 18 00 00 00 00 00 E7 18 18 00 00 00
31F0 44 AA 09 00 44 AA 09 00 DF BB B7 DB BF DB DB EF
3400 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3410 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3420 20 20 20 20 20 20 20 20 0C 09 16 05 13 20 20 20
3430 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3440 20 20 20 3D 20 20 20 20 20 20 20 20 20 20 20 20
3450 0C 05 16 05 0C 20 20 20 20 20 20 20 20 20 20 20
3460 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3470 30 20 20 20 20 20 20 20 13 03 0F 12 05 20 30 30
3480 30 30 30 30 20 20 20 20 20 20 20 20 20 20 20 20
3490 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
34A0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
34B0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
34C0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
34D0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
34E0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 1D 1C
34F0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3500 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3510 20 1D 1C 1C 1C 1C 1F 3F 20 20 20 20 20 20 20 20
3520 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3530 20 20 20 20 20 20 20 1D 1C 3F 1E 1F 3F 1F 1E 3F
3540 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3550 20 20 20 20 20 20 20 20 20 20 20 20 1D 1C 1C 3F
3560 1E 1F 1F 1E 1F 1E 3F 1F 20 20 20 20 20 20 20 20
```

```
3570 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3580 20 20 1D 1C 1E 3F 1F 3F 1E 3F 1E 3F 1F 3F 1E 3F
3590 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
35A0 20 20 20 20 20 20 20 20 1D 1C 1F 1E 1E 3F 3F 1F
35B0 1F 1E 1F 1F 1F 3F 1F 3F 20 20 20 20 20 20 20 20
35C0 20 20 20 20 20 20 20 20 20 20 20 20 1D 1C 1C 1C
35D0 3F 3F 1E 1F 3F 1E 1F 3F 1E 1E 1E 3F 1E 3F 1F 1E
35E0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
35F0 1D 1C 1C 1C 1F 1E 1F 1F 3F 1E 1F 3F 1E 3F 3F 1F
3600 1F 3F 1E 1E 1F 3F 1F 1F 20 20 20 20 20 20 20 20
3610 20 20 20 20 20 20 1D 1C 1F 1F 3F 3F 1F 1F 1E 3F
3620 3F 1E 3F 1F 3F 3F 1F 3F 1F 1E 3F 1F 1E 1E 1F 1E
3630 20 20 20 20 20 20 20 20 20 1D 1C 1C 1C 1C 1E 1E
3640 1F 1F 1E 1F 1F 3F 1E 1F 1E 1F 3F 1E 3F 1E 3F 1E
3650 1F 3F 1E 1F 1E 1F 1E 1E 20 20 20 20 20 20 1D 1C
3660 1C 1E 3F 1E 1F 1E 3F 3F 1E 3F 1F 1E 3F 3F 1F 1E
3670 1F 1E 3F 3F 1E 1F 1F 1F 1F 1E 1F 1F 1E 1E 3F 1F
3680 20 20 20 1D 1C 1C 1F 1F 1F 3F 1E 1F 1F 3F 3F 1F
3690 1E 1F 3F 3F 1E 1E 1F 3F 1F 1E 1F 3F 3F 1F 1F 1F
36A0 3F 3F 1F 1E 3F 1F 1F 3F 20 1C 1C 1F 1F 3F 1F 3F
36B0 1E 1E 3F 3F 1E 1F 1F 1E 3F 1E 3F 1E 1F 1E 3F 1E
36C0 1E 1F 3F 3F 1F 3F 1F 1E 3F 1E 3F 3F 3F 1E 3F 3F
36D0 20 1F 3F 3F 3F 3F 1E 1F 3F 1F 1E 1F 1F 1F 1E 1E
36E0 1E 3F 3F 1E 1F 3F 3F 1E 3F 1F 3F 1F 1F 3F 1E 3F
36F0 3F 1F 1E 1F 1E 3F 1F 3F 20 3F 3F 1E 1F 1F 3F 3F
3700 1E 1F 3F 3F 1F 1F 1E 1F 3F 1F 3F 3F 1F 3F 1E 1F
3710 1F 1F 1F 3F 1F 3F 1E 1E 1F 3F 1F 1F 1E 1E 1F 3F
3720 20 3F 3F 1F 3F 1F 3F 1E 1F 1F 3F 3F 1F 1F 1F 1F
```

```
3730 1F 1F 1E 1E 3F 3F 1E 3F 1E 1F 3F 1F 1E 1F 1E 1F
3740 1F 3F 3F 1E 3F 1E 3F 1F 20 1F 1E 3F 1F 1E 3F 1F
3750 1F 1E 3F 3F 1F 1F 3F 1F 1E 1F 1F 1F 1F 1F 3F 1F
3760 3F 1F 3F 1F 1E 1F 1F 1F 1F 3F 3F 1F 3F 1E 3F 3F
3770 20 1F 1F 1E 3F 1F 1E 1E 3F 3F 3F 3F 3F 1E 1F 1F
3780 1F 1E 1F 1E 3F 1F 1E 1E 3F 3F 1F 1E 1F 3F 3F 1E
3790 1F 1F 3F 1F 1E 3F 1F 3F 20 1F 1E 3F 1F 1E 1F 3F
37A0 1F 3F 1F 1F 1E 1F 1E 1E 1E 1E 1F 3F 3F 3F 1F 3F
37B0 1F 1E 1E 1E 1F 3F 3F 1E 1F 1F 3F 1F 1F 1E 1F 1F
37C0 20 1F 1E 1E 1F 1F 3F 3F 3F 3F 3F 1F 1F 3F 3F 1F
37D0 1F 3F 3F 1E 1F 1F 1E 3F 1E 3F 3F 3F 1E 1F 1F 3F
37E0 1F 1E 3F 1E 1E 1E 1E 1F

3980 00 00 00 00 00 00 00 00 00 00 00 00 00 07 80 00
3990 07 40 00 07 80 00 07 80 00 0F C0 00 0B 40 00 0B
39A0 40 00 0B 40 00 0C C0 00 07 80 00 03 00 00 03 00
39B0 00 03 00 00 03 00 00 03 00 00 03 80 00 03 C0 00
39C0 00 00 00 00 00 00 00 00 00 00 00 00 00 07 80 00
39D0 07 40 00 07 80 00 07 80 00 0F C0 00 0B 40 00 06
39E0 60 00 1C 70 00 38 D8 00 07 80 00 03 80 00 03 C0
39F0 00 1F 60 00 1F 60 00 18 38 00 10 3C 00 00 00 00
3A00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
3A10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
3A20 00 00 1E 00 00 3F 00 00 5E 80 00 ED C0 00 F3 C0
3A30 00 F3 C0 00 ED C0 00 5E 80 00 3F 00 00 1E 00 00
3A40 00 00 00 00 00 00 00 0F C0 00 F5 60 0F AA A0 F5
3A50 55 60 FF AA E0 FF F5 E0 FF FF E0 8F FF E0 80 FF
3A60 E0 F0 0F A0 FF 00 20 FF F0 60 FF FF E0 0F FF E0
3A70 00 FF C0 00 0F 80 00 00 00 00 00 00 00 00 00 00
3A80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
3A90 18 00 00 3C 00 00 66 00 00 3C 00 00 78 00 00 78
3AA0 00 00 78 00 00 3C 00 00 1E 00 00 1E 00 00 1E 00
3AB0 00 3C 00 00 78 00 00 78 00 00 30 00 00 20 00 00
3AC0 00 24 00 00 18 00 00 08 00 00 10 00 00 08 00 00
3AD0 18 00 00 3C 00 00 66 00 00 3C 00 00 78 00 00 78
3AE0 00 00 78 00 00 3C 00 00 1E 00 00 1E 00 00 1E 00
3AF0 00 3C 00 00 78 00 00 78 00 00 30 00 00 20 00 00
3B00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
3B10 00 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF
3B20 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00
3B30 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00
3B40 00 00 00 00 00 00 00 00 00 00 C0 00 01 20 C0 02
3B50 1B 30 1A 04 08 24 08 08 42 08 10 82 04 60 84 00
3B60 90 80 00 08 40 00 08 44 2B 30 38 41 40 08 40 80
3B70 07 23 00 00 DC 00 00 00 00 00 00 00 00 00 00 00
4000 EA EA EA A9 09 8D 20 D0 A9 03 8D 21 D0 A9 00 85
4010 FB A9 41 85 FC A0 00 EA B1 FB F0 0E 20 D2 FF EA
4020 E6 FB D0 02 E6 FC 18 90 EF EA EA EA EA EA EA EA
4030 EA EA EA EA EA A0 FF C8 BE 20 43 C8 B9 20 43 F0
4040 18 C8 84 FB A8 20 F0 FF A4 FB B9 20 43 F0 E8 C8
4050 20 D2 FF 18 90 F4 EA EA EA A9 13 20 96 40 EA EA
4060 A9 09 8D 20 D0 A9 0F 8D 21 D0 A9 A0 85 FB A9 43
4070 85 FC A0 00 EA B1 FB F0 0E 20 D2 FF EA E6 FB D0
4080 02 E6 FC 18 90 EF EA 20 E4 FF C9 53 D0 F9 60 00
4090     (6 unused bytes)     C9 13 90 03 A9 13 EA EA EA EA
40A0 85 FB A0 1C A9 00 99 00 D4 88 D0 FA EA A9 0F 8D
40B0 18 D4 EA A9 00 8D 05 D4 A9 F0 8D 06 D4 A0 00 B9
40C0 60 43 8D 01 D4 C8 B9 60 43 8D 00 D4 C8 84 FC A9
40D0 00 AA A8 20 DB FF A4 FC EA B9 60 43 C8 84 FC 85
40E0 FD EA A9 11 8D 04 D4 EA 20 DE FF C5 FD D0 F9 EA
40F0 A4 FC A9 00 8D 04 D4 EA C6 FB D0 C3 8D 18 D4 60
```

```
41Ø0 93 95 8E A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9
411Ø A9 A9 A9 A9 A9 A9 A9 8E ØD 95 A9 A9 A9 A9 A9 A9
412Ø A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 8E 9Ø 7D
413Ø 2Ø 2Ø 2Ø 2Ø 2Ø 57 52 49 54 54 45 4E 2Ø 42 59 ØD
414Ø 95 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9
415Ø A9 A9 A9 9Ø 2Ø 7D 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 41 4E 47
416Ø 55 53 2Ø 41 47 45 52 95 ØD 95 A9 A9 A9 A9 A9 A9
417Ø A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 8E 9Ø 2Ø 2Ø 7D
418Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 4B 45 4E 2Ø 54 49 4E 44 45
419Ø 4C 4C ØD 95 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9
41AØ A9 A9 A9 A9 9Ø 2Ø 2Ø 2Ø 7D 2Ø 2Ø 2Ø 2Ø 2Ø 44 45
41BØ 53 49 47 4E 45 44 2Ø 42 59 ØD 95 A9 A9 A9 A9 A9
```

```
41CØ A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 8E 9Ø 2Ø 2Ø 2Ø 2Ø
41DØ 7D 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 41 4C 41 53 54 41 49 52
41EØ 2Ø 44 4F 45 ØD 95 A9 A9 A9 A9 A9 A9 A9 A9 A9 A9
41FØ A9 A9 A9 A9 9Ø 2Ø 2Ø 2Ø 2Ø 2Ø 7D 95 ØD A9 A9 A9
42ØØ A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 ØD A9 A9 A9 A9 A9
421Ø A9 A9 A9 A9 A9 A9 A9 ØD A9 A9 A9 A9 A9 A9 A9 A9
422Ø A9 A9 A9 ØD A9 A9 A9 A9 A9 A9 A9 A9 A9 A9 ØD A9
423Ø A9 A9 A9 A9 A9 A9 A9 A9 ØD A9 A9 A9 A9 A9 A9 A9
424Ø A9 ØD A9 A9 A9 A9 A9 A9 A9 ØD A9 A9 A9 A9 A9 A9
425Ø ØD A9 A9 A9 A9 A9 ØD A9 A9 A9 A9 ØD A9 A9 A9 ØD
426Ø A9 A9 ØD A9 ØD 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 8E 1F B2
427Ø 2Ø B2 2Ø 75 63 69 2Ø 75 69 B2 2Ø 75 63 69 2Ø 75
428Ø 63 69 2Ø BØ 63 69 ØD 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 7D
429Ø 2Ø 7D 2Ø 7D 2Ø 7D 2Ø 7D 7D 7D 2Ø 7D 2Ø 7D 2Ø 7D
42AØ 2Ø 2Ø 2Ø 7D 2Ø 7D ØD 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø AB
42BØ 63 B3 2Ø AB 63 B3 2Ø 7D 7D 7D 2Ø 7D 2Ø 2Ø 2Ø AB
42CØ B3 2Ø 2Ø AB B2 6B ØD 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 7D
42DØ 2Ø 7D 2Ø 7D 2Ø 7D 2Ø 7D 7D 7D 2Ø 7D 2Ø B2 2Ø 7D
42EØ 2Ø 2Ø 2Ø 7D 7D ØD 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø B1 2Ø
42FØ B1 2Ø B1 2Ø B1 2Ø B1 6A 6B 2Ø 6A 63 6B 2Ø 6A 63
43ØØ 6B 2Ø B1 CA CB ØØ
```

```
432Ø Ø8 15 1F 75 69 ØØ Ø9 15 62 ØØ ØA 15 6A 6B B2 ØØ
433Ø ØB 17 62 ØØ ØC 17 AD BD B2 ØØ ØD 19 62 ØØ ØE 19
434Ø B1 BØ AE ØØ ØF 1A AB ØØ 1Ø 1A B1 BØ AE ØØ 11 1B
435Ø AB ØØ 12 1B B1 ØØ ØØ ØØ FF ØØ FF FF ØØ ØØ FF 5F
436Ø 1C 7E 1Ø 2D C6 2Ø 33 61 1Ø 39 AC 4Ø 3D 7E 1Ø 39
437Ø AC 2Ø 33 61 2Ø 2B 34 1Ø 22 4B 4Ø 26 7E 1Ø 2B 34
438Ø 2Ø 2D C6 2Ø 26 7E 1Ø 26 7E 4Ø 22 4B 1Ø 26 7E 2Ø
439Ø 2B 34 2Ø 24 55 1Ø 1C D6 2Ø ØØ FF FF ØØ ØØ FF FF
43AØ 93 ØD ØD ØD 1F 54 48 45 2Ø 53 54 4F 52 59 2Ø 53
43BØ 4F 2Ø 46 41 52 2Ø 2E 2Ø 2E 2Ø 2E 2Ø 2E ØD A3 A3
43CØ A3 A3 A3 A3 A3 A3 A3 A3 A3 A3 A3 A3 A3 A3 A3 ØD
43DØ ØD ØD 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 41 46 54 45 52 2Ø 41 2Ø
43EØ 53 48 4F 52 54 2Ø 57 41 4C 4B 2Ø 57 49 4C 4C 59
43FØ 2Ø 52 45 54 55 52 4E 53 2Ø 2Ø 54 4F 2Ø 46 49 4E
44ØØ 44 2Ø 54 48 45 2Ø 47 4F 41 54 53 2Ø 48 41 56 45
441Ø 2Ø 53 5Ø 52 45 41 44 2Ø 48 49 53 2Ø 5Ø 49 43 4E
442Ø 49 43 47 4F 4F 44 49 45 53 2Ø 41 4C 4C 2Ø 4F 56
443Ø 45 52 2Ø 41 2Ø 52 4F 43 4B 59 2Ø 45 4D 42 41 4E
444Ø 4B 4D 45 4E 54 2E 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
445Ø 57 49 4C 4C 59 2Ø 53 45 54 53 2Ø 4F 46 46 2Ø 54
446Ø 4F 2Ø 52 45 43 4C 41 49 4D 2Ø 48 49 53 2Ø 4C 4F
447Ø 53 45 5Ø 4F 43 45 53 53 49 4F 4E 53 2C 2Ø 42 55
448Ø 54 2Ø 49 53 2Ø 48 41 4D 5Ø 45 52 45 44 2Ø 42 59
449Ø 2Ø 46 41 4C 4C 49 4E 47 2Ø 2Ø 42 55 4F 55 4C 44
44AØ 45 52 2C 2Ø 5Ø 4F 54 48 4F 4C 45 53 2Ø 41 4E 44
44BØ 2Ø 56 49 43 49 4F 55 53 2Ø 53 4E 41 4B 45 53 2E
44CØ 2Ø 2Ø 54 4F 2Ø 4D 41 4B 45 2Ø 4D 41 54 54 45 52
44DØ 53 2Ø 57 4F 52 53 45 2Ø 54 48 45 2Ø 54 49 44 45
44EØ 2Ø 49 53 2Ø 52 41 53 49 4E 47 41 4E 44 2Ø 48 45
44FØ 2Ø 43 41 4E 4E 4F 54 2Ø 53 57 49 4D 2E 2Ø 2Ø 2Ø
45ØØ 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
451Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 54 4F 2Ø 48 45 4C 5Ø 2Ø
452Ø 57 49 4C 4C 59 2Ø 4F 4E 2Ø 48 49 53 2Ø 51 55 45
453Ø 53 54 2Ø 52 45 41 44 2Ø 2Ø 2Ø 54 48 45 2Ø 46 4F
454Ø 4C 4C 4F 57 49 4E 47 2Ø 43 4F 4E 54 52 4F 4C 2Ø
455Ø 49 4E 53 54 52 55 43 54 49 4F 4E 53 2Ø 41 4E 44
456Ø 2Ø 2Ø 5Ø 52 45 53 53 2Ø 2Ø 53 2Ø 2Ø 54 4F 2Ø 53
457Ø 54 41 52 54 2E ØD ØD ØD 95 2Ø 2Ø 2Ø 2Ø 2Ø 4B 45
458Ø 59 42 4F 41 52 44 2Ø 2D 2Ø 53 48 49 46 54 3D 4D
459Ø 4F 56 45 2Ø 52 49 47 48 54 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
45AØ 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
45BØ 2Ø 2Ø 5A 2Ø 2Ø 2Ø 3D 4A 55 4D 5Ø 2Ø 56 45 52 54
45CØ 49 43 41 4C 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
45DØ 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 42 4F 54 48 2Ø 3D 4A
45EØ 55 4D 5Ø 2Ø 52 49 47 48 54 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
45FØ 2Ø ØD 2Ø 2Ø 2Ø 2Ø 2Ø 4A 4F 59 53 54 49 43 4B 2Ø
46ØØ 2D 2Ø 49 4E 53 45 52 54 2Ø 49 4E 2Ø 5Ø 4F 52 54
461Ø 2Ø 32 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
```

```
5ØØØ AD 53 Ø3 8D 84 Ø3 A9 ØØ 8D 85 Ø3 18 A2 Ø5 18 2E
5Ø1Ø 84 Ø3 2E 85 Ø3 CA DØ F6 AD 53 Ø3 8D 86 Ø3 A9 ØØ
5Ø2Ø 8D 87 Ø3 A2 Ø3 18 2E 86 Ø3 2E 87 Ø3 CA DØ F6 18
5Ø3Ø AD 84 Ø3 6D 86 Ø3 8D 84 Ø3 AD 85 Ø3 6D 87 Ø3 18
5Ø4Ø 69 Ø4 8D 85 Ø3 18 AD 52 Ø3 6D 84 Ø3 8D 84 Ø3 AD
5Ø5Ø 85 Ø3 69 ØØ 8D 85 Ø3 AD 84 Ø3 85 FB AD 85 Ø3 85
5Ø6Ø FC AØ ØØ 6Ø
```

```
51ØØ AD Ø8 CØ 18 6A 8D 52 Ø3 AD Ø9 CØ 18 6A 8D 53 Ø3
511Ø 2Ø ØØ 5Ø B1 FB 6Ø
```

```
515Ø AØ ØØ B1 FB 8D 84 Ø3 A5 FC 18 69 D4 85 FC AD 84
516Ø Ø3 C9 1C FØ 1C C9 1D FØ 18 C9 3F FØ 19 C9 1E FØ
517Ø 15 C9 1F FØ 11 A9 Ø2 91 FB A5 FC 38 E9 D4 85 FC
518Ø 6Ø A9 Ø5 4C 77 51 A9 Ø1 4C 77 51
52ØØ A9 ØØ 8D 84 Ø3 AD ØØ DC A8 29 1Ø FØ 13 AD 8D Ø2
521Ø C9 Ø1 FØ ØC 98 29 Ø8 FØ ØF A5 C5 C9 ØC FØ Ø9 6Ø
522Ø A9 Ø2 8D 84 Ø3 4C 14 52 AD 84 Ø3 Ø9 Ø1 8D 84 Ø3
523Ø 6Ø
```

```
53ØØ AD Ø1 DØ 38 E9 Ø4 8D Ø1 DØ CE 11 CØ AD ØØ DØ 18
531Ø 69 Ø4 8D ØØ DØ A9 ØØ 2A 8D 84 Ø3 AD 1Ø DØ 4D 84
532Ø Ø3 8D 1Ø DØ EE 12 CØ 6Ø
```

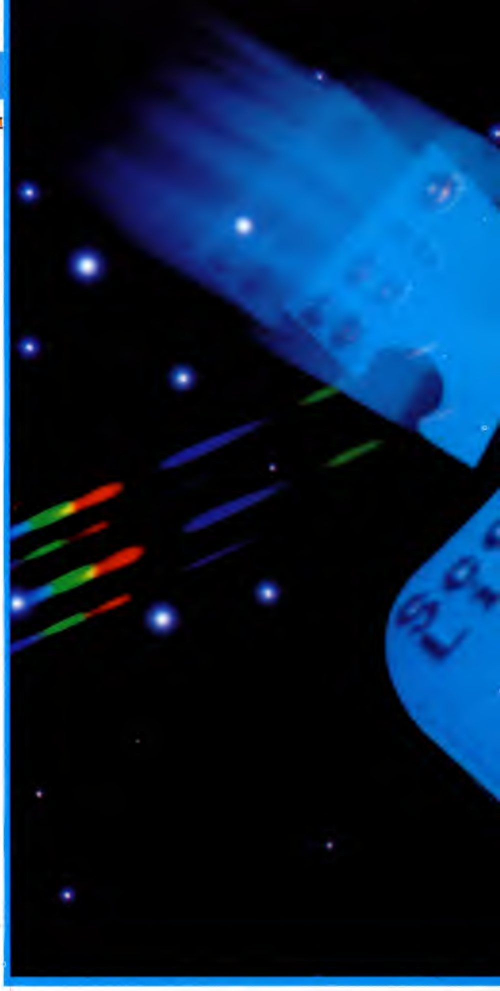
```
535Ø AD 12 CØ 18 6A 8D 52 Ø3 AD 11 CØ 18 6A 8D 53 Ø3
536Ø 2Ø ØØ 5Ø AØ ØØ B1 FB 6Ø
```

```
54ØØ EE 11 CØ AD Ø1 DØ 18 69 Ø4 8D Ø1 DØ 6Ø
```

```
545Ø CE 11 CØ AD Ø1 DØ 38 E9 Ø4 8D Ø1 DØ 6Ø
```

```
56ØØ AD ØØ CØ C9 Ø1 FØ 1C C9 Ø2 FØ 18 C9 Ø3 FØ 14 A2
561Ø ØØ AØ Ø3 DE 64 C3 FØ ØC BD 64 C3 C9 ØA FØ 12 E8
562Ø 88 DØ FØ 6Ø A9 EB 9D FB Ø7 A9 14 9D 64 C3 4C 1F
563Ø 56 A9 EA 9D FB Ø7 4C 1F 56
```

```
565Ø A2 ØØ AØ ØØ BD 5Ø C3 85 FB BD 5A C3 85 FC B1 FB
566Ø C9 3D DØ Ø4 A9 39 91 FB B1 FB 18 69 Ø1 91 FB E8
```



```
567Ø EØ Ø2 DØ EØ 6Ø
```

```
57ØØ AØ Ø8 A2 ØØ BD FØ 31 2A 3E FØ 31 E8 88 DØ F5 CE
571Ø ØC CØ FØ Ø1 6Ø AD Ø2 CØ 8D ØC CØ AØ ØØ A2 28 AD
572Ø ØD CØ 85 FB AD ØE CØ 85 FC 38 A5 FB E9 28 85 FB
573Ø A5 FC E9 ØØ 85 FC A5 FB 8D ØD CØ A5 FC 8D ØE CØ
574Ø 18 69 D4 85 FE A5 FB 85 FD A3 3E 91 FB A9 Ø6 91
575Ø FD C8 CA DØ F4 6Ø
```

```
58ØØ AD ØB CØ FØ 1Ø EE Ø4 DØ AD Ø4 DØ C9 FF DØ Ø5 A9
581Ø ØØ 8D ØB CØ 6Ø CE Ø4 DØ AD Ø4 DØ C9 3C DØ F5 A9
582Ø Ø1 8D ØB CØ 6Ø
```

```
585Ø A9 38 8D Ø2 DØ AD 1Ø DØ Ø9 Ø2 8D 1Ø DØ A9 51 8D
586Ø Ø3 DØ A9 48 8D Ø8 CØ A9 ØD 8D Ø9 CØ 6Ø
```

```
59ØØ AD 15 DØ 29 Ø2 FØ 33 CE Ø8 CØ AD Ø8 CØ C9 39 FØ
591Ø 2E CE Ø2 DØ CE Ø2 DØ CE Ø2 DØ CE Ø2 DØ 2Ø ØØ 51
592Ø C9 2Ø DØ ØF EE Ø9 CØ EE Ø3 DØ EE Ø3 DØ EE Ø3 DØ
593Ø EE Ø3 DØ AD Ø8 CØ C9 Ø4 FØ Ø1 6Ø 2Ø 5Ø 58 6Ø A9
594Ø FC 8D Ø2 DØ AD 1Ø DØ 29 FD 8D 1Ø DØ 4C 11 59
```

```
6ØØØ 4C 59 6Ø AD ØØ CØ C9 Ø1 DØ Ø6 A9 47 8D 15 DØ 6Ø
6Ø1Ø C9 Ø2 DØ 18 A9 7D 8D 15 DØ A2 Ø3 AØ ØØ A9 ØF 99
6Ø2Ø 2A DØ A9 EC 99 FB Ø7 C8 CA DØ F2 6Ø C9 Ø3 DØ Ø8
6Ø3Ø A9 7F 8D 15 DØ 4C 19 6Ø C9 Ø4 DØ 18 A9 7D 8D 15
6Ø4Ø DØ A2 Ø3 AØ ØØ A9 Ø5 99 2A DØ A9 EA 99 FB Ø7 C8
```

```
6Ø5Ø CA DØ F2 6Ø A9 7F 4C 3E 6Ø A9 Ø7 8D ØE CØ A9 E8
6Ø6Ø 8D ØD CØ AD Ø2 CØ 8D Ø2 CØ 8D ØC CØ A9 Ø2 8D 12
6Ø7Ø CØ A9 21 8D 11 CØ A9 12 8D ØØ DØ A9 A1 8D Ø1 DØ
```

```
6080 A9 42 8D 10 D0 A9 4A 8D 0D A9 40 8D 0C D0 AD
6090 1E D0 AD 1F D0 4C 03 60.

6100 A9 74 8D 06 D0 A9 92 8D 07 D0 A9 C0 8D 08 D0 A9
6110 7A 8D 09 D0 A9 A0 8D 0A D0 A9 82 8D 0B D0 A2 03
6120 A0 00 A9 14 99 64 C3 E9 05 C8 CA D0 F7 60
6150 A9 01 8D 0B C0 A9 32 8D 04 D0 A9 46 8D 05 D0 60
6200 A9 E6 8D F8 07 AD 05 C0 F0 0C A9 E7 8D F8 07 20
6210 00 53 CE 05 C0 60 AD 06 C0 F0 0C A9 E6 8D F8 07
6220 20 50 54 CE 06 C0 60 20 50 53 C9 20 D0 04 20 00
6230 54 60 20 00 52 AD 84 03 C9 03 F0 2F C9 02 D0 0B
6240 20 50 54 A9 04 8D 06 C0 4C 73 62 C9 01 D0 2F 20
6250 50 53 A5 FB 38 E9 27 85 FB A5 FC E9 00 85 FC A0
6260 00 B1 FB C9 20 D0 17 20 7F 62 60 20 50 54 A9 04
6270 8D 05 C0 A9 02 8D 84 03 20 50 68 20 00 68 60 AD
6280 00 D0 18 69 04 8D 00 D0 A9 00 2A 8D 84 03 AD 10
6290 D0 4D 84 03 8D 10 D0 EE 12 C0 A9 01 8D 84 03 20
62A0 50 68 20 00 68 60

6300 A9 00 85 FB A9 04 85 FC A9 34 85 FE A9 00 85 FD
6310 A0 00 B1 FD 91 FB 20 50 51 E6 FB D0 02 E6 FC E6
6320 FD D0 02 E6 FE A5 FB C9 E8 D0 E7 A5 FC C9 07 D0
6330 E1 AE 01 C0 A0 00 A9 1B 99 2E 04 C8 CA D0 F9 AD
6340 00 C0 C9 06 D0 0B A9 01 8D 00 C0 EE 01 C0 CE 02
6350 C0 AE 00 C0 A0 00 A9 09 99 56 04 C8 CA D0 F9 60
6400 A9 00 8D 15 D0 8D 05 C0 8D 06 C0 8D 0C C0 8D 1E
6410 D0 8D 1F D0 AD 18 D0 29 F0 09 0C 8D 18 D0 60
6450 AD 1E D0 8D 5C 03 AD 1F D0 8D 5D 03 AD 5C 03 29
6460 40 F0 0E A9 04 8D 84 03 20 50 68 EE 00 C0 A9 9B
6470 60 AD 5C 03 29 01 F0 0E A9 03 8D 84 03 20 50 68
6480 CE 01 C0 A9 FE 60 AD 5D 03 29 01 F0 0B A9 03 8D
6490 84 03 20 50 68 4C 80 64 A9 00 60

6500 A9 01 8D 00 C0 A9 05 8D 01 C0 A9 28 8D 0C C0 8D
6510 02 C0 A9 0F 8D 20 D0 8D 21 D0 A9 0A 8D 27 D0 A9
6520 02 8D 28 D0 A9 01 8D 29 D0 A9 05 8D 2A D0 8D 2B
6530 D0 8D 2C D0 A9 00 8D 04 D4 8D 0B D4 8D 12 D4 A9
6540 0F 8D 18 D4 A9 43 8D 50 C3 A9 04 8D 5A C3 A9 70
6550 8D 51 C3 A9 04 8D 5B C3 A9 E6 8D F8 07 A9 E8 8D
6560 F9 07 A9 ED 8D FA 07 A9 EA 8D FB 07 8D FC 07 8D
6570 FD 07 A9 E9 8D FE 07 60
6600 20 00 59 20 00 58 20 00 57 20 50 56 20 00 56 60
6700 20 00 64 20 00 63 20 50 58 20 50 61 20 00 61 20
6710 00 60 60

6750 20 00 40 20 00 65 20 00 63 20 A9 67 A9 00 8D 15
6760 D0 AD 1E D0 AD 1F D0 A9 38 8D 17 D0 20 00 67 20
6770 B8 67 AD 1E D0 AD 1F D0 20 00 66 20 00 62 AE 02
6780 C0 A0 FF 88 D0 FD CA D0 F8 20 50 64 F0 EA 20 A9
6790 67 A9 00 8D 15 D0 AD 1E D0 AD 1F D0 AD 01 C0 D0
67A0 BB A9 15 8D 18 D0 4C 50 67 A2 00 A0 06 BD 7E 04
67B0 9D 82 C3 E8 88 D0 F6 60 A2 00 A0 06 BD 82 C3 9D
67C0 7E 04 E8 88 D0 F6 60

6800 A2 05 BD 7E 04 C9 39 F0 04 FE 7E 04 60 A9 30 9D
6810 7E 04 CA 4C 02 60

6850 A9 0F 8D 18 D4 A9 1E 8D 01 D4 A9 00 8D 06 D4 AD
6860 84 03 C9 01 D0 08 A9 01 8D 05 D4 4C 7A 68 AD 84
6870 03 C9 02 D0 10 A9 55 8D 05 D4 A9 00 8D 04 D4 A9
6880 21 8D 04 D4 60 A2 1E A9 00 8D 04 D4 A9 81 8D 04
6890 D4 A9 F0 8D 06 D4 AD 84 03 C9 03 F0 0B E8 E0 32
68A0 D0 0B A9 00 8D 04 D4 60 CA E0 0A F0 F5 8E 01 D4
68B0 A9 5A A0 FF 88 D0 FD 18 E9 01 D0 F6 4C 96 68
```

Use your machine code monitor to PEEK your BBC or Electron's memory and check that it tallies with the following hex dump. If you have an Electron you will find some slight differences in the hex dump due to changes made to speed it up.

```
0D00 16 02 17 00 0A 20 00 00 00 00 00 00 00 00 05 04
0D10 1E E3 05 10 FF 05 02 E3 05 02 97 05 10 7B 05 1E
0D20 97 04 20 FF 05 20 AF 05 24 A7 05 28 AF 04 28 D3
0D30 05 28 A7 05 3C E9 05 3C F5 05 37 FF 05 32 F5 05
0D40 32 85 05 2D 7B 05 28 85 05 28 91 05 4A E9 05 4A
0D50 F5 05 45 FF 05 40 F5 05 40 85 05 3B 7B 05 36 85
0D60 05 36 91 05 56 EB 05 56 F7 05 52 FF 05 4E F7 05
0D70 4E A7 04 4E C7 05 54 D3 05 5A C7 05 5A A7 04 68
0D80 C7 05 62 D3 05 5C C7 05 5C B3 05 62 A7 05 68 B3
0D90 04 68 D3 05 68 B3 05 6A A7 05 6A D3 04 6A C7 05
0DA0 70 D3 05 76 C7 05 76 A7 04 84 C7 05 7E D3 05 78
0DB0 C7 05 78 B3 05 7E A7 05 84 B3 04 84 D3 05 84 85
0DC0 05 7F 7B 05 7A 85 05 7A 97 05 92 C7 05 8C D3 05
0DD0 86 C7 05 86 B3 05 8C A7 05 94 B7 04 94 D3 05 94
0DE0 A7 04 94 C7 05 9A D3 05 A0 C7 45 28 D7 00 00 07
0DF0 04 A0 08 04 A0 6F A5 78 08 55 78 6F 04 6B 6D 55
0E00 78 56 00 00 02 04 A0 77 04 A0 6F 55 78 77 55 78
0E10 6F 55 66 73 55 67 6B 00 00 06 04 00 00 04 00 07
0E20 55 A0 00 55 A0 07 00 00 03 04 5C 6F 04 54 6F 55
0E30 62 7B 55 4E 7B 55 62 8B 55 4E 8B 55 5C 97 55 54
0E40 97 04 62 7F 04 62 87 55 65 7F 04 56 6E 04 56 6A
0E50 55 5A 6E 55 5A 6A 00 00 00 04 62 7B 05 5A 7B 00
0E60 00 01 45 5E 8A 00 00 04 04 4D 6E 04 54 6E 55 4E
0E70 8B 55 56 8B 55 50 8F 55 5C 8F 00 00 01 04 54 9B
0E80 04 4E 90 55 5C 9B 55 62 90 05 65 90 00 00 01 04
0E90 5A 2C 04 5F 2C 55 5A 1C 55 5F 1C 00 00 02 04 54
0EA0 6B 04 5C 6B 55 4C 5B 55 64 5B 55 4C 38 55 64 38
0EB0 55 54 28 55 5C 28 00 00 01 04 56 5E 04 59 58 55
0EC0 59 64 55 63 58 55 60 64 55 6A 68 55 65 70 00 00
0ED0 03 04 68 6D 04 6C 6D 55 69 74 00 00 01 04 56 3E
0EE0 04 56 34 55 5B 3E 55 60 2C 55 66 34 55 61 1C 55
0EF0 68 1C 00 00 04 04 62 1B 04 62 14 55 66 1B 55 6B
0F00 14 04 5B 1B 04 5B 14 55 5E 1B 55 63 14 00 00 04
0F10 04 06 24 04 06 2C 55 0E 08 55 17 14 55 42 08 55
0F20 1C 14 04 42 08 55 34 28 55 40 20 55 38 28 00 00
0F30 00 04 42 10 04 42 1E 55 36 14 04 31 1F 04 31 18
0F40 55 34 IF 55 34 18 00 00 07 04 31 1F 05 31 18 05
0F50 34 18 04 42 11 1D 37 14 1D 40 1C A0 00 B9 00 0D
0F60 20 EE FF C8 C0 0C D0 F5 A2 00 86 70 A2 0D 86 71
0F70 B1 70 F0 45 AA A9 19 20 EE FF 8A 20 EE FF 20 D1
0F80 0F B1 70 0A 0A 0A 20 EE FF B1 70 4A 4A 4A 4A 4A
0F90 20 EE FF 20 D1 0F B1 70 0A 0A 20 EE FF B1 70 4A
0FA0 4A 4A 4A 4A 4A 20 EE FF 20 D1 0F C0 5B D0 07 A5
0FB0 71 C9 0F D0 01 60 4C 70 0F A9 12 20 EE FF 20 D1
0FC0 0F B1 70 20 EE FF 20 D1 0F B1 70 20 EE FF 4C A8
0FD0 0F C8 D0 02 E6 71 60 89 0A 95 94 9D 0A A5 0F A9
0FE0 05 A5 0A 9D 94 91 0A 81 0F 89 05 91 0A 95 94 89
0FF0 0A 89 0F 85 05 89 0A 91 94 85 0A 75 14 89 0A 95
1000 94 9D 0A A5 0F A9 05 A5 0A 9D 94 91 0A 81 0F 89
1010 05 91 0A 95 8F 91 05 89 0A 85 0F 7D 05 85 0A 89
```

```
1020 9E 89 14 B1 9E B1 0F A9 05 A5 0A 9D 94 91 0A 81
1030 0F 89 05 91 0A 95 94 89 0A 89 0F 85 05 89 0A 91
1040 94 85 0A 75 1E B1 9E B1 0F A9 05 A5 0A 9D 94 91
1050 0A 81 0F 89 05 91 0A 95 8F 91 05 89 0A 85 0F 7D
1060 05 85 0A 89 9E 89 1E 89 9E 95 1E 81 9E 91 1E 89
1070 9E 79 1E 75 8A A5 0A 91 0A 75 1E 89 9E 95 1E 81
1080 9E 91 1E 89 8F 91 05 95 05 9D 05 A5 14 75 0A 89
1090 8A B9 0A A5 0A 89 14 95 8F 9D 05 A5 0A A5 0F 9D
10A0 05 95 0A 81 8F 89 05 91 0A 91 0F 89 05 81 0A 89
10B0 8F 91 05 95 0A 95 0F 91 05 89 0A 75 8A A5 0A 91
10C0 0A 75 14 6D 0A 65 8F 6D 05 75 0A 75 0F 6D 05 65
10D0 0A 81 8F 89 05 91 0A 91 0F 89 05 81 0A 89 8F 91
10E0 05 95 05 9D 05 A5 14 75 0A 89 8A B9 0A A5 0A 89
10F0 1E 02 00 02 00 91 00 0A 00 00 00 00 00 00 00 00
1100 A5 80 29 80 F0 01 60 A5 80 29 04 F0 0E A5 80 29
1110 FB 09 03 85 80 A9 00 85 81 85 82 A5 80 29 01 F0
1120 57 A9 80 A2 F9 A0 FF 20 F4 FF 8A F0 4B A9 02 8D
1130 F1 10 8D F3 10 A9 00 8D F2 10 8D F4 10 8D F6 10
1140 8D F8 10 A6 81 BD D7 0F 8D F5 10 BD D8 0F 29 80
1150 F0 05 A9 01 8D F2 10 BD D8 0F 29 7F 8D F7 10 A9
1160 07 A2 F1 A0 10 20 F1 FF A6 81 E8 E8 86 81 E0 90
1170 D0 06 A5 80 29 FE 85 80 A5 80 29 02 F0 57 A9 80
1180 A2 F8 A0 FF 20 F4 FF 8A F0 4B A9 03 8D F1 10 8D
1190 F3 10 A9 00 8D F2 10 8D F4 10 8D F6 10 8D F8 10
11A0 A6 82 BD 67 10 8D F5 10 BD 68 10 29 80 F0 05 A9
11B0 01 8D F2 10 BD 68 10 29 7F 8D F7 10 A9 07 A2 F1
11C0 A0 10 20 F1 FF A6 82 E8 E8 86 82 E0 8A D0 06 A5
11D0 80 29 FD 85 80 60 06 01 42 42 43 20 43 6C 69 66
11E0 66 68 61 6E 67 65 72 20 42 79 20 44 2E 53 75 6D
11F0 6D 65 72 73 0D 06 02 2A 2A 2A 2A 2A 2A 2A 2A 2A
1200 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
1210 2A 2A 2A 0D 04 04 41 66 74 65 72 20 61 20 73 68
1220 6F 72 74 20 77 61 6C 6B 20 57 69 6C 6C 79 20 20
1230 72 65 74 75 72 6E 73 20 74 6F 0D 00 05 66 69 6E
1240 64 20 20 74 68 65 20 67 6F 61 74 73 20 20 68 61
1250 76 65 20 73 70 72 65 61 64 20 20 68 69 73 20 70
1260 69 63 6E 69 63 0D 00 06 67 6F 64 64 69 65 73 20
1270 20 61 6C 6C 20 20 6F 76 65 72 20 20 61 20 72 6F
1280 63 6B 79 20 20 65 6D 62 61 6E 6B 6D 65 6E 74 2E
1290 0D 00 07 57 69 6C 6C 79 20 20 20 73 65 74 73 20
12A0 20 6F 66 20 20 74 6F 20 20 72 65 63 6C 61 69 6D
12B0 20 20 68 69 73 20 20 6C 6F 73 74 0D 00 08 70 6F
12C0 73 73 65 73 73 69 6F 6E 73 20 2C 62 75 74 20 66
12D0 69 6E 64 73 20 68 65 20 69 73 20 68 61 6D 70 65
12E0 72 65 64 20 62 79 0D 00 09 66 61 6C 6C 69 6E 67
12F0 20 20 62 6F 75 6C 64 65 72 73 20 2C 70 6F 74 68
1300 6F 6C 65 73 20 20 61 6E 64 20 76 69 63 69 6F 75
1310 73 0D 00 0A 73 6E 61 6B 65 73 2E 54 6F 20 6D 61
1320 6B 65 20 6D 61 74 74 65 72 73 20 77 6F 72 73 65
1330 20 74 68 65 20 74 69 64 65 20 69 73 0D 00 0B 72
1340 69 73 69 6E 67 20 61 6E 64 20 68 65 20 69 73 20
1350 69 6E 20 20 64 61 6E 67 65 72 20 6F 66 20 62 65
1360 69 6E 67 20 63 75 74 0D 00 0C 6F 66 66 20 2E 54
1370 6F 20 68 65 6C 70 20 57 69 6C 6C 79 20 69 6E 20
1380 68 69 73 20 71 75 65 73 74 20 72 65 61 64 20 74
1390 68 65 0D 00 0D 66 6F 6C 6C 6F 77 69 6E 67 20 69
13A0 6E 73 74 72 75 63 74 69 6F 6E 73 20 3A 2D 0D 0A
13B0 10 4E 20 20 20 2D 20 20 20 4C 65 66 74 0D 0A 12
13C0 4D 20 20 20 2D 20 20 20 52 69 67 68 74 0D 08 14
13D0 53 48 49 46 54 29 2D 20 20 20 4A 75 6D 70 0D 08
13E0 17 50 72 65 73 73 20 53 50 41 43 45 20 66 6F 72
```

```
13F0 20 73 74 61 72 74 0D 0D A9 16 20 EE FF A9 06 20
1400 EE FF 20 83 14 A9 D6 85 70 A9 11 85 71 A2 28 CA
1410 A9 1F 20 EE FF 8A 20 EE FF A0 01 B1 70 20 EE FF
1420 C8 B1 70 C9 0D F0 3E 20 EE FF 86 72 98 18 65 72
1430 C9 29 D0 EC A9 50 20 6E 14 84 72 A0 00 8A D1 70
1440 D0 CD A4 72 C8 C8 98 18 65 70 85 70 90 02 E6 71
1450 A0 00 B1 70 C9 0D D0 B5 A9 81 A0 FF A2 9D 20 F4
1460 FF 8A F0 F4 60 A9 20 20 EE FF 88 4C 34 14 85 72
1470 8A 48 98 48 A6 72 A0 FF 88 D0 FD CA D0 F8 68 A8
1480 68 AA 60 A2 02 BD 00 0D 20 EE FF E8 E0 0C D0 F5
1490 60 04 01 00 00 00 00 00 00 7E FF FF FF 7E 7E 04
14A0 01 00 00 00 00 00 00 02 FD 81 81 28 00 08 A2 91
14B0 A0 14 20 F1 FF A9 07 A2 A0 A0 14 20 F1 FF 60 10
14C0 00 04 00 04 00 0A 00 00 20 2B 15 A9 08 A2 9F A0
14D0 14 20 F1 FF A9 07 A2 BF A0 14 20 F1 FF 60 01 00
14E0 04 00 8C 00 14 00 00 20 2B 15 A9 08 A2 91 A0 14
14F0 20 F1 FF A9 07 A2 DE A0 14 20 F1 FF 60 01 02 FF
1500 00 00 FF 00 00 7E FF FF FF 7E 7E 00 01 00 01 00
1510 D2 00 37 00 00 20 2B 15 A9 08 A2 FD A0 14 20 F1
1520 FF A9 07 A2 0C A0 15 20 F1 FF 60 A5 80 29 80 F0
1530 02 68 68 60 FF FF FF FF FF FF FF FF 18 3C 3C 18
1540 00 00 00 00 00 00 00 00 3C 3C 3C 3C 3C 3C 00 00
1550 00 00 00 00 00 00 00 00 00 08 08 08 10 00 18 18
1560 18 18 18 38 00 00 00 00 00 08 08 78 03 00 18 24
1570 42 41 C2 00 3C 3C 18 18 18 18 18 38 03 03 01 02
1580 04 04 0C 00 42 41 C2 00 00 00 00 00 FF 3E 18 18
1590 18 18 18 38 FF 63 C2 00 00 00 00 00 18 3C 6E 7E
15A0 7E 3C 18 18 22 14 08 08 08 08 08 08 00 DD FF FF
15B0 FF FF FF FF 00 01 07 0F 1F 3F 3F 7F 00 80 E0 F0
15C0 F8 FC FC FE 1C 3E 7F FF FF FE 7C 38 38 7C FE FF
15D0 FF 7F 3E 1C FF 23 01 02 04 04 0C 00 00 78 86 01
15E0 01 00 00 00 00 1E 61 80 80 00 00 00 00 00 00 87
15F0 79 00 00 00 00 00 00 E1 9E 00 00 00 00 00 00 00
1600 00 10 10 10 08 00 18 18 18 18 18 1C 00 00 00 00
1610 00 10 10 1E C0 00 18 24 42 82 43 00 3C 3C 18 18
1620 18 18 18 1C 0F 03 01 02 04 08 04 00 42 82 43 00
1630 00 00 00 00 FF 3E 18 18 18 18 18 1C FF A2 43 00
1640 00 00 00 00 00 00 07 18 20 40 40 80 00 00 1F A0
1650 40 00 00 00 00 00 80 40 5C 22 02 02 80 40 3E 02
1660 02 01 00 00 00 00 00 04 0A 11 E0 00 02 04 08 04
1670 04 04 F8 00 00 00 00 00 81 81 81 83 C3 C3 C7 C7
1680 E7 EF EF EF 08 10 08 04 02 04 02 01 02 04 08 04
1690 08 10 20 10 20 40 80 40 20 40 20 40 20 10 20 10
16A0 00 10 00 10 10 08 04 08 04 02 01 02 04 02 04 08
16B0 10 08 10 20 40 20 40 80 40 20 40 20 10 20 10 20
16C0 10 00 10 00 00 00 FF FF FF 00 00 00 00 00 00 FF
16D0 FF FF 00 00 00 00 00 00 7F 7F 00 00 00 00 00
16E0 00 FE FE 00 00 7F 7F 00 00 00 00 00 00 FE FE 00
16F0 00 00 00 00 00 00 00 00 00 00 00 7F 00 00 00 00
1700 00 00 00 FE 7F 00 00 00 00 00 00 FE 00 00 00
1710 00 00 00 00 00 00 00 00 3C 7E FF FF FF FF FF FF
1720 7F 7F 3F 0F FF FF FF FF FE FE FC F0 00 00 00 01
1730 01 01 00 00 38 78 C0 80 80 80 00 00 18 18 18 18
1740 24 42 81 81 81 81 81 81 81 81 81 FF 00 7E 7E 7E
1750 7E 7E 7E 00 00 00 01 01 03 00 00 00 20 F8 FC FC
1760 FE 00 00 00 00 00 00 00 00 03 01 01 00 00 00 00
1770 00 FE FC FC F8 F8 F8 70 70 70 20 20 00 00 01 7F
1780 7F 7F 3C 80 00 07 FC F0 C0 00 00 00 80 C0 F0 70
1790 30 00 00 00 00 00 00 00 80 80 C0 70 00 00 00 00
17A0 00 03 0F 3F 70 33 0F 8F CC F0 70 30 FC F0 C0 00
17B0 00 00 00 00 00 00 00 00 00 00 03 0F 00 00 03 0F
```

```
17C0 3F FC F0 C0 0F 0C 00 00 03 0F 0F 0C 03 0F 3F FC
17D0 F0 C0 00 00 A9 17 20 EE FF 8A 20 EE FF A9 00 85
17E0 71 98 0A 26 71 0A 26 71 0A 26 71 18 69 34 85 70
17F0 A5 71 69 15 85 71 A0 00 B1 70 20 EE FF C8 C0 08
1800 D0 F6 60 85 72 8A 48 98 48 A5 72 29 80 D0 08 A5
1810 72 29 EE FF 4C 26 18 A5 72 29 7F A8 A2 FF 20 D4
1820 17 A9 FF 20 EE FF 68 A8 68 AA A5 72 60 A2 00 86
1830 72 A5 72 18 69 E0 AA A4 72 20 D4 17 E6 72 A6 72
1840 E0 17 D0 ED 60 06 01 05 00 03 07 04 06 02 01 05
1850 00 03 07 06 06 A2 00 A9 13 20 EE FF 8A 20 EE FF
1860 BD 45 18 20 EE FF A9 00 20 EE FF 20 EE FF 20 EE
1870 FF E8 E0 10 D0 E1 60 11 08 1F 13 06 44 61 24 61
1880 21 61 24 43 25 61 21 47 13 51 12 51 14 54 EF EF
1890 00 EF F1 F0 A2 00 BD 77 18 20 EE FF E8 E0 05 D0
18A0 F5 A2 00 86 70 BD 7C 18 4A 4A 4A 4A 48 A8 B9 8D
18B0 18 85 71 BD 7C 18 29 0F AA A5 71 20 EE FF A9 08
18C0 20 EE FF 8A 48 A9 86 20 F4 FF 84 72 A9 09 20 EE
18D0 FF A9 08 20 EE FF A9 0A 20 EE FF A9 E0 20 EE FF
18E0 C8 C0 1E D0 EC A9 1F 20 EE FF 8A 20 EE FF A5 72
18F0 20 EE FF 68 AA 68 48 29 01 F0 05 A9 09 20 EE FF
1900 68 48 29 02 F0 05 A9 08 29 EE FF 68 48 29 04 F0
1910 05 A9 0A 20 EE FF CA D0 A0 68 E6 70 A6 70 E0 12
1920 D0 83 A9 F1 20 EE FF 60 04 10 09 0D 10 0B 11 08
1930 20 08 0A 20 08 0A 20 08 0A A8 08 0A A8 A0 00 A9
1940 1F 20 EE FF B9 29 19 20 EE FF B9 29 19 20 EE FF
1950 C8 C8 A2 00 BD 2E 19 20 03 18 E8 E0 0F D0 F5 C0
1960 06 D0 DC 60 A9 19 20 EE FF A9 04 20 EE FF 8A 0A
1970 0A 0A 0A 0A 20 EE FF 8A 4A 4A 4A 20 EE FF 98 0A
1980 0A 0A 0A 38 E9 04 08 20 EE FF 98 4A 4A 4A 4A 28
1990 E9 00 20 EE FF 60 05 12 00 04 8D 08 0A AA 08 0A
19A0 AB 08 0A AC 08 0A AD 08 0B 0B 0B 12 00 02 AE 08
19B0 0A AF 08 0A B0 08 0A B1 04 20 3D 19 A0 00 98 48
19C0 B9 28 19 0A AA A9 20 38 F9 29 19 0A A8 20 64 19
19D0 68 A8 A2 00 BD 96 19 20 03 18 E8 E0 23 D0 F5 C8
19E0 C8 C0 06 D0 D9 60 1F 01 01 11 01 53 63 6F 72 65
19F0 3A 30 30 30 30 30 30 1F 01 02 4C 69 76 65 73 3A
1A00 20 20 20 20 20 35 1F 02 04 11 05 A2 A3 A4 0A 08
1A10 08 08 A5 A6 A7 1F 0A 05 A2 A3 A4 0A 08 08 08 A5
1A20 A6 A7 1F 05 04 11 03 95 96 1F 00 08 95 96 A2 00
1A30 BD E6 19 20 03 18 E8 E0 48 D0 F5 60 A9 1F 20 EE
1A40 FF A9 07 20 EE FF A9 01 20 EE FF A9 11 20 EE FF
1A50 A9 01 20 EE FF A2 00 B5 8A 18 69 30 20 EE FF E8
1A60 E0 06 D0 F3 A9 1F 20 EE FF A9 0C 20 EE FF A9 02
1A70 20 EE FF A5 89 18 69 30 20 EE FF 60 86 1A AB 1A
1A80 BE 1A CC 1A DC 1A 12 00 04 B2 B2 0A 08 08 B3 B3
1A90 0B 08 08 12 00 08 B4 B5 0A 08 08 B6 B7 0B 08 08
1AA0 12 00 02 B8 B9 0A 08 08 BA BB FF 12 00 01 BC BC
1AB0 0A 08 08 BD BE 0B 08 08 12 00 03 BF C0 FF 12 00
1AC0 03 09 C1 08 0A C2 08 12 00 01 C3 FF 12 00 04 C4
1AD0 C5 08 08 12 00 02 C6 C7 08 0A C8 FF 12 00 04 C9
1AE0 CA 08 08 0A CB 08 0B 12 00 01 CC CD 08 08 0A CE
1AF0 CF 08 08 0B 12 00 08 D0 D1 08 08 0A D2 D3 FF A5
1B00 83 0A AA BD 7C 1A 85 73 BD 7D 1A 85 74 A2 24 A0
1B10 3C 20 64 19 A9 05 20 EE FF A0 00 B1 73 20 03 18
1B20 C8 B1 73 C9 FF D0 F4 A9 04 20 EE FF 60 01 04 02
1B30 05 06 A9 16 20 EE FF A9 02 20 EE FF 20 83 14 20
1B40 55 18 20 2D 18 20 2E 1A 20 3C 1A 20 94 18 A6 83
1B50 BD 2D 1B 29 01 F0 03 20 3D 19 A6 83 BD 2D 1B 29
1B60 02 F0 03 20 B9 19 20 FF 1A 60 03 01 00 00 00 00
1B70 00 00 7E FF FF FF 7E 7E A9 02 8D 6A 1B A9 08 A2
1B80 6A A0 1B 20 F1 FF A9 03 8D 6A 1B A9 08 A2 6A A0
```

```
1B9Ø 1B 2Ø F1 FF 6Ø Ø3 Ø2 ØØ ØØ ØØ ØØ ØØ ØØ 3F FF FF
1BAØ FF 3F 3F A9 Ø2 8D 95 1B A9 Ø8 A2 95 AØ 1B 2Ø F1
1BBØ FF A9 Ø3 8D 95 1B A9 Ø8 A2 95 AØ 1B 2Ø F1 FF 6Ø
1BCØ Ø1 ØØ F1 FF ØA ØØ Ø1 ØØ 2Ø 2B 15 A9 Ø7 A2 CØ AØ
1BDØ 1B 2Ø F1 FF 6Ø Ø1 81 Ø1 ØØ FF 14 ØA 14 7E ØØ ØØ
1BEØ 82 7E 7E 11 ØØ Ø1 ØØ 8Ø ØØ ØA ØØ 2Ø 2B 15 A9 Ø8
1BFØ A2 D5 AØ 1B 2Ø F1 FF A9 Ø7 A2 E3 AØ 1B 2Ø F1 FF
1CØØ 6Ø Ø5 12 Ø3 Ø3 F5 F6 Ø4 2Ø 67 1C A5 85 29 4Ø DØ
1C1Ø 13 E6 85 A5 85 29 3F C9 17 DØ 19 A5 85 Ø9 4Ø 85
1C2Ø 85 4C 34 1C C6 85 A5 85 29 3F C9 ØØ DØ Ø6 A5 85
1C3Ø 29 BF 85 85 A5 86 29 4Ø DØ 13 E6 86 A5 86 29 3F
1C4Ø C9 1E DØ 19 A5 86 Ø9 4Ø 85 86 4C 5D 1C C6 86 A5
1C5Ø 86 29 3F C9 ØA DØ Ø6 A5 86 29 BF 85 86 A5 85 49
1C6Ø 8Ø 85 85 2Ø 67 1C 6Ø A5 85 29 8Ø DØ Ø5 AØ 15 4C
1C7Ø 74 1C AØ 17 A2 F5 98 48 2Ø D4 17 A2 F6 68 A8 C8
1C8Ø 2Ø D4 17 AØ 3Ø A5 85 29 3F AA 2Ø 64 19 A2 ØØ BD
1C9Ø Ø1 1C 2Ø EE FF E8 EØ Ø7 DØ F5 AØ 38 A5 86 29 3F
1CAØ AA 2Ø 64 19 A2 ØØ BD Ø1 1C 2Ø EE FF E8 EØ Ø7 DØ
1CBØ F5 6Ø Ø6 Ø7 ØE ØF 19 Ø1 ØØ Ø5 ØØ ØØ 12 ØØ Ø5 19
1CCØ ØØ ØØ FB Ø4 ØØ 19 11 ØØ Ø5 ØØ ØØ A9 13 2Ø EE FF
1CDØ A6 87 BD B2 1C 2Ø EE FF A9 Ø4 2Ø EE FF A9 ØØ 2Ø
1CEØ EE FF 2Ø EE FF 2Ø EE FF E6 87 A5 87 29 Ø3 85 87
1CFØ A9 13 2Ø EE FF A6 87 BD B2 1C 2Ø EE FF A9 Ø6 2Ø
1DØØ EE FF A9 ØØ 2Ø EE FF 2Ø EE FF 2Ø EE FF C6 77 FØ
1D1Ø Ø1 6Ø A9 19 2Ø EE FF A9 Ø4 2Ø EE FF A9 ØØ 2Ø EE
1D2Ø FF 2Ø EE FF A5 88 ØA 26 7Ø ØA 26 7Ø 2Ø EE FF A5
1D3Ø 7Ø 29 Ø3 2Ø EE FF A9 12 2Ø EE FF A9 ØØ 2Ø EE FF
1D4Ø A6 87 BD B2 1C 2Ø EE FF A2 ØØ BD B6 1C 2Ø EE FF
1D5Ø EB EØ 15 DØ F5 E6 88 A9 Ø5 85 77 6Ø ØØ ØØ Ø1 26
1D6Ø 2E 14 ØE ØØ ØØ ØE ØØ Ø4 ØØ ØØ ØØ ØA ØØ ØA ØØ ØØ
1D7Ø Ø5 ØØ ØØ ØØ ØØ ØØ ØØ A2 ØØ BD 5C 1D 95 75 E8 EØ
1D8Ø 1B DØ F6 2Ø A3 1B 6Ø ØØ ØØ Ø1 26 2E 14 ØE ØØ Ø8
1D9Ø ØE ØØ FF ØØ ØØ FF FF ØØ ØA ØØ ØØ 2Ø 32 1B A2 ØØ
1DAØ BD 87 1D C9 FF FØ Ø2 95 75 E8 EØ 14 DØ F2 AØ ØØ
1DBØ A9 Ø1 85 77 2Ø CB 1C C8 CØ Ø8 DØ F4 6Ø 48 A9 11
1DCØ 2Ø EE FF 68 18 69 8Ø 2Ø EE FF 86 7Ø 84 71 A9 1F
1DDØ 2Ø EE FF 8A 4A 2Ø EE FF A9 41 38 E5 71 4A 2Ø EE
1DEØ FF A9 87 2Ø F4 FF 8A 29 1F A6 7Ø A4 71 6Ø A6 78
1DFØ A4 79 2Ø 64 19 A9 Ø5 2Ø EE FF A9 12 2Ø EE FF A9
1EØØ Ø3 2Ø EE FF A9 Ø3 2Ø EE FF A2 F2 A5 75 29 Ø1 FØ
1E1Ø Ø2 A2 F3 8A 2Ø EE FF A9 Ø4 2Ø EE FF 6Ø 2Ø EE 1D
1E2Ø A5 88 4A 4A 18 69 Ø3 C5 79 9Ø ØE A9 26 85 78 A9
1E3Ø 2E 85 79 A9 ØØ 85 75 85 76 A5 75 29 1F DØ ØA E6
1E4Ø 76 A6 76 BD 7C 18 ØA 85 75 A5 75 29 8Ø FØ Ø2 C6
1E5Ø 79 A5 75 29 4Ø FØ Ø2 C6 78 A5 75 29 2Ø FØ Ø2 E6
1E6Ø 78 C6 75 2Ø EE 1D 6Ø Ø5 12 Ø3 Ø4 ØB E1 12 Ø3 Ø1
1E7Ø Ø8 E2 Ø8 ØA E3 12 Ø3 Ø2 Ø8 ØB E4 Ø8 ØA E5 Ø4 ØØ
1E8Ø Ø5 12 Ø3 Ø4 ØB E1 12 Ø3 Ø1 Ø8 E2 Ø8 ØA E3 12 Ø3
1E9Ø Ø2 Ø8 ØB E6 Ø8 ØA E7 Ø4 ØØ A6 7A A4 7B 2Ø 64 19
1EAØ A2 ØØ A5 7C 29 Ø1 FØ Ø2 A2 19 BD 67 1E 2Ø EE FF
1EBØ E8 C9 ØØ DØ F5 6Ø A5 7A 29 Ø1 FØ Ø7 A5 7B 29 Ø1
1ECØ FØ Ø1 6Ø A9 ØØ A6 7A A4 7B 88 88 2Ø BD 1D C9 ØD
1EDØ FØ 12 C9 ØE FØ ØE C8 C8 A5 7D 29 ØF 2Ø BD 1D C9
1EEØ ØØ FØ Ø1 6Ø A5 7C Ø9 Ø4 85 7C 2Ø 15 15 6Ø A5 7C
1EFØ 29 Ø1 FØ Ø1 6Ø A5 7C 29 7F 85 7C A9 ØØ A6 7A A4
1FØØ 7B 88 88 2Ø BD 1D C9 ØF FØ 14 C9 1Ø FØ 1Ø C9 11
1F1Ø FØ ØC A5 7C 29 18 4A Ø9 1Ø Ø5 7C 85 7C 6Ø A5 7C
1F2Ø 29 Ø7 85 7C A9 81 AØ FF A2 FF 2Ø F4 FF 8A FØ Ø9
1F3Ø A5 7C Ø9 8Ø 85 7C 2Ø EB 1B A9 81 AØ FF A2 AA 2Ø
1F4Ø F4 FF 8A FØ Ø6 A5 7C Ø9 4Ø 85 7C A9 81 AØ FF A2
1F5Ø 9A 2Ø F4 FF 8A FØ Ø6 A5 7C Ø9 2Ø 85 7C A9 81 AØ
```

```
1F6Ø FF A2 EF 2Ø F4 FF 8A FØ Ø6 A5 8Ø Ø9 8Ø 85 8Ø A9
1F7Ø 81 AØ FF A2 AE 2Ø F4 FF 8A FØ Ø6 A5 8Ø 29 7F 85
1F8Ø 8Ø 6Ø A5 7C 29 6Ø DØ Ø1 6Ø 85 7Ø A5 7D 29 6Ø C5
1F9Ø 7Ø DØ Ø1 6Ø A5 7D 29 9F Ø5 7Ø 85 7D AØ Ø4 A9 Ø4
1FAØ 8D D9 15 A9 ØC 8D DA 15 A5 7Ø 29 4Ø DØ ØC AØ 19
1FBØ A9 Ø8 8D D9 15 A9 Ø4 8D DA 15 A2 E4 86 72 84 73
1FCØ 2Ø D4 17 A6 72 A4 73 E8 C8 EØ ED DØ EF A2 F4 AØ
1FDØ 14 2Ø D4 17 6Ø 2Ø 99 1E A5 7C 29 Ø4 FØ Ø6 C6 7B
1FEØ 2Ø 99 1E 6Ø 2Ø EE 1E A5 7C 29 Ø4 FØ Ø3 2Ø 15 15
1FFØ A5 7C 29 5Ø C9 5Ø DØ 17 A6 7A CA CA A4 7B 88 88
2ØØØ A9 ØØ 2Ø BD 1D C9 11 DØ Ø6 A5 7C 29 EF 85 7C A5
2Ø1Ø 7C 29 3Ø C9 3Ø DØ 17 A6 7A E8 E8 A4 7B 88 88 A9
2Ø2Ø ØØ 2Ø BD 1D C9 1Ø DØ Ø6 A5 7C 29 EF 85 7C A5 7C
2Ø3Ø 29 8Ø FØ Ø2 E6 7B A5 7C 29 1C FØ Ø2 C6 7B A5 7C
2Ø4Ø 29 4Ø FØ 19 A6 7A DØ Ø3 4C CD 2Ø CA A4 7B C8 A9
2Ø5Ø ØØ 2Ø BD 1D C9 11 FØ 75 C6 7A 2Ø C8 1B A5 7C 29
2Ø6Ø 2Ø FØ 19 A6 7A EØ 26 FØ 64 E8 E8 A4 7B C8 A9 ØØ
2Ø7Ø 2Ø BD 1D C9 1Ø FØ 56 E6 7A 2Ø C8 1B A9 ØØ 85 7Ø
2Ø8Ø 85 72 A5 7A 29 FE 18 69 Ø1 4A 66 7Ø 4A 66 7Ø 4A
2Ø9Ø 66 7Ø 85 71 A5 7B 29 FE 38 E9 Ø1 4A 66 72 4A 66
2ØAØ 72 4A 66 72 4A 66 72 85 73 AØ ØØ A2 7Ø A9 Ø9 2Ø
2ØBØ F1 FF A5 7D 29 FØ Ø5 74 85 7D A5 7C 29 F8 FØ Ø6
2ØCØ A5 7C 49 Ø1 85 7C 2Ø 82 1F 2Ø 99 1E 6Ø A5 7C 29
2ØDØ 9F 85 7C 4C 7C 2Ø 11 83 11 Ø4 1F Ø5 14 2Ø 2Ø 2Ø
2ØEØ 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø ØD 1F Ø5 15 2Ø 47 41 4D
2ØFØ 45 2Ø 4F 56 45 52 2Ø ØD 1F Ø5 16 2Ø 2Ø 2Ø 2Ø 2Ø
21ØØ 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2D A5 7B FØ Ø1 6Ø C6 89 A5 7D
211Ø Ø9 8Ø 85 7D A9 ØF A2 ØØ 2Ø F4 FF A5 89 FØ Ø1 6Ø
212Ø A2 ØØ BD D6 2Ø 2Ø EE FF E8 EØ 31 DØ F5 6Ø A2 Ø6
213Ø B5 89 C9 ØA 9Ø Ø7 38 E9 ØA 95 89 F6 88 CA DØ FØ
214Ø 6Ø A5 7E C5 7B 9Ø Ø1 6Ø A5 8F 18 69 Ø7 85 8F 2Ø
215Ø 2E 21 A9 11 2Ø EE FF A9 8Ø 2Ø EE FF 2Ø 3C 1A A5
216Ø 7B 85 7E C9 38 FØ Ø1 6Ø A5 8C 18 65 83 85 8C E6
217Ø 8C E6 8C 2Ø 2E 21 E6 83 A5 83 C9 Ø5 DØ Ø6 A9 ØØ
218Ø 85 83 E6 89 A5 7D Ø9 8Ø 85 7D A9 ØF A2 ØØ 2Ø F4
219Ø FF 2Ø E7 14 A5 85 C9 ØØ FØ Ø5 18 E9 Ø1 85 84 6Ø
21AØ Ø1 Ø7 ØD 14 1E 19 A2 Ø3 AØ ØØ DE 9F 21 DØ 3F BD
21BØ A2 21 9D 9F 21 86 7Ø 84 71 A9 Ø5 2Ø EE FF A9 12
21CØ 2Ø EE FF A9 Ø3 2Ø EE FF A9 Ø1 2Ø EE FF B9 28 19
21DØ ØA AA A9 21 38 F9 29 19 ØA A8 2Ø 64 19 A9 EE 2Ø
21EØ EE FF A9 Ø4 2Ø EE FF 2Ø C8 14 A6 7Ø A4 71 C8 C8
21FØ CA DØ B7 6Ø ØØ ØØ ØØ Ø1 Ø3 Ø1 Ø2 Ø1 Ø1 Ø1 Ø5 Ø1
22ØØ Ø5 Ø1 Ø1 Ø1 Ø1 2Ø 78 1B 2Ø 5B ØF A9 Ø4 85 8Ø 2Ø
221Ø ØØ 11 A5 8Ø DØ F9 29 F8 13 A5 8Ø 48 2Ø 77 1D 68
222Ø 85 8Ø 2Ø 9B 1D A5 8Ø Ø9 Ø4 85 8Ø A9 ØF A2 ØØ 2Ø
223Ø F4 FF A9 ØØ 85 7Ø A9 Ø2 A2 7Ø AØ ØØ 2Ø F1 FF 2Ø
224Ø 99 1E A6 83 BD 2D 1B 29 Ø4 FØ Ø3 2Ø EE 1D CE F9
225Ø 21 DØ 12 AD FA 21 8D F9 21 A6 83 BD 2D 1B 29 Ø4
226Ø FØ Ø3 2Ø 1D 1E 2E FB 21 DØ Ø9 AD FC 21 8D FB 21
227Ø 2Ø CB 1C CE FD 21 DØ Ø9 AD FE 21 8D FD 21 2Ø Ø8
228Ø 1C CE FF 21 DØ Ø9 AD ØØ 22 8D FF 21 2Ø ØØ 11 CE
229Ø Ø1 22 DØ 12 AD Ø2 22 8D Ø1 22 A6 83 BD 2D 1B 29
22AØ Ø2 FØ Ø3 2Ø A6 21 CE F7 21 DØ ØF AD F8 21 8D F7
22BØ 21 2Ø D5 1F 2Ø 41 21 2Ø Ø7 21 CE Ø3 22 DØ ØF AD
22CØ Ø4 22 8D Ø3 22 A5 7C 29 Ø4 DØ Ø3 2Ø B6 1E A9 81
22DØ AØ FF A2 8F 2Ø F4 FF 8A FØ Ø3 4C 16 22 2Ø Ø9 23
22EØ A5 7D 29 8Ø DØ Ø3 4C 4E 22 A9 FF 2Ø 6E 14 A9 FF
22FØ 2Ø 6E 14 A5 89 FØ Ø3 4C 22 22 A9 81 AØ FF A2 9D
23ØØ 2Ø F4 FF 8A FØ F4 4C 16 22 A9 Ø1 A2 7Ø AØ ØØ 2Ø
231Ø F1 FF A5 7Ø C5 84 9Ø F1 38 E5 84 85 7Ø A9 Ø2 A2
232Ø 7Ø AØ ØØ 2Ø F1 FF 6Ø ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```

You'll notice that the music routine is complete out on its own, away from the main program. This does not matter with the *INPUT* assembler, but if you are using another assembler you may find that it tries to assemble the music routine over the assembler itself.

If you find this happening, give the music routine a new origin. Try using 24,000 instead of 30,000—and don't forget to change the **EQU**ates and **JSR 3ØØØØ**s so that the processor jumps to the new start address when the music routine is called.

If you have a Tandy you'll notice that Willie walks and jumps on the left and right arrows, rather than when you press the M and N keys. To alter that do the following **POKE**s:

```
POKE &H4EØC,&HFD
POKE &H4E15,&HFD
POKE &H4ELD,&HFD
```

Use your machine code monitor to **PEEK** your computer's memory and check that it tallies with the following hex dump. Note on the key-scan differences and different ROM calls will result in slight differences in the Tandy versions.

```
4268 Ø3 ØC Ø9 Ø6 Ø6 Ø8 Ø1 ØE Ø7 Ø5 12 Ø3 12 Ø5 Ø1 14
4278 Ø5 Ø4 2Ø Ø2 19 2Ø Ø1 2E Ø4 ØF Ø5 17 12 Ø9 14 14
4288 Ø5 ØE 2Ø Ø2 19 2Ø 13 2E ØB Ø5 ØC ØC Ø1 17 Ø1 19
4298 Ø1 ØE Ø4 2Ø Ø7 2E Ø8 Ø5 Ø4 ØC Ø5 19 2Ø Ø1 Ø6 14
42A8 Ø5 12 2Ø Ø1 2Ø 13 Ø8 ØF 12 14 2Ø 17 Ø1 ØC ØB 2Ø
42B8 17 Ø9 ØC ØC Ø9 Ø5 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 Ø5 14 15
42C8 12 ØE 13 2Ø 14 ØF 2Ø Ø6 Ø9 ØE Ø4 2Ø 14 Ø8 Ø5 2Ø
42D8 Ø7 ØF Ø1 14 13 2Ø Ø8 Ø1 16 Ø5 2Ø 2Ø 13 1Ø 12 Ø5
42E8 Ø1 Ø4 2Ø Ø8 Ø9 13 2Ø 1Ø Ø9 Ø3 ØE Ø9 Ø3 2Ø Ø7 ØF
42F8 ØF Ø4 Ø9 Ø5 13 2Ø Ø1 ØC ØC 2Ø 2Ø 2Ø ØF 16 Ø5 12
43Ø8 2Ø Ø1 2Ø 12 ØF Ø3 ØB 19 2Ø Ø5 ØD Ø2 Ø1 ØE ØB ØD
4318 Ø5 ØE 14 2E 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 17 Ø9 ØC
4328 ØC Ø9 Ø5 2Ø 13 Ø5 14 13 2Ø ØF Ø6 Ø6 2Ø 14 ØF 2Ø
4338 12 Ø5 Ø3 ØC Ø1 Ø9 ØD 2Ø Ø8 Ø9 13 2Ø ØC ØF 13 14
4348 2Ø 1Ø ØF 13 13 Ø5 13 13 Ø9 ØF ØE 13 2C Ø2 15 14
4358 2Ø Ø9 13 2Ø Ø8 Ø1 ØD 1Ø Ø5 12 Ø5 Ø4 Ø2 19 2Ø Ø6
4368 Ø1 ØC ØC Ø9 ØE Ø7 2Ø Ø2 ØF 15 ØC Ø4 Ø5 12 13 2C
4378 1Ø ØF 14 2Ø Ø8 ØF ØC Ø5 13 2C 2Ø 2Ø Ø1 ØE Ø4 2Ø
4388 16 Ø9 Ø3 Ø9 ØF 15 13 2Ø 13 ØE Ø1 ØB Ø5 13 2E 14
4398 ØF 2Ø ØD Ø1 ØB Ø5 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø ØD Ø1 14 14
43A8 Ø5 12 13 2Ø 17 ØF 12 13 Ø5 2Ø 14 Ø8 Ø5 2Ø 14 Ø9
43B8 Ø4 Ø5 2Ø Ø9 13 2Ø 12 Ø9 13 Ø9 ØE Ø7 Ø1 ØE Ø4 2Ø
43C8 Ø8 Ø5 2Ø Ø9 13 2Ø Ø9 ØE 2Ø Ø4 Ø1 ØE Ø7 Ø5 12 2Ø
43D8 ØF Ø6 2Ø Ø2 Ø5 Ø9 ØE Ø7 2Ø Ø3 15 14 ØF Ø6 Ø6 2E
43E8 14 ØF 2Ø Ø8 Ø5 ØC 1Ø 2Ø 17 Ø9 ØC ØC Ø9 Ø5 2Ø Ø9
43F8 ØE 2Ø Ø8 Ø9 13 2Ø 11 15 Ø5 13 14 2Ø 12 Ø5 Ø1 Ø4
44Ø8 2Ø 14 Ø8 Ø5 2Ø Ø6 ØF ØC ØC ØF 17 Ø9 ØE Ø7 2Ø Ø1
4418 ØE Ø4 2Ø 1Ø 12 Ø5 13 13 2Ø 27 13 27 14 ØF 2Ø 13
4428 14 Ø1 12 14 2E ØE 2Ø 2Ø 2Ø 2Ø 2D 2Ø 12 15 ØE ØD
4438 2Ø 2Ø 2Ø 2Ø 2D 2Ø 16 Ø5 12 14 Ø9 Ø3 Ø1 ØC 2Ø ØA
4448 15 ØD 1Ø Ø2 ØF 14 Ø8 2Ø 2D 2Ø Ø4 Ø9 Ø1 Ø7 ØF ØE
```

```
4458 Ø1 ØC 2Ø ØA 15 ØD 1Ø 23 23 23 21 23 23 21 23 23
4468 23 21 23 23 21 23 23 23 23 21 23 23 21 23 23 21
4478 23 23 23 21 23 23 21 21 55 54 5Ø 5Ø 4Ø 4Ø ØØ ØØ
4488 7F 5F 57 D7 F5 FF D5 75 DD 77 5D D5 7D 75 5D 77
4498 F5 FD 57 75 DD 77 5D D5 FD 5F 57 D7 5D FF D5 7F
44A8 75 77 FD F7 D5 5D 75 77 55 D5 D5 5D 5D D7 F5 7D
44B8 D5 5D 5D D7 55 57 FF 7F 57 57 FD FD 55 75 D5 55
44C8 55 75 D5 55 75 75 D5 D5 5D 5D D5 D5 5D 5D D7 55
44D8 57 5D D7 5D D5 D5 5D 5D 75 D7 DD 75 7D 5D 75 D5
44E8 5D 75 5D D7 57 75 5D 5D F5 D5 57 75 5D D5 57 75
44F8 57 55 55 D7 F7 55 55 DD 57 55 55 D5 77 55 55 DF
45Ø8 D5 D5 57 55 5D D5 57 7F F5 75 5D 55 57 75 5D F5
4518 57 5D 75 5F 5D 57 D7 55 75 75 57 55 55 77 75 D5
4528 55 D7 75 D5 55 D7 75 75 57 57 5D 75 57 57 5D 55
4538 55 57 5D 55 7D D7 D7 D7 7D 5D 7D 5D 5D FF 7D D7
4548 5D 75 FF FD 57 FD 57 FD 5D 7D DD FF 5D FF D5 7D
4558 57 FD 7F D5 FD D7 7D FF 57 5D 75 75 7D D7 7D D7
4568 7D 7F D7 7F 57 57 57 5F 5F 57 5F 5F 5F 5F D5 F5
4578 F5 D5 F5 F5 F5 F5 5F 5F 57 57 57 57 57 57 F5 F5
4588 D5 D5 D5 D5 D5 FD 55 55 55 55 55 55 55 55 57 5F
4598 5F 57 55 57 57 57 D5 F5 F5 D5 55 55 55 FD 55 55
45A8 55 55 55 55 55 55 55 55 55 55 55 55 55 55 FD 55
45B8 57 5D 75 D5 75 55 55 55 D5 75 5D 5D 5F 55 55 55
45C8 55 55 55 55 55 55 55 55 55 55 57 5F 5F 57 55 55
45D8 55 55 D5 F5 F5 D5 55 57 57 57 FD 55 57 5D 55 55
45E8 55 FD 55 55 D5 75 75 D5 75 55 55 55 55 55 5D 5D
45F8 5F 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
46Ø8 55 55 57 5F 5F 57 55 55 55 55 D5 F5 F5 D5 55 55
4618 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 57
4628 57 57 FD 55 57 5D 55 55 55 FD 55 55 D5 75 55 55
4638 55 55 55 55 55 55 55 55 55 55 55 55 55 55 75 D5
4648 75 55 55 55 55 55 5D 5D 5F 55 55 55 55 55 55 55
4658 55 55 55 55 55 55 57 5F 7F FF FF FF 7F 5F F5 FD
4668 FF FF FF FD F5 D5 55 55 55 55 55 55 55 55 5F 7F
4678 FF FF FF 7F 5F 57 D5 F5 FD FF FF FF FD F5 55 55
4688 55 55 55 55 55 55 5D 57 55 55 55 55 55 55 5D 75
4698 D5 D5 D5 D5 D5 D5 57 5F 7D 7F 7F 5F 57 57 D5 F5
46A8 FD FD FD F5 D5 D5 57 57 55 55 55 55 55 55 D5 D5
46B8 F5 F5 7D 7D 5F 5F 55 55 55 55 57 57 5F 5F 7D 7D
46C8 F5 F5 D5 D5 55 55 7D 7D F5 F5 7D 7D 5F 5F 55 55
46D8 7D 5F 7D F5 D5 D5 55 55 AA AA 56 56 AA AA 55 55
46E8 AA AA 95 95 AA AA 55 55 7F FF FF FF 7F 5F 81 15
46F8 7D FF FF FF FD F5 57 57 57 5F 5F 5F 5F 5F 55 55
47Ø8 55 D5 D5 D5 D5 D5 57 5F 7F 5F 5F 5F 57 57 55 D5
4718 F5 D5 D5 D5 55 55 AA AA AA A6 99 6A AA AA AA AA
4728 AA AA A9 66 9A AA AA AA AA 6A 9A A6 A9 AA AA AA
4738 AA A6 99 6A AA AA ØØ Ø5 ØØ ØØ ØØ ØØ ØØ ØØ Ø6 1D
4A38 BD 4A CC 8E Ø4 21 1Ø 8E 42 68 C6 Ø5 BD 4A E6 8E
4A48 Ø4 67 C6 Ø6 BD 4A E6 8E Ø5 61 C6 1Ø BD 4A E6 8E
4A58 Ø5 AØ C6 15 BD 4A E6 8E Ø5 C7 C6 ØC BD 4A E6 86
4A68 Ø5 8E FF FF 3Ø 1F 26 FC 4A 26 F6 BD 4A CC 8E Ø4
4A78 ØØ 1Ø 8E 42 A4 5F BD 4A E6 C6 89 BD 4A E6 3Ø 88
4A88 18 C6 ØA BD 4A E6 3Ø 88 16 C6 14 BD 4A E6 3Ø ØC
4A98 C6 14 BD 4A E6 BD 8Ø Ø6 81 53 26 F9 39 BD 4A EE
4AA8 BD 4A D9 8E 15 FF 1Ø 8E 44 5F C6 2Ø 34 Ø4 BD 4A
4AB8 FD BD 4B 12 35 Ø4 5A 26 F3 1Ø 8E 44 C4 8E Ø6 21
4AC8 BD 4B 43 39 8E Ø4 ØØ 86 8Ø A7 8Ø 8C Ø6 ØØ 25 F9
4AD8 39 8E Ø6 ØØ 86 55 A7 8Ø 8C 1E ØØ 25 F9 39 A6 AØ
4AE8 A7 8Ø 5A 26 F9 39 86 E5 B7 FF 22 B7 FF C3 B7 FF
4AF8 C5 B7 FF C7 39 34 3Ø 8E Ø6 ØØ 1Ø 8E Ø6 Ø1 A6 AØ
4BØ8 A7 8Ø 8C 1D FF 25 F7 35 3Ø 39 34 1Ø A6 AØ 8Ø 21
4B18 26 1C 35 1Ø 3Ø 89 FF ØØ 34 1Ø 34 2Ø 1Ø 8E 44 8Ø
4B28 C6 Ø8 A6 AØ A7 84 3Ø 88 2Ø 5A 26 F6 35 2Ø 6F 84
4B38 3Ø 88 2Ø 8C 1E ØØ 25 F6 35 1Ø 39 C6 1E 34 Ø4 C6
4B48 Ø4 A6 AØ A7 8Ø 5A 26 F9 3Ø 88 1C 35 Ø4 5A 26 ED
4B58 39 B6 47 3E C6 1Ø 3D C3 46 DE 1F Ø3 8E ØA DE BD
4B68 4B CA B6 47 3E 27 ØE 34 Ø2 BD 4B 7E 35 Ø2 81 Ø1
4B78 27 Ø3 BD 4B 9A 39 8E 13 E7 CE ØB FF BD 4B B6 8E
4B88 11 EF CE ØB FF BD 4B B6 8E ØE F9 CE ØB FF BD 4B
4B98 B6 39 8E 13 E7 CE 46 9E BD 4B B6 8E 11 EF CE 46
4BA8 9E BD 4B B6 8E ØE F9 CE 46 9E BD 4B B6 39 C6 Ø4
4BB8 34 54 BD 4B CA 35 54 3Ø 89 Ø1 ØØ 33 C8 1Ø 5A 26
4BC8 EF 39 C6 Ø2 34 14 C6 Ø8 37 Ø2 A7 84 3Ø 88 2Ø 5A
4BD8 26 F6 35 14 3Ø Ø1 5A 26 EB 39 BD 4A 38 86 Ø5 B7
4BE8 47 3F 7F 47 3E 8E 47 4Ø C6 Ø6 6F 8Ø 5A 26 FB 86
4BF8 Ø6 B7 47 46 8E 1D ØØ BF 47 47 8E 13 EØ BF 47 49
4CØ8 7F 47 4B 7F 47 4C 8E ØB FE BF 47 4D 7F 47 4F 86
4C18 Ø5 B7 47 5Ø 86 ØA B7 47 51 BD 4A A5 8E Ø7 ØF CE
4C28 44 88 C6 Ø5 86 Ø3 37 2Ø 1Ø AF 81 4A 26 F8 3Ø 88
4C38 1A 5A 26 FØ BD 4C 77 8E Ø8 ØF CE 44 A6 C6 Ø5 86
4C48 Ø3 37 2Ø 1Ø AF 81 4A 26 F8 3Ø 88 1A 5A 26 FØ B6
4C58 47 3F C6 Ø5 3D C3 45 3C 1F Ø3 8E Ø8 16 C6 Ø5 37
4C68 Ø2 A7 84 3Ø 88 2Ø 5A 26 F6 BD 75 3Ø 7E 51 C4 34
4C78 36 8E 47 4Ø C6 Ø6 1Ø 8E Ø7 16 A6 84 34 14 C5 Ø1
4C88 26 24 C6 Ø5 3D C3 45 3C 1F Ø1 34 2Ø C6 Ø5 A6 8Ø
4C98 A7 A4 31 A8 2Ø 5A 26 F6 35 2Ø 31 21 35 14 3Ø Ø1
4CA8 5A 26 D7 35 36 39 C6 Ø5 3D C3 45 3C 1F Ø1 34 2Ø
4CB8 C6 Ø5 A6 8Ø 34 Ø2 84 ØF 48 48 48 48 8A Ø5 A7 21
4CC8 35 Ø2 44 44 44 44 8A 5Ø A7 A4 31 A8 2Ø 5A 26 E2
4CD8 35 2Ø 31 22 2Ø C6 CE 47 1E B6 47 46 85 Ø2 27 Ø3
4CE8 CE 47 2E BE 47 47 86 1Ø 34 42 BD 4B CA 35 42 4A
4CF8 26 F6 7A 47 46 26 ØF 86 ØA B7 47 46 BE 47 47 3Ø
4DØ8 89 FF ØØ BF 47 47 39 7A 47 52 26 3Ø 86 Ø5 B7 47
4D18 52 13 8E Ø6 21 86 1E 34 Ø2 86 Ø2 1C FE 34 Ø1 5F
4D28 35 Ø1 66 85 34 Ø1 5C C1 ØE 26 F5 68 84 35 Ø1 66
4D38 84 4A 26 E7 3Ø 88 2Ø 35 Ø2 4A 26 DB 39 B6 47 3E
4D48 27 Ø5 81 Ø3 27 Ø1 39 BE 47 4D CE Ø6 ØØ 34 1Ø BD
4D58 4B CA 35 1Ø 3Ø 1F 8C 14 EØ 27 4B BF 47 4D A6 84
4D68 81 AA 27 42 81 55 27 Ø9 81 5D 27 Ø5 86 Ø2 B7 47
4D78 4C 3Ø 89 Ø1 21 A6 84 81 AA 27 2B 81 55 26 Ø6 3Ø
4D88 88 DF BF 47 4D B6 47 54 27 ØD BE 47 4D CE 46 76
4D98 BD 4B CA 7F 47 54 39 BE 47 4D CE 46 5E BD 4B CA
4DAB 86 Ø1 B7 47 54 39 BE 47 4D CE Ø6 ØØ BD 4B CA 8E
4DB8 ØB FE BF 47 4D 39 FC 47 49 C4 1F C1 1E 26 Ø5 86
4DC8 Ø1 B7 47 4C B6 47 55 1Ø 26 ØØ D9 BE 47 49 3Ø 89
4DD8 Ø2 2Ø AE 84 8C 55 55 1Ø 27 ØØ BB 8C AA AA 1Ø 27
4DE8 ØØ B4 8C 5F F5 1Ø 27 ØØ AD 5F 7F 47 58 86 BF B7
4DF8 FF Ø2 B6 FF ØØ B7 47 56 86 DF B7 FF Ø2 B6 FF ØØ
4EØ8 B7 47 57 81 F7 26 1A C6 Ø1 B6 47 56 81 F7 26 Ø2
4E18 C6 81 F7 47 55 B6 47 58 26 15 BE 47 49 34 1Ø 2Ø
4E28 3C B6 47 56 81 F7 26 EA 86 Ø1 B7 47 58 2Ø E3 BE
4E38 47 49 CE Ø6 ØØ BD 4B CA 3Ø 89 ØØ FE BD 4B CA BE
4E48 47 49 3Ø Ø1 34 1Ø 3Ø 89 Ø1 61 A6 84 81 D5 27 44
4E58 81 FF 27 4Ø 81 5Ø 27 44 B6 47 4B 27 16 AE E4 CE
4E68 45 6E BD 4B CA AE E4 3Ø 89 Ø1 ØØ BD 4B CA 7F 47
4E78 4B 2Ø 19 AE E4 CE 45 96 BD 4B CA AE E4 3Ø 89 Ø1
4E88 ØØ CE 45 B6 BD 4B CA 86 Ø1 B7 47 4B AE E4 BF 47
4E98 49 35 1Ø 39 35 1Ø 86 Ø2 B7 47 4C 39 35 1Ø 3Ø 1F
4EA8 34 1Ø 2Ø B9 BD 51 6F B6 47 55 81 Ø1 26 2C 7C 47
4EB8 55 BE 47 49 34 1Ø 3Ø 89 Ø1 ØØ CE Ø6 ØØ BD 4B CA
4EC8 35 1Ø 3Ø 89 FF ØØ BF 47 49 CE 45 96 BD 4B CA 3Ø
4ED8 89 ØØ FE CE 45 B6 BD 4B CA 39 81 Ø2 26 1F 7C 47
4EE8 55 BE 47 49 3Ø 89 FF ØØ CE 45 CE BD 4B CA 3Ø 89
4EF8 ØØ FE BD 4B CA 3Ø 89 ØØ FE BD 4B CA 39 81 Ø3 26
4FØ8 2F 7C 47 55 BE 47 49 3Ø 89 FF ØØ CE Ø6 ØØ BD 4B
4F18 CA 3Ø 89 ØØ FE CE 46 Ø6 BD 4B CA 3Ø 89 ØØ FE CE
4F28 46 26 BD 4B CA 3Ø 89 ØØ FE CE 46 46 BD 4B CA 39
4F38 81 Ø4 26 25 7F 47 55 BE 47 49 34 1Ø CE Ø6 ØØ BD
4F48 4B CA 35 1Ø 3Ø 89 Ø1 ØØ BF 47 49 CE 45 6E BD 4B
4F58 CA 3Ø 89 ØØ FE BD 4B CA 39 81 81 26 3Ø 7C 47 55
4F68 BE 47 49 3Ø 89 Ø1 22 A6 84 81 57 1Ø 27 FF 27 BD
4F78 5Ø 3F BE 47 49 3Ø 89 FF Ø1 BF 47 49 CE 45 96 BD
4F88 4B CA 3Ø 89 ØØ FE CE 45 B6 BD 4B CA 39 81 82 26
4F98 4Ø 7C 47 55 BD 5Ø 3F BE 47 49 3Ø 89 FF Ø1 BF 47
4FA8 49 CE 45 CE BD 4B CA 3Ø 89 ØØ FE BD 4B CA 3Ø 89
4FB8 ØØ FE BD 4B CA BE 47 49 3Ø 89 Ø3 6Ø A6 8Ø 81 FF
4FC8 27 Ø7 A6 84 81 FF 27 Ø1 39 86 Ø4 C6 Ø5 BD 51 ØF
4FD8 39 81 83 26 2B 7C 47 55 BD 5Ø 3F 3Ø 89 ØØ FE CE
4FE8 Ø6 ØØ BD 4B CA BE 47 49 3Ø 89 Ø1 Ø1 BF 47 49 CE
4FF8 45 96 BD 4B CA 3Ø 89 ØØ FE CE 45 B6 BD 4B CA 39
5ØØ8 81 84 26 1C BD 5Ø 3F BE 47 49 34 1Ø 3Ø 89 Ø2 ØØ
5Ø18 1Ø AE 84 35 1Ø 1Ø 8C 55 55 26 Ø5 3Ø Ø1 BF 47 49
5Ø28 7F 47 55 7F 47 4B BE 47 49 CE 45 6E BD 4B CA 3Ø
5Ø38 89 ØØ FE BD 4B CA 39 BE 47 49 CE Ø6 ØØ BD 4B CA
5Ø48 3Ø 89 ØØ FE BD 4B CA 39 86 88 8E ØØ 8C BD 51 33
5Ø58 86 83 8E ØØ D5 BD 51 33 BE 47 49 CE Ø6 ØØ BD 4B
5Ø68 CA 3Ø 89 ØØ FE BF 47 49 CE 45 6E BD 4B CA 3Ø 89
5Ø78 ØØ FE BD 4B CA 86 1E 8E ØØ 71 BD 51 33 BE 47 49
5Ø88 8C 1B ØØ 25 D3 7A 47 3F 1Ø 26 FB 63 86 Ø5 8E FF
5Ø98 FF 3Ø 1F 26 FC 4A 26 F6 BD 4A CC B6 FF 22 84 ØF
5ØA8 B7 FF 22 B7 FF C2 B7 FF C4 B7 FF C6 1Ø 8E Ø5 ØB
5ØB8 8E Ø7 Ø1 AF A1 8E ØD Ø5 AF A1 8E 2Ø ØF AF A1 8E
5ØC8 16 Ø5 AF A1 8E 12 21 AF A1 86 C8 8E ØØ FF BD 51
5ØD8 33 86 C8 8E ØØ C8 BD 51 33 86 FF 8E ØØ FF BD 51
5ØE8 33 86 64 B7 51 EE 16 FA F1 86 FF 8E ØØ 96 BD 51
5ØF8 33 B6 47 3E 4C 84 Ø3 B7 47 3E 7A 51 EE C6 Ø5 86
51Ø8 Ø3 BD 51 ØF 16 FA E8 1E 89 8E 47 4Ø 3A 34 12 BD
5118 51 23 35 12 4A 26 F2 BD 4C 77 39 A6 84 4C 81 ØA
5128 26 Ø6 6F 84 3Ø 1F 2Ø F3 A7 84 39 34 Ø2 B6 FF Ø1
5138 84 F7 B7 FF Ø1 B6 FF Ø3 84 F7 B7 FF Ø3 B6 FF 23
5148 8A Ø8 B7 FF 23 1A 5Ø 35 Ø2 34 1Ø C6 FC F7 FF 2Ø
5158 3Ø 1F 26 FC AE E4 7F FF 2Ø 3Ø 1F 26 FC AE E4 4A
5168 26 EB 1C AF 35 1Ø 39 8E ØØ 62 86 Ø4 BD 51 33 39
5178 B6 47 3E 81 Ø2 24 Ø1 39 1Ø 8E 47 4F 8E 13 E7 BD
5188 51 97 8E 11 EF BD 51 97 8E ØE F9 BD 51 97 39 34
5198 1Ø BE 47 47 3Ø 88 1F AC E4 35 1Ø 22 Ø1 39 A6 A4
51A8 4C 84 ØF A7 AØ 81 Ø7 24 Ø1 39 CE 46 8E 81 ØF 26
51B8 Ø3 CE Ø6 ØØ 3Ø 89 FF ØØ BD 4B CA 39 86 Ø5 B7 47
51C8 52 7F 47 55 BD 4B 59 BD 4D BE BD 4D 45 BD 51 78
51D8 BD 4C DE BD 4D ØF B6 47 4C 81 Ø1 1Ø 27 FF ØA 81
51E8 Ø2 1Ø 27 FE 63 C6 64 4F 4A 26 FD 5A 26 F9 BD 8Ø
51F8 Ø6 81 Ø3 26 D2 39 39 3
```

CLIFFHANGER MUSIC

```
753Ø 8E 75 8A BF 75 88 86 13 34 Ø2 B6 FF Ø1 84 F7 B7
754Ø FF Ø1 B6 FF Ø3 84 F7 B7 FF Ø3 B6 FF 23 8A Ø8 B7
755Ø FF 23 FE 75 88 1A 5Ø 37 12 11 83 75 CD 25 Ø3 CE
756Ø 75 8A FF 75 88 34 1Ø C6 FC F7 FF 2Ø 3Ø 1F 26 FC
757Ø AE E4 7F FF 2Ø 3Ø 1F 26 FC AE E4 4A 26 EB 32 62
758Ø 6A E4 26 CE 1C AF 35 82 75 C3 62 ØØ BD E9 ØØ 9E
759Ø 83 ØØ 8D DC ØØ 7D 4E ØØ 76 93 ØØ 7D FF ØØ 8D 6E
75AØ ØØ A8 83 ØØ D4 31 ØØ BD 6E ØØ A8 E9 ØØ 9E 62 ØØ
75BØ BD 93 ØØ BD 2C ØØ D4 62 ØØ BD DC ØØ A8 5C ØØ C8
75CØ DC ØØ FC
```

CUMULATIVE INDEX

An interim index will be published each week. There will be a complete index in the last issue of *INPUT.*

The publishers accept no responsibility for unsolicited material sent for publication in INPUT. All tapes and written material should be accompanied by a stamped, self-addressed envelope.

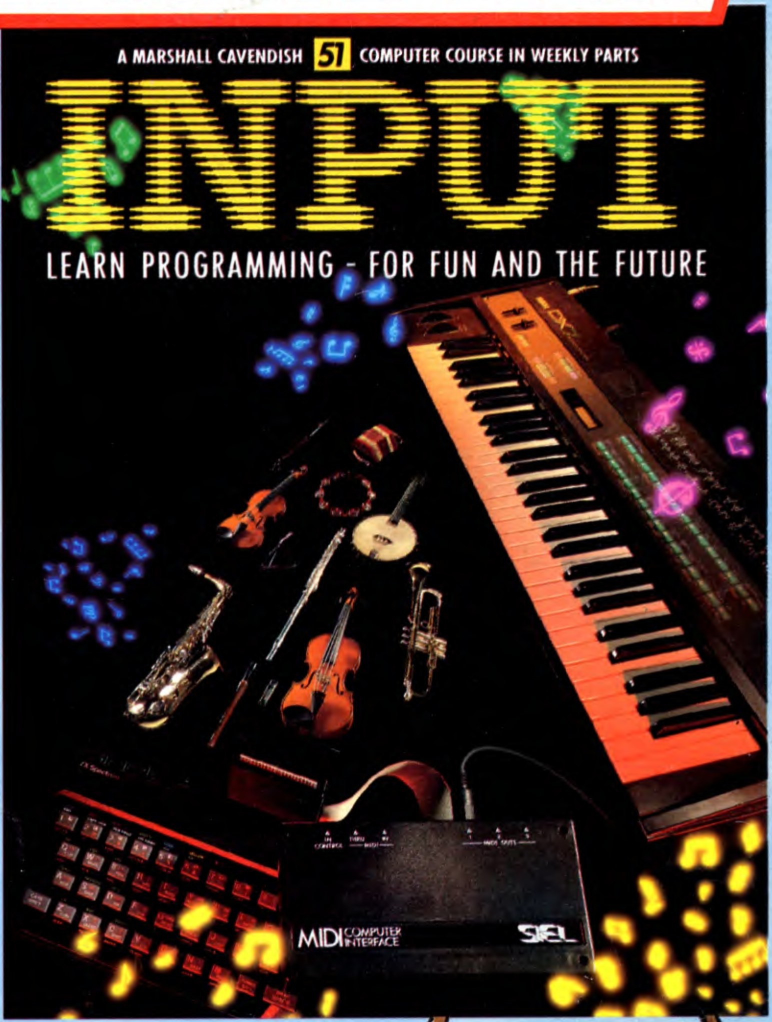
COMING IN ISSUE 51 . . .

❑ *There's a complete program to let up to six players take part in an exciting DICE GAME needing real strategy*

❑ *For budding musicians, a look at the WORLD OF MIDI, the system that lets your micro control a full-fledged musical instrument*

❑ *And for disk users, there's a handy DISK-EDITING UTILITY that gives you direct access to the bytes in storage*

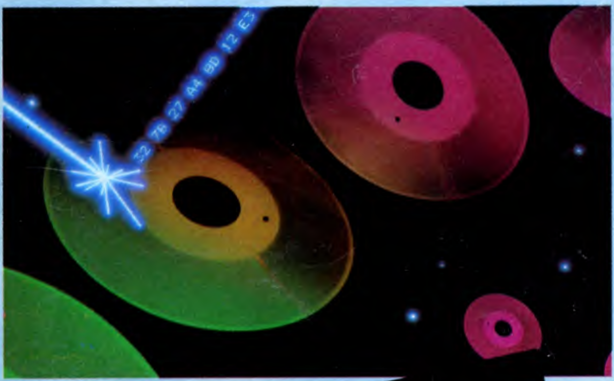
❑ *Plus, for COMMODORE users, the final part of the routine to EXPAND THE GRAPHICS COMMANDS*

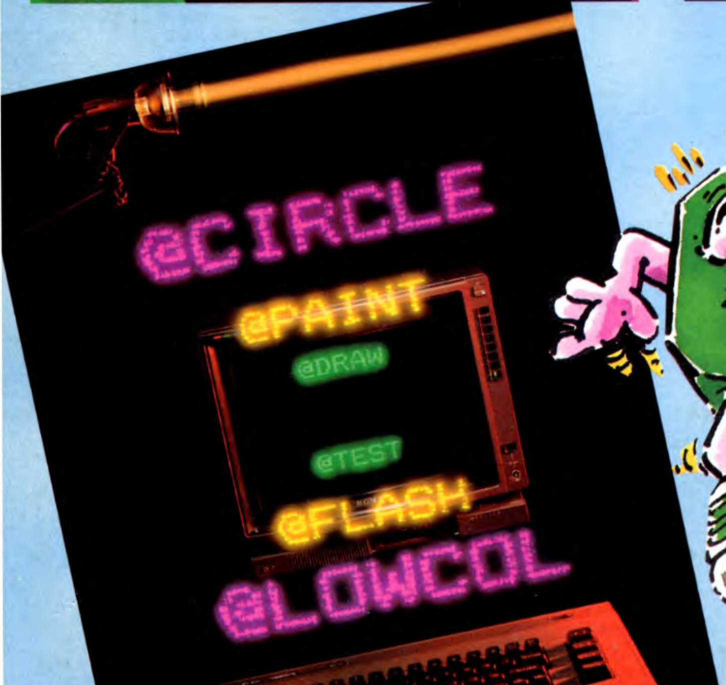


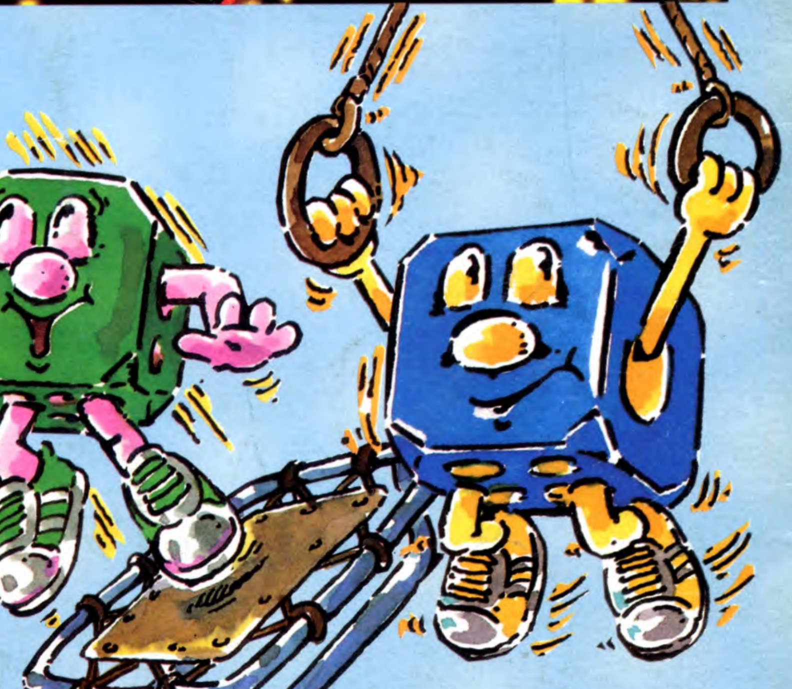
A MARSHALL CAVENDISH 51 COMPUTER COURSE IN WEEKLY PARTS

INPUT

LEARN PROGRAMMING - FOR FUN AND THE FUTURE







ASK YOUR NEWSAGENT FOR INPUT