



EDUCATIONAL PROGRAMS FOR THE DRAGON 32

Ian Murray



EDUCATIONAL PROGRAMS FOR THE DRAGON 32

Ian Murray
with Dragon conversions by
Laurence Owen



CENTURY PUBLISHING
LONDON

Copyright © Ian Murray 1983

All rights reserved

First published in Great Britain in 1983

by Century Publishing Co. Ltd.

76 Old Compton Street, London W1V 5PA

ISBN 0 7126 0259 3

Reproduced, printed and bound in Great Britain by
Hazell Watson & Viney Ltd, Aylesbury, Bucks

Cover illustration by Tony Roberts/Young Artists

Contents

Acknowledgements	7
Introduction	9
1. Tudor History	11
2. Life	18
3. Boating	26
4. Word Find Tester	30
5. Vector Race Game	39
6. Rotation	47
7. Building Blocks	50
8. 3-D Prisms	54
9. N-Particle	59
10. Anagram	65
11. Fraction Towers	73
12. Stock Market	78
13. Four Stroke Engine	92
14. Attributes	96
15. Interference	104
16. Meter Reading	111
17. Wordsquares	117
18. Simon	123
19. Hexadecimal-Binary	128
20. Number Guess	133
21. Queue	137
22. Back to Front	141
23. Maths Stars	146
24. Periodic Table	149
25. Animals	162

Acknowledgements

Contributors to this book include Alan Baugh, Ian Clarke, B Crow, Alan Dixon, Ian Murray, Andrew Pusey, Ron Woodworth and Chris Whytehead. My thanks go to all these, and the Head, teachers and pupils of Holloway School who tolerated my irritability, gave support and helped test some of the programs. My especial thanks go to Laurence Owen who converted the programs for the Dragon.

Introduction

The programs in this book have all been tried out in the classroom. Generally, young people seem to react best to the more 'competitive' programs, either against the clock or each other. I do not condone either educationally. However, to encourage young people to use the computer as a learning medium, I have included some programs in this category, but tried to steer the emphasis towards using the computer interactively. Here, the potential for the computer is enormous.

The graphics facilities of the Dragon are too good to waste, and a number of the demonstrative programs are included precisely because in a classroom, none of us can reproduce diagrams and drawings in quite the same fashion on the blackboard – and the 20 seconds' loading time for a program is hardly time wasted.

Some of the programs lend themselves readily to adaption by those with a working knowledge of Dragon Basic . . . and I am happy to encourage this. In some cases the data in the programs is there to show how the data needs to be formulated rather than as a working example of data. I would be pleased to hear of any developments of either programs or data.

Child-proofing programs is important, and this has been done to differing degrees in these programs. Do not assume that adult programmers have always thought of some of the more ludicrous button pushing of the younger generation. If every program had trapped every possible error, this book might have contained just half a dozen programs. As it is some programs are quite long.

The programs, in some cases, creep to unsavoury lengths. This has been done for readability purposes and once you understand the program, then it would be wise to shorten the variable and procedure names.

Finally my thanks go to the contributors: to the teachers and sixth formers who helped develop the software and generate the

ideas. But remember the golden rule of software: 'there does not exist the finished program.'

Ian Murray

1. Tudor History

General Description

This program is an example of a history tester program. It provides facts about and tests for the Elizabethan period. On starting the program you are faced with a menu, and unless you know your dates, I would suggest a quick peek at the facts. It will be possible to rewrite the data held from lines 1720 onwards, if you so wish.

Detailed Description

Lines 10–220 The main structure in which the introduction, menu and options are controlled.

230–320 Print statements displaying an introduction page.

330–410 The details of the menu are held in defined strings at the end of the program, otherwise this procedure is just print statements.

420–550 Again the messages are held in defined strings at the end of the program. The questions are held in data statements from 2150 onwards.

560–900 These display pages of information about the Tudors or the goodbye message. Again refer to the end of the program for the definition of the strings.

910–950 Display title and continuation information.

960–1030 Reads question and answer from data and checks for the last question.

1040–1100 This is the score display routine. Note that the space bar is the normal method of progressing through procedures.

1110–1440 These routines are the tester part of the program.

1450–1710 These are all sub-procedures called from other procedures. The answer routines (1550–1710) delete the number of characters entered so far if too many characters are

entered, this saves having to keep track of screen management. It is not possible to delete wrong characters as you enter them; you must press ENTER and accept you have put in a wrong answer.

Educational Notes

This is a program suitable for a small group who may then discuss the answers amongst themselves. It does presuppose a knowledge of the tutors to some degree but can be operated just from the data. I would not recommend this in the classroom, but as a self tutor at home it would be useful. You can, of course, use this structure to create your own questions on subject areas you have covered, in which case the data from lines 1890–2330 would need to be rewritten.

Program Listing

```
0 REM*****
20 REM*      TUDOR HISTORY      *
30 REM*      *                  *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*      *                  *
60 REM*      BY RAY NEW 1983    *
70 REM*****
80 CLEAR 3000
90 CLS
100 GOSUB 1730 'SET WORDS
110 GOSUB 240 'INTRO
120 IF INKEY$="" THEN 120
130 CLS
140 GOSUB 340 'MENU
150 A$=INKEY$
160 IF A$="1" GOSUB 430:GOTO130 'TEST
170 IF A$="2" GOSUB 570:GOTO130 'SUMM
180 IF A$="3" GOSUB 680:GOTO130 'ELIZ
190 IF A$<>" " THEN 150
200 CLS
210 GOSUB 870 'BYE
220 END
230 'INTRODUCTION
240 PRINT@34,"WELCOME TO THE TUDOR PERIOD"
250 PRINT@98,"THIS PROGRAM WILL TEACH YOU "
260 PRINT@130,"ABOUT ELIZABETH I, THE LAST"
270 PRINT@162,"TUDOR MONARCH OF ENGLAND."
```

```

280 PRINT@256,"MOVE FROM ONE SCREEN TO THE NEXT"
290 PRINT@288,"BY KEYING THE NUMBER OF YOUR"
300 PRINT@320,"CHOICE OR PRESSING THE SPACE BAR"
310 PRINT@480,PR$;" CONTINUE";
320 RETURN
330 'MENU
340 PRINT@34,CH$
350 PRINT@102,"1";OT$
360 PRINT@134,"2";SOR$
370 PRINT@166,"3";" ";E1$
380 PRINT@256,O5$
390 PRINT@288,O6$
400 PRINT@480,PR$;" END";
410 RETURN
420 'MAIN TEST ROUTINE
430 CLS
440 PRINT@0,O1$
450 PRINT@32,O2$
460 'INITIALISE
470 ARIGHT=0
480 RESTORE
490 ROW=3
500 'ASK QUESTION
510 GO SUB 970
520 'ANY MORE QUESTIONS?
530 IF QUESTION$<>"FIN"THEN510
540 GOSUB 1050
550 RETURN
560 'DISPLAY SUMMARY OF REIGNS
570 COLUMN=5:ROW=1:TITLE$=SOR$
580 GOSUB 920
590 PRINT@160,TAB(14);S1$;TAB(23);S2$;
600 PRINT@192,H7$;S3$
610 PRINT@224,H8$;S4$
620 PRINT@256,E6$;S5$
630 PRINT@288,MT$;S6$
640 PRINT@320,E1$;S7$
650 IF INKEY$<>" "THEN650
660 RETURN
670 'ELIZABETH
680 COLUMN=10:ROW=0:TITLE$=E1$:GOSUB 920
690 PRINT@32,EB$;EC$
700 PRINTED$
710 PRINTEE$;EF$
720 PRINTEG$
730 PRINTEH$;EI$
740 PRINTEJ$
750 PRINTEK$
760 PRINTEL$
770 PRINTEM$
780 IF INKEY$<>" "THEN780
790 GOSUB 920

```

```

800 PRINT@32,EN$;EO$;EP$;EQ$
810 PRINTER$
820 PRINTES$
830 PRINTET$
840 IF INKEY$<>" "THEN840
850 RETURN
860 'BYE
870 PRINT@43,"GOODBYE"
880 PRINT@68,"THANK YOU FOR YOUR TIME"
890 IF INKEY$=""THEN 890
900 RETURN
910 'DISPLAY REQUESTED SCREEN TITLE
920 CLS
930 PRINT@ROW*32+COLUMN,TITLE$;
940 PRINT@480,FR$;" CONTINUE";
950 RETURN
960 'MORE TEST
970 READ QUESTION$,ANSWER$
980 'CHECK FOR LAST QUESTION
990 IF QUESTION$<>"FIN"GOSUB1120
1000 'ANSWER$ NOW HOLDS NUMBER
1010 'OF QUESTIONS ASKED
1020 ATOTAL=VAL(ANSWER$)
1030 RETURN
1040 'RESULTS
1050 PRINT@416,04$
1060 WAIT=25:GOSUB 1460
1070 PRINT@448,"YOU HAVE SCORED ";ARIGHT;" OUT OF
";ATOTAL;
1080 IF ARIGHT=ATOTAL THENPRINT@480,"WELL DONE--FU
LL MARKS";
1090 IF INKEY$<>" "THEN1090
1100 RETURN
1110 'ASK AGAIN
1120 PRINT@32*ROW,QUESTION$;
1130 PRINTTAB(25);
1140 '*SHOWS TWO LINE QUESTION
1150 IF ANSWER$="*"GOSUB1500
1160 'KEEP ASKING UNTIL CORRECT
1170 'ANSWER GIVEN OR REQUESTED
1180 REPLY$="OK"
1190 'TEST LENGTH OF ANSWER AS
1200 'IT IS ENTERED - MAX IS 5
1210 COUNT=0:LIMIT=5:INANSWER$=""
1220 GOSUB 1560
1230 IF ANSWER$=INANSWER$ GOSUB 1310 :GOTO 1260
1240 IF INANSWER$="?"THEN PRINTANSWER$:GOTO 1260
1250 GOSUB 1350
1260 IF REPLY$="WRONG" THEN 1180
1270 ROW=ROW+1

```



```

1280 IF ROW>12 THEN WAIT=500:GOSUB 1460:PRINT@64:P
RINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRI
NT:PRINT:PRINT:PRINT:PRINT: ROW=3
1290 RETURN
1300 'TICK
1310 ARIGHT=ARIGHT+1
1320 PRINTTAB(31);"r";
1330 RETURN
1340 'CROSS
1350 REPLY$="WRONG"
1360 PRINTTAB(31);"x";
1370 PRINT@67,03$:WAIT=100:GOSUB1460
1380 PRINT@82,07$:WAIT=100:GOSUB1460
1390 PLAY"T20DDDDDDAAAAA"
1400 PRINT@64,"
1410 PRINT@ROW*32,"
"
1420 PRINT@ROW*32,QUESTION$;
1430 PRINTTAB(25);
1440 RETURN
1450 'WAIT
1460 FOR I=1 TO WAIT
1470 NEXT I
1480 RETURN
1490 'SECOND LINE
1500 READ QUESTION$,ANSWER$
1510 ROW=ROW+1
1520 PRINT@ROW*32,QUESTION$;
1530 PRINTTAB(25);
1540 RETURN
1550 'COUNT CHARACTERS
1560 COUNT=0
1570 A$=INKEY$
1580 IF A$=""THEN1570
1590 IF ASC(A$)<>13 AND ASC(A$)<>127 GO SUB 1640
1600 IF COUNT>LIMIT OR ASC(A$)=13 THEN 1610 ELSE G
OTO1570
1610 IF COUNT>LIMIT GOSUB 1680:GOTO1560
1620 RETURN
1630 'INCREASE COUNT
1640 PRINTA$;:COUNT=COUNT+1
1650 INANSWER$=INANSWER$+A$
1660 RETURN
1670 'TOO MANY
1680 INANSWER$=""
1690 PRINT@ROW*32+25," ";
1700 PRINT@ROW*32+25,"";
1710 RETURN
1720 'SET WORDS
1730 H7$="HENRY VII"
1740 H8$="HENRY VIII"
1750 MT$="MARY TUDOR"

```

1760 E6\$="EDWARD VI"
1770 E1\$="ELIZABETH I"
1780 FR\$="PRESS THE SPACE BAR TO"
1790 CH\$="CHOOSE ONE OF THE FOLLOWING"
1800 OT\$=" OVERALL TEST"
1810 SOR\$=" SUMMARY OF REIGNS"
1820 O1\$="PRESS ENTER TO GET A QUESTION."
1830 O2\$="ENTER ? TO SEE THE RIGHT ANSWER"
1840 O3\$="hold everthing"
1850 O4\$="THAT WAS THE LAST QUESTION"
1860 O5\$="YOU MAY WANT TO CHOOSE 1 BEFORE"
1870 O6\$="AND AFTER TAKING THE TUTORIAL"
1880 O7\$="try again"
1890 S1\$="LIVED":S2\$="REIGNED"
1900 S3\$=" 1457-1509 1485-1509"
1910 S4\$=" 1491-1547 1509-1547"
1920 S5\$=" 1537-1553 1547-1553"
1930 S6\$=" 1526-1558 1553-1558"
1940 S7\$=" 1533-1603 1558-1603"
1950 EB\$="1533-HENRY VIII MARRIES ANNE BOL
EYN"
1960 EC\$="ELIZABETH BORN"
1970 ED\$="1536-ANNE BOLEYN EXECUTED"
1980 EE\$="1558-MARY TUDOR DIES"
1990 EF\$="ELIZABETH BECOMES QUEEN"
2000 EG\$="1564-SHAKESPEARE BORN"
2010 EH\$="1568-MARY QUEEN OF SCOTS ARRIVES"
2020 EI\$=" IN ENGLAND"
2030 EJ\$="1569-REBELLION IN THE NORTH"
2040 EK\$="1570-POPE EXCOMMUNICATES ELI
ZABETH"
2050 EL\$="1571-RIDOLFI PLOT,DUKE OF NOR
FOLK ARRESTED"
2060 EM\$="1572-NORFOLK EXECUTED"
2070 EN\$="1577-DRAKE SETS OUT IN GOLDEN HIN
D"
2080 EO\$=" ON FIRST CIRCUMNAVIGA
TION OF"
2090 EP\$=" WORLD BY ENGLISHMEN"
2100 EQ\$=" JOURNEY TAKES 3 YEARS"
2110 ER\$="1587-MARY QUEEN OF SCOTS EXE
CUTED"
2120 ES\$="1588-SPANISH ARMADA DEFEATED"
2130 ET\$="1603-ELIZABETH DIES"
2140 RETURN
2150 DATA WHEN DID HENRY VIII DIE,1547
2160 DATA HOW OLD WAS ELIZABETH WHEN,*
2170 DATA HER MOTHER WAS BEHEADED,3
2180 DATA WHEN WAS ANNE BOLEYN,*
2190 DATA EXECUTED,1536
2200 DATA WHEN WAS MARY QUEEN OF,*
2210 DATA SCOTS EXECUTED,1587

2220 DATA HOW OLD WAS ELIZABETH WHEN,*
 2230 DATA SHE BECAME QUEEN,25
 2240 DATA WHO WAS THE FIRST ENGLISHMAN,*
 2250 DATA TO SAIL ROUND THE WORLD,DRAKE
 2260 DATA HOW MANY YEARS DID HIS,*
 2270 DATA JOURNEY TAKE,3
 2280 DATA WHEN DID THE SPANISH,*
 2290 DATA ARMADA SAIL,1588
 2300 DATA HOW MANY YEARS DID,*
 2310 DATA ELIZABETH REIGN,45
 2320 DATA WHEN DID SHE DIE,1603
 2330 DATA FIN,10

2. Life

General Description

LIFE has been about in many forms for several years. I first wrote it in Z80 assembler for the NASCOM four years ago, and I would recommend that any person who understands the logic of LIFE may enjoy doing it in 6809 Assembler.

To the uninitiated, LIFE simulates cell regeneration. Even operating relatively slowly, as it does in BASIC, it produces a generation every minute creating some lovely results including splendid examples of cell division. The rules of cell regeneration are:

- a cell DIES if it has one or less immediate neighbours, or more than three neighbours,
- a cell is STABLE if it has just two neighbours,
- a new cell is BORN if the location at which it will be created has just three neighbours.

The more ghoulish friends of mine have invented 'CANCER' as a derivative of this simulation.

After the pretty multi-coloured introductions, you decide where you will place a cell in the closed environment of your screen and use the enter key to copy it on to the screen. When satisfied with your creation press the SPACE BAR to wait for the next generation to be computed. Break gets you out of the program when you have found a stable cell structure or when you are bored.

Detailed Description

Lines 10–150 Here is a pleasure of Basic. The main structure reduced to relative simplicity.

160–260 A procedure that produces a colourful initial screen. The RND is used to determine colour and sound.

270–540 The instructions here are a bit more specific than the

rules above. There are actually only three rules for LIFE but for children we have elucidated a bit.

550–720 This routine simply keeps track of the cursor control keys and calls the toggle procedure to put a cell on or take a cell off the screen.

730–930 The guts of the operation. Consider a cell pattern as below:

()	()	(1)
()	(*)	()
(2)	(3)	()

For any and every position on the screen, if the star represents the point currently being computed each of the other positions marked must be scanned for an alive or non-existent cell. If there are cells at 1, 2 and 3 then the star will be regenerated. This routine does the scan.

940–1000 The new cell positions are stored in the OCELL array allowing the NCELL array to be used for the next generation.

1010–1090 This routine simply draws the closed environment of the screen.

1100–1120 The toggle procedure mentioned above.

Educational Notes

Apart from looking at pretty patterns, the real point of the program is to see if you can create the environment for cell division. To do this students ought to be issued with graph or patterned paper on which their initial pattern can be recorded or decided before entering it to the program. This simulation is very difficult for the less able, and I found that as a simulation it was best restricted to better fifth formers and sixth formers. On the other hand, lower down in the school, fun was had by less able youngsters doing the following:

They were explained, in some cases several times, the rules of LIFE and then issued with 30 cells. They designed, each of them, an initial pattern, and a competition winner was the person whose structure was kept regenerating the longest. Those who reached a stable structure had to have the generation of stability recorded. Inevitably, the processing of the patterns had to be at low use times for the micro.

Two versions of this program are presented here. The first, documented, version is written entirely in BASIC. It takes about one minute to process a generation, and allows the user to observe the rules of Life being performed row by row. The second version, LIFE 2, is of more practical value in the classroom, where time is limited. A machine language routine is used to speed up the cycle time per generation, and gives some indication of the capability of the Dragon's 6809 processor.

The cycle time in this version has had to be slowed down (by suitably microbic noises) to an acceptable four generations per second.

Program Listing

```

10 REM*****
20 REM*           LIFE           *
30 REM*           *               *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*           *               *
60 REM*   BY   RAY NEW  1983   *
70 REM*****
80 DIM OCELL(32,16)
90 DIM NCELL(32,16)
100 GOSUB 170 'HEADER
110 GOSUB 280 'INSTRUCTIONS
120 GOSUB 560 'SET UP
130 GOSUB 740 'NEW GENERATION
140 GOSUB 950 'TIDY
150 GOTO130
160 'HEADER*****
170 CLS
180 PRINT
190 FOR I=1TO14
200 FOR J=1TO 4
210 PRINTCHR$(127+RND(8)*16);"-life-";CHR$(127+RND
(8)*16);
220 SOUNDNRND(100+RND(50)),1
230 NEXT J
240 NEXT I
250 FOR I=1TO4000:NEXTI
260 RETURN
270 'INSTRUCTIONS*****
280 CLS
290 PRINT"-----LIFE-----";
300 PRINT
310 PRINT"THIS IS THE GAME OF LIFE. THE "
320 PRINT"AIM IS TO SET UP A PATTERN OF "
330 PRINT"CELLS ON THE SCREEN WHICH STAY"

```

```

340 PRINT"ALIVE. THE RULES ARE SIMPLE :-"
350 PRINT"1.A CELL WITH NO NEIGHBOURS DIES";
360 PRINT"2.A CELL WITH 1 NEIGHBOUR DIES"
370 PRINT"3.A CELL WITH 2 OR 3 NEIGHBOURS"
380 PRINT" LIVES TO THE NEXT GENERATION"
390 PRINT"4.A CELL WITH 4 OR MORE NEIGH-
400 PRINT" BOURS DIES"
410 PRINT"5.IF THERE ARE 3 NEIGHBOURS A"
420 PRINT" NEW CELL IS BORN"
430 PRINT@4B3,"PRESS ANY KEY TO CONTINUE";
440 IF INKEY$="" THEN 440
450 CLS
460 PRINT"-----LIFE-----":PRINT;
470 PRINT"USE THE CURSOR KEYS TO MOVE "
480 PRINT"AROUND AND THE 'ENTER' KEY TO"
490 PRINT"INSERT OR DELETE CELLS('*')."
500 PRINT"WHEN YOU HAVE COMPLETED YOUR"
510 PRINT"PATTERN, PRESS THE 'SPACE BAR'"
520 PRINT@4B3,"PRESS ANY KEY TO CONTINUE";
530 IF INKEY$="" THEN 530
540 RETURN
550 'SET UP*****
560 GEN=0
570 GOSUB 1020 'SCREEN
580 X=2:X1=2
590 Y=2:Y1=2
600 PRINT@(Y-1)*32+X-1,"";
610 A$=INKEY$
620 IF A$="" THEN 610
630 IF ASC(A$)=13GOSUB 1110
640 IF ASC(A$)=94 THEN Y1=Y-1
650 IF ASC(A$)=10 THEN Y1=Y+1
660 IF ASC(A$)=8 THEN X1=X-1
670 IF ASC(A$)=9 THEN X1=X+1
680 IF X1<2 OR X1>31 THEN X1=X
690 IF Y1<2 OR Y1>15 THEN Y1=Y
700 X=X1:Y=Y1
710 IF A$=" " THEN 720 ELSE 600
720 RETURN
730 'NEW GENERATION*****
740 GEN=GEN+1
750 PRINT@20,GEN;
760 FOR Y=2 TO 15
770 PRINT@(Y-1)*32+1,"";
780 FOR X=2 TO 31
790 Z=0
800 IF OCELL(X-1,Y-1)=1 THEN Z=Z+1
810 IF OCELL(X-1,Y)=1 THEN Z=Z+1
820 IF OCELL(X-1,Y+1)=1 THEN Z=Z+1
830 IF OCELL(X,Y-1)=1 THEN Z=Z+1
840 IF OCELL(X,Y+1)=1 THEN Z=Z+1
850 IF OCELL(X+1,Y-1)=1 THEN Z=Z+1

```



```

860 IF OCELL(X+1,Y)=1 THEN Z=Z+1
870 IF OCELL(X+1,Y+1)=1 THEN Z=Z+1
880 IF OCELL(X,Y)=1 AND Z<>2 AND Z<>3 THEN NCELL(X,Y)=0
890 IF OCELL(X,Y)=0 AND Z=3 THEN NCELL(X,Y)=1
900 IF NCELL(X,Y)=0 THEN PRINT " "; ELSE PRINT "*";
910 NEXT X
920 NEXT Y
930 RETURN
940 'TIDY*****
950 FOR X=2 TO 31
960 FOR Y=2 TO 15
970 OCELL(X,Y)=NCELL(X,Y)
980 NEXT Y
990 NEXT X
1000 RETURN
1010 'SCREEN*****
1020 CLS
1030 FOR I=1 TO 10:PRINTCHR$(159);:NEXT I
1040 PRINT"generation";GEN;
1050 FOR I=1 TO 9:PRINTCHR$(159);:NEXT I
1060 FOR I=1 TO 14:PRINT@I*32,CHR$(159):PRINT@I*32+31,CHR$(159);:NEXT I
1070 FOR I=1 TO 31:PRINTCHR$(159);:NEXT I
1080 POKE1535,159
1090 RETURN
1100 'TOGGLE*****
1110 IF OCELL(X,Y)=1 THEN OCELL(X,Y)=0:NCELL(X,Y)=0:PRINT " "; ELSE OCELL(X,Y)=1:NCELL(X,Y)=1:PRINT "*"
1120 RETURN

```

```

10 '*****
20 '*          LIFE 2          *
30 '*          *              *
40 '* WRITTEN FOR THE DRAGON *
50 '*          *              *
60 '* BY          RAY NEW      1983 *
70 '*****
80 CLEAR 200,25000
90 GOSUB1500 'INTRODUCTION
100 CLS
110 PRINT@0,"LIFE          INITIAL MODE "
120 PRINT@64,"BACKGROUND COLOUR(0-8) ";
130 PRINT@224,"H=HELP(INITIAL)"
140 PRINT@256,"ARROWS=CURSOR CONTROL(SET UP)"
150 PRINT@288,"ENTER=SELECT(SET UP/HELP)"
160 PRINT@320,"R=INITIAL(SET UP/RUN)"
170 PRINT@352,"CLEAR=CLEAR SET UP(SET UP/RUN)";
180 PRINT@384,"G=GENERATION DISPLAY(RUN)"
190 PRINT@416,"C=CONTINUOUS DISPLAY(RUN)"

```



```

200 PRINT@448,"S=RUN(SET UP)";
210 PRINT@480,"SPACE=CONTINUE (HELP/GENERATION)";
220 IK$=INKEY$
230 IF IK$<>"H"THEN320
240 FOR I=0 TO 8
250 PRINT@86,I;
260 IF I=0 THEN POKE 1119,128 ELSE POKE1119,127+16
*I
270 IK$=INKEY$
280 IF IK$=""THEN270
290 IF ASC(IK$)=13 THEN BG=I:I=8:NEXTI:GOTO350
300 NEXTI
310 GO TO 240
320 IF IK$<"0"OR IK$>"8"THEN220
330 PRINT@87,IK$;
340 BG=VAL(IK$)
350 NC=0
360 PRINT@121," ";
370 PRINT@96,"FOREGROUND SYMBOL(1-255) ";
380 IK$=INKEY$
390 IF IK$=""THEN380
400 IF IK$<>"H" THEN 510
410 FOR I=1 TO 255
420 IF I=32 OR (I>64 AND I<91) OR (I>111 AND I<122
) THEN480
430 PRINT@120,I;
440 POKE1151,I
450 IK$=INKEY$
460 IF IK$=""THEN450
470 IF ASC(IK$)=13 THEN M=I:I=255:NEXTI:GOTO580
480 NEXT I
490 PRINT@121," ";
500 GOTO410
510 IF (IK$<"0"OR IK$>"9") AND ASC(IK$)<>13 THEN 3
80
520 NC=NC+1
530 PRINTIK$;
540 C1$(NC)=IK$
550 IF NC<3 AND ASC(IK$)<>13THEN380
560 C0$=C1$(1)+C1$(2)+C1$(3)
570 M=VAL(C0$)
580 IF M>255 OR M<1 THEN 350
590 IF M=32 OR (M>64 AND M<91) OR (M>111 AND M<122
) THEN350
600 PRINT@128,"GENERATION DISPLAY(0-1) ";
610 IK$=INKEY$
620 IF IK$<>"H" THEN 710
630 FOR I=0 TO 1
640 PRINT@151,I;
650 IF I=1 THEN PRINT@159,"Y"; ELSE PRINT@159,"N";
660 IK$=INKEY$
670 IF IK$=""THEN660

```

```

680 IF ASC(ΙΚ$)=13 THEN ST=I:I=1:NEXTI:GOTO740
690 NEXTI
700 GOTO 630
710 IF ΙΚ$<>"0"AND ΙΚ$<>"1"THEN610
720 PRINTΙΚ$
730 ST=VAL(ΙΚ$)
740 IF BG=0 THEN N=128 ELSE N=127+16*BG
750 POKE&HFF23,PEEK(&HFF23)OR8
760 POKE&HFF01,PEEK(&HFF01)AND247
770 POKE&HFF03,PEEK(&HFF03)AND247
780 POKE&H7800,N
790 POKE&H7801,N
800 POKE&H7802,N
810 POKE&H7803,M
820 POKE&H7804,N
830 POKE&H7805,N
840 POKE&H7806,N
850 POKE&H7807,N
860 POKE&H7808,N
870 POKE&H7809,N
880 I=24999
890 I=I+1
900 READ A
910 IF A=-1 THEN POKE25001,M:GOTO1100
920 POKE I,A
930 GOTO 890
940 DATA 198,,142,4,32,206,112,0,16,142,120,0
950 DATA 225,1,38,2,49,33
960 DATA 225,31,38,2,49,33
970 DATA 225,136,224,38,2,49,33
980 DATA 225,136,223,38,2,49,33
990 DATA 225,136,225,38,2,49,33
1000 DATA 225,136,032,38,2,49,33
1010 DATA 225,136,031,38,2,49,33
1020 DATA 225,136,033,38,2,49,33
1030 DATA 247,120,20
1040 DATA 198,0,182,255,32,187,1,19,183,255,32,92,
193,10,38,248
1050 DATA 246,120,20
1060 DATA 16,140,120,2,38,6,225,132,38,2,49,33
1070 DATA 18,166,164,18,167,196,48,1,51,65,140,6,0
,38,149
1080 DATA 142,4,32,206,112,0,18,166,196,18,167,132
,51,65,48,1,140,6,0,38,241
1090 DATA 57,-1
1100 '*****
1110 CLSBG
1120 G=0
1130 PRINT@0,"LIFE      SET UP MODE      ";
1140 PRINT USING "#####";G;
1150 CP=1056
1160 NP=1056

```

```

1170 PC=PEEK(CP)
1180 POKE CP,24
1190 IK$=INKEY$
1200 IF IK$="" THEN 1190
1210 IF IK$="R" THEN RUN80
1220 IF ASC(IK$)=12 THEN 1110
1230 IF ASC(IK$)<>13 THEN 1270
1240 IF PC=N THEN POKE CP,M:GOTO 1170
1250 POKE CP,N
1260 GOTO 1170
1270 IF ASC(IK$)=94 THEN NP=NP-32:GOTO 1320
1280 IF ASC(IK$)=10 THEN NP=NP+32:GOTO 1320
1290 IF ASC(IK$)=08 THEN NP=NP-01:GOTO 1320
1300 IF ASC(IK$)=09 THEN NP=NP+01:GOTO 1320
1310 IF IK$="S" THEN 1370
1320 POKE CP,PC
1330 IF NP<1056 THEN CP=NP+480:NP=CP:GOTO 1360
1340 IF NP>1535 THEN CP=NP-480:NP=CP:GOTO 1360
1350 CP=NP
1360 GOTO 1170
1370 G=1
1380 PRINT@0,"LIFE          RUN MODE      ";
1390 PRINT USING "#####";G;
1400 EXEC 25000
1410 IK$=INKEY$
1420 IF ST=1 AND IK$="C" THEN ST=0
1430 IF ST=0 AND IK$="G" THEN ST=1
1440 IF ST=1 AND IK$="" THEN 1410
1450 IF IK$="" THEN 1480
1460 IF ASC(IK$)=12 THEN 1110
1470 IF IK$="R" THEN RUN80
1480 G=G+1
1490 GOTO 1380
1500 CLS2
1510 PRINT@43,"## LIFE ##";
1520 PRINT@77," THIS VERSION OF  THE GAME OF ";
1530 PRINT@129," LIFE  USES A  MACHINE CODE ";
1540 PRINT@161,"ROUTINE TO SPEED UP THE CYCLE ";
1550 PRINT@193,"TIME PER GENERATION.  YOU ARE ";
1560 PRINT@225,"ASKED TO ENTER CHOICE OF BACK-";
1570 PRINT@257,"GROUND COLOUR,CELL SYMBOL, AND";
1580 PRINT@289,"WHETHER YOU WANT TO HOLD EACH ";
1590 PRINT@321,"GENERATION DISPLAY OR WATCH A ";
1600 PRINT@353,"CONTINUOUS RUN.  THERE ARE 3 ";
1610 PRINT@385,"MODES: INITIAL- ENTRIES&OPTIONS";
1620 PRINT@417,"          SET UP - START CELLS ";
1630 PRINT@449,"          RUN   - CELL GENERATION";
1640 PRINT@488," PRESS ANY KEY ";
1650 KB$=INKEY$:IF KB$="" THEN 1650
1660 RETURN

```


3. Boating

General Description

The object of this next simulation is to compel the user to consider logical problem solving techniques in action in a (rather unlikely) real life situation.

User instructions appear on the screen as soon as the program is run. The object of the exercise is to transport two men and two girls across a river. But there's a problem. The only available boat has a very limited capacity: at any one time it can only carry a maximum of one man or two girls. What is the minimum number of times that the boat has to cross the river before all four are on the other side?

Detailed Description

Lines 10–270 The main structure which sets up the variables and draws the river. The other procedures are controlled from here.

280–490 Display of instructions.

500–600 Displays Gs and Ms, also displays number of rides taken.

610–830 Get input from keyboard, e.g. G, GG or M, and validate. Also move person across river in memory but not visually.

840 Data for validation of keyboard input.

850–1000 Move boat across the river visually.

1010–1070 Check to see if everybody is on the same side of the river but not on the same side as where they started, if so, then go to finish.

1080–1170 Routine for finish, displaying congratulatory message.

Educational Notes

It is often difficult to demonstrate to students that logic has a place in the real world, and that many problems are *not* most effectively dealt with by means of intuition or commonsense. Textbook examples often tend to reinforce the suspicion that logic is mathematics in the esoteric mode. This program has been designed to breathe a little life into logical examples.

Program Listing

```
10 REM*****
20 REM*          BOATING          *
30 REM*          *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*          *
60 REM*    BY    RAY NEW    1983  *
70 REM*****
80 GOSUB 290 'INTRO
90 C=0
100 NR=0
110 S$="L":OS$=""
120 DIM P$(4)
130 FOR T=1 TO 4
140 P$(T)="L"
150 NEXT T
160 CLS
170 FOR X=0TO63
180 FOR Y=10TO 21
190 SET(X,Y,3)
200 NEXT Y
210 NEXT X
220 GOSUB 510 'DISPLAY
230 GOSUB 620 'KEYBOARD
240 GOSUB 860 'BOAT
250 GOSUB 1020 'CHECK FINISH
260 D$=S$:S$=OS$:OS$=D$:NR=NR+1
270 GOTO220
280 'INTRODUCTION*****
290 CLS
300 PRINT"-----BOATING-----";
310 PRINT"  THE IDEA OF THIS GAME IS TO"
320 PRINT"GET THE TWO GIRLS AND TWO MEN"
330 PRINT"ACROSS THE RIVER IN A SMALL BOAT";
340 PRINT"THE BOAT CAN CARRY A MAXIMUM OF"
350 PRINT"2-GIRLS OR 1-MAN."
360 PRINT"SO AT ANY TIME THE BOAT CAN"
370 PRINT"CARRY 1-MAN,1-GIRL OR 2-GIRLS."
```

```

380 PRINT"I THINK THE MINIMUM NUMBER OF"
390 PRINT"CROSSES IS 9,SEE IF YOU CAN"
400 PRINT"BEAT IT."
410 PRINT"    TO MOVE PEOPLE ACROSS THE"
420 PRINT"RIVER TYPE 'G' FOR 1 GIRL,'GG'"
430 PRINT"FOR 2 GIRLS AND 'M' FOR 1 MAN."
440 PRINT
450 PRINT" DO YOU WISH TO STOP(Y/N) ?";
460 A$=INKEY$
470 IF A$<>"N" AND A$<>"Y"THEN460
480 IF A$="Y"THEN END
490 RETURN
500 'DISPLAY*****
510 PRINT@142,"    ";
520 PRINT@366,"    ";
530 FOR T=1TO4
540 IF P$(T)="L"AND T<3THENPRINT@141+T,"g";
550 IF P$(T)="R"AND T<3THENPRINT@365+T,"g";
560 IF P$(T)="L"AND T>2THENPRINT@141+T,"m";
570 IF P$(T)="R"AND T>2THENPRINT@365+T,"m";
580 NEXT T
590 PRINT@7,"NUMBER OF RIDES=";NR
600 RETURN
610 'KEYBOARD*****
620 M$=""
630 PRINT@466,"    ";
640 PRINT@457,"";
650 INPUT"MOVE > ";M$
660 L=LEN(M$)
670 IF L=0 OR L>2THEN730
680 RESTORE
690 FOR T=1TO3
700 READ D$
710 IF D$=M$THENE=T:T=3:NEXTT:GOTO750
720 NEXTT
730 PRINT@466,"    "
740 GOTO620
750 FOR Q=1TO4
760 T=E
770 IF P$(Q)<>S$THEN810
780 IFT=1 AND Q<3 THEN P$(Q)=OS$:Q=4:NEXTQ:GOTO830
790 IFT=2 AND Q>2 THEN P$(Q)=OS$:Q=4:NEXTQ:GOTO830
800 IFT=3 AND P$(1)+P$(2)=S$+S$ THEN P$(1)=OS$:P$(
2)=OS$:Q=4:NEXTQ:GOTO830
810 NEXTQ
820 GOTO730
830 RETURN
840 DATA G,M,GG,
850 'BOAT*****
860 IF S$="R"THEN940
870 FOR Y=10TO21
880 SET(29,Y,2)

```

```

890 PLAY"T5001AA03CC05EE"
900 FOR I=1TO200:NEXTI
910 SET(29,Y,3)
920 NEXTY
930 GOTO1000
940 FOR Y=21TO10STEP-1
950 SET(29,Y,2)
960 PLAY"T5005EE03CC01AA"
970 FOR I=1TO200:NEXTI
980 SET(29,Y,3)
990 NEXTY
1000 SOUND100,3
1010 ^CHECK FINISH*****
1020 C=0
1030 FOR T=1TO4
1040 IF P$(T)="R"THEN C=C+1
1050 NEXTT
1060 IF C=4 GOSUB 1090
1070 RETURN
1080 ^FINISH*****
1090 PRINT@43,"WELL DONE";
1100 PRINT@72,"YOU CROSSED THE";
1110 PRINT@105,"RIVER";NR+1;"TIMES";
1120 FOR I=1TO2000
1130 KB$=INKEY$:IFKB$=""THEN1150
1140 GOTO 90
1150 NEXT
1160 CLS:PRINT@78,"BYE!":END
1170 RETURN

```


4. Word Find Tester

General Description

This program tests the ability of children to see words within words, randomly accessing a data file of 'test' words. It subsequently displays a series of randomly chosen smaller words inviting a 'yes/no' response from the user as to whether the smaller word can be made from the test word. It checks each word before accepting the user's response and provides laudatory sounds and flashing congratulations if the user is successful. The user is invited each time to progress to the next word or may exit to the results display. This provides a display of the words, along with three stars for the correct answers. Users are meant to add up how many correct answers they scored and the computer will indicate whether it agrees with their assessment. The 'game' may be replayed.

It is important that the educational notes are read with this program as it can, with suitably chosen data, be made to do many things. The data that is given is sample data, and is a little too extensive for proper educational value.

Detailed Description

Lines 10–310 This is the main structure of the program calling several dependent procedures.

320–590 This displays the introductory messages for the program. Change the sound if you find it offensive, but I have found that sound is as important as visual stimulation in keeping interest levels high.

600–640 This chooses the testword that is going to be displayed on the screen. The data is kept in data statements from line 2460. It would be possible to keep an extensive data file and through judicious use of the random number in line 630 restrict access to subsets of the data.

650-740 This hardly needed to be placed in a separate procedure but does allow for the respond routine to be altered without affecting this routine.

750-920 This displays the remainder of the instructions and then invites a reply (yes or no) as to whether the smaller word can be made from the larger. A flag 'REPLY' is set which is used in the check procedure in line 890.

930-980 This generates delays depending on the value of WAIT.

990-1100 This routine generates the screen border to the colour indicated by COL.

1110-1170 Reads the testwords into an array.

1180-1280 This routine is used to decide whether to progress to the next testword. I could have arranged automatic progression, but feel the user must be encouraged to interact with the machine at all times. Moreover, an assumption is made, if progression is automatic, that the user will not be bored in the middle of the 'game'. Concentration spans vary enormously.

1290-1660 The display of the results for the teacher, again the user is asked to interact with the computer. Not only are there stars against correct results but the user is required to add up the number of words he thinks were right. The computer adds a discrete 'also' line 1620 if it agrees.

1670-1870 The menu hardly merits the title 'menu'. It is an attempt to make the computer seem friendly and to discourage the user from repeatedly coming back to the same program by indicating that it, the computer, gets tired.

1880-1990 This procedure randomly accesses the data for a smaller word that can be made from the larger. Note this data starts at line 2560, and the dummy reads to bypass the testwords. The first real read obtains a number which is used as the parameter in the random number generator, this number used judiciously can allow selective retrieval of data.

2000-2180 This routine contains the only real processing in the program. Each letter in the smaller word is checked off against the testword. If an unused letter in the testword is found it is exchanged for another character, a '9', and the routine continues the search. A 'checkflag' is set depending on the reply from the user and whether there exists a usable letter in the testword. The checkflag is used to signal suitable success or failure sound as well as deciding what message is displayed and

putting the relevant stars into the array WORD\$ all in the next routine.

2190–2340 This uses the checkflag, called CF in the procedure, to provide the sounds and messages outlined above.

2350–2450 Again self evident . . . a GOODBYE message is displayed.

2460–2550 Main testword data.

2560–2860 Smaller testword data.

Educational Notes

As it stands, with the data given the program is entertaining and varied but lacks a structured progression for learning. This can be provided through sensible use of the data.

Firstly the larger testwords should be kept in a family of words: e.g. FAMILIAR, FAMILIES etc. The smaller words should be a subset of the same letters with some unacceptable words added: e.g. FAME, FAMINE, FAIR, LIES, FILE, FLIES, FLAME, FLEES, FILL etc. From using the program it seems that the children respond better to being able to say 'yes it does not fit in' rather than 'not it does not'. Experiment for yourself, but as with the anagram program, the power of the program depends very much on the data used. Build up a series of similar programs using different data. They can be suitably graded. Remedial 11 year olds were not bored for 30 minutes' work, but towards the end tended to look for letter agreement between the two words rather than word shape agreement.

Program Listing

```
10 '*****
20 '*      WORD FINDER TEST      *
25 '*                                     *
30 '*  WRITTEN FOR THE DRAGON  *
35 '*                                     *
40 '*    BY      RAY NEW      1983 *
50 '*****
60 CLEAR 2000
70 DIM SWORDS$(20)
80 DIM WORD$(40)
90 GOSUB 1140 'READ WORDS
100 P1$="T1004AACCEE66"
```

```

110 P2$="T1001AACCEEGG"
120 P3$="T2005ABCDGEFFEGDCA"
130 P4$="T401AABBBCCC"
140 CLS
150 F1=0:F2=0
160 CF=0:REPLY=0:SLOT=0:SCORE=0
170 GOSUB 350 'INTRO
180 CLS
190 GOSUB 630 'WORD
200 GOSUB 690 'SEEWORD
210 GOSUB 810 'RESPOND
220 WAIT=10
230 GOSUB 960 'DELAY
240 F3=0
250 GOSUB 1210 'GIVE UP
260 IF F1=0 AND SLOT<=23 THEN 180
270 GOSUB 1320 'SEE RESULT
280 GOSUB 1700 'MENU
290 IF F2=0 THEN 140
300 GOSUB 2380 'END
310 END
320 REM #
330 REM INTRODUCTION TO PROGRAM
340 REM #
350 REPLY$=""
360 COL=6
370 GOSUB 1020 'BORDER
380 PRINT@43,"WORD SPOT";
390 PLAY P1$
400 WAIT=4
410 GOSUB 960 'DELAY
420 SOUND200,3
430 PRINT@102,"YOU WILL SEE A WORD";
440 WAIT=5
450 GOSUB 960 'DELAY
460 SOUND200,3
470 PRINT@165,"AND THEN YOU WILL SEE";
480 PRINT@197,"OTHER WORDS WHICH CAN";
490 GOSUB 960 'DELAY
500 SOUND200,3
510 PRINT@261,"OR EVEN CANNOT FIT IN";
520 PRINT@293,"THE WORD THAT YOU SEE";
530 GOSUB 960 'DELAY
540 SOUND200,3
550 D$=INKEY$
560 PRINT@450,"WHEN YOU UNDERSTAND TYPE 'Y'";
570 IF INKEY$<>"Y" THEN 570
580 SOUND225,2
590 RETURN
600 REM #
610 REM WORD SELECTION PROCEDURE
620 REM #

```



```

630 TWORD$=SWORDS$(RND(20))
640 RETURN
650 REM #
660 REM DISPLAY TESTWORD AND
670 REM MESSAGE FOR THE LUCKLESS
680 REM #
690 CLS
700 COL=2
710 GOSUB 1020 'BORDER
720 PRINT@64+(32-LEN(TWORD$))/2,TWORD$;
730 PLAY P1$
740 RETURN
750 REM #
760 REM GET THE CHILDS RESPONSE
770 REM TO RANDOMLY SELECTED
780 REM WORDS WHICH COULD FIT
790 REM IN THE DISPLAYED WORD
800 REM #
810 GOSUB 1910 'GET WORD
820 REPLY$=""
830 PRINT@161,"CAN THE WORD ";GWORD$;" BE MADE";
840 D$=INKEY$
850 PRINT@193,"FROM ";TWORD$;" ABOVE ?";
860 REPLY$=""
870 REPLY$=INKEY$
880 IF REPLY$<>"N"AND REPLY$<>"Y" THEN 870
890 IF REPLY$="Y"THEN PRINT" YES";:REPLY=1:SOUND 2
25,2 ELSE PRINT" NO";:REPLY=0:SOUND 25,2
900 GOSUB 2030 'CHECK
910 GOSUB 2250 'PICTURE
920 RETURN
930 REM #
940 REM DELAY PROCEDURE
950 REM #
960 FOR I=1TO WAIT*400
970 NEXTI
980 RETURN
990 REM #
1000 REM SCREEN BORDER
1010 REM #
1020 FOR I=0 TO 63
1030 SET(I,0,COL)
1040 SET(I,31,COL)
1050 NEXTI
1060 FOR I=0 TO 31
1070 SET(0,I,COL)
1080 SET(63,I,COL)
1090 NEXT I
1100 RETURN
1110 REM #
1120 REM READ WORDS INTO STORE
1130 REM #

```

```

1140 FOR I=1 TO 20
1150 READ SWORDS$(I)
1160 NEXT I
1170 RETURN
1180 REM #
1190 REM GIVE UP PROCEDURE
1200 REM #
1210 PRINT@449,"DO YOU WISH TO CONTINUE ?";
1220 D$=INKEY$
1230 REPLY$=INKEY$
1240 IF REPLY$<>"N"AND REPLY$<>"Y" THEN 1230
1250 IF REPLY$="Y"THEN PRINT" YES";:F1=0:SOUND225,
2 ELSE PRINT" NO";:F1=1:SOUND25,2
1260 WAIT=1
1270 GOSUB 1020 'DELAY
1280 RETURN
1290 REM #
1300 REM DISPLAY RESULTS FOR TEACHER
1310 REM #
1320 Y=0:ADD$=""
1330 CLS
1340 COL=8
1350 GOSUB1020
1360 PRINT@44,"results";
1370 PRINT@65,"THREE STARS MEAN YOU GOT THE";
1380 PRINT@97,"ANSWER CORRECT IN THE TEST";
1390 FOR X=1TOSLOT STEP 3
1400 FOR Z=0TO2
1410 PRINT@97+(32*(X+2))/3+(10*Z),WORD$(X+Z);
1420 IF X+Z=SLOT THEN Z=2
1430 NEXTZ
1440 NEXTX
1450 PRINT@385,"NOW ADD UP THE ONES";
1460 PRINT@417,"YOU GOT RIGHT >>> ";
1470 D$=INKEY$
1480 K$=INKEY$
1490 IF K$=""THEN1480
1500 IF ASC(K$)=13THEN K$="0":GOTO1580
1510 IF K$<"0"ORK$>"9"THEN1480
1520 PRINT@436,K$;
1530 K1$=INKEY$
1540 IF K1$=""THEN 1530
1550 IF ASC(K1$)=13THEN K1$="0":GOTO1600
1560 IF K1$<"0"ORK1$>"9"THEN1530
1570 PRINT@437,K1$;
1580 KSCORE=VAL(K$)*10+VAL(K1$)
1590 GOTO1610
1600 KSCORE=VAL(K$)
1610 PRINT@449,"I THINK YOU GOT ";SCORE;
1620 IF KSCORE=SCORE THEN PRINT" ALSO !";
1630 WAIT=12
1640 GOSUB 960 'DELAY

```

```

1650 PLAY F1$
1660 RETURN
1670 REM #
1680 REM MENU
1690 REM #
1700 CLS
1710 COL=7
1720 GOSUB 1020 'BORDER
1730 PRINT@45,"menu";
1740 PRINT@129,"YOU HAVE NOW FINISHED THE TEST";
1750 PRINT@161,"BUT OF COURSE YOU MAY HAVE";
1760 PRINT@193,"ANOTHER GO WITH ANOTHER WORD.";
1770 WAIT=5
1780 GOSUB 960 'DELAY
1790 PRINT@257,"I GET TIRED AFTER TOO LONG";
1800 PRINT@289,"AT THIS GAME,WHAT DO YOU THINK";
1810 PRINT@353,"SHALL WE GO ON ?";
1820 D$=INKEY$
1830 REPLY$=INKEY$
1840 IF REPLY$<>"N"AND REPLY$<>"Y"THEN1830
1850 IF REPLY$="N"THEN F2=1:PRINT " NO";:SOUND25,3
ELSE F2=0:PRINT " YES";:SOUND225,3
1860 WAIT=2:GOSUB 960 'DELAY
1870 RETURN
1880 REM #
1890 REM GET SMALL WORD
1900 REM #
1910 RESTORE
1920 FOR I=1TO20
1930 READ DUMMY$
1940 NEXTI
1950 READ X
1960 FOR I=1 TO RND(X)
1970 READ GWORD$
1980 NEXT I
1990 RETURN
2000 REM #
2010 REM CHECK THE CHOOSEN WORD
2020 REM #
2030 CF=1
2040 L=LEN(GWORD$)
2050 LL=LEN(TWORDS$)
2060 AWORD$=TWORD$
2070 X=0
2080 X=X+1
2090 L$=MID$(GWORD$,X,1)
2100 Y=INSTR(1,AWORD$,L$)
2110 IF Y=0THEN2130
2120 AWORD$=LEFT$(AWORD$,Y-1)+"9"+RIGHT$(AWORD$,LL-Y)
2130 IF Y=0 AND REPLY=1 THEN CF=0:X=L
2140 IF Y=0 AND REPLY=0 THEN CF=1:X=L

```



```

2150 IF X<L THEN 2080
2160 IF Y>0 AND REPLY=0 THEN CF=0
2170 IF Y>0 AND REPLY=1 THEN CF=1
2180 RETURN
2190 REM #
2200 REM DISPLAY COMPUTER
2210 REM ANALYSIS OF CHILDS
2220 REM RESPONSE AND FOR FUTURE
2230 REM DISPLAY FOR THE TEACHER
2240 REM #
2250 IF CF=1 THEN PLAY P3$: SCORE=SCORE+1 ELSE PLAY
  P4$
2260 SLOT=SLOT+1
2270 WORD$(SLOT)=GWORD$
2280 IF CF=0 THEN 2330
2290 IF L=7 THEN 2320
2300 FOR I=1 TO 7-L
2310 WORD$(SLOT)=WORD$(SLOT)+" ":NEXT I
2320 WORD$(SLOT)=WORD$(SLOT)+"****"
2330 IF CF=1 THEN PRINT@363,"excellent"; ELSE PRIN
T@363,"< WRONG >";
2340 RETURN
2350 REM #
2360 REM GOODBYE
2370 REM #
2380 CLS
2390 COL=1
2400 GOSUB 1020 'BORDER
2410 PRINT@236,"goodbye";
2420 PLAY P1$
2430 WAIT=5
2440 GOSUB 960 'DELAY
2450 RETURN
2460 REM #
2470 REM THIS DATA CAN BE CHANGED
2480 REM #
2490 DATA THEATRES,THERMAL,HEATERS
2500 DATA THREESOME,TIRESOME,THREADBARE
2510 DATA THROUGHOUT,THOROUGH,THOUGHT
2520 DATA TOUGHEN,TERRIBLE,TERRIFIED
2530 DATA FRIGHTENED,THREATENED,TREATED
2540 DATA FRICTION,FRACTIONS,TRACTION
2550 DATA STEALTHY,THIRSTY
2560 REM #
2570 REM SO CAN THIS
2580 REM #
2590 REM MAXIMUM LENGTH=7
2600 DATA 104
2610 DATA HEAT,TREAT,TREE,EAT
2620 DATA ARM,HARM,ERA,REAL
2630 DATA MEAL,TERM,MALE,HELM
2640 DATA HER,MARE,HOME,SOME

```

2650 DATA REST, HERE, SHEER, MERE
 2660 DATA TIE, TRIES, THREAD, READ
 2670 DATA BARE, THREAD, BEAD, BED
 2680 DATA BAD, BREAD, RARE, DARE
 2690 DATA RED, REAR, ROUGH, OUT
 2700 DATA GOUT, TROUT, ROT, ROUT
 2710 DATA TOUT, TOR, TROT, TROUGH
 2720 DATA TUG, OUGHT, HEN, THEN
 2730 DATA TEN, HOG, HUG, HOT
 2740 DATA RITE, ERR, RIB, BEER
 2750 DATA BIT, FRIED, FED, RIFT
 2760 DATA FRIGHT, RIGHT, RIG, FIGHT
 2770 DATA TENT, NET, NEED, TENDER
 2780 DATA DEN, DEATH, TONIC, TIN
 2790 DATA TIC, FICTION, TON, RATION
 2800 DATA RAT, FAT, CAT, FAST
 2810 DATA CAST, AFT, RAFT, FONT
 2820 DATA FIT, ACT, FACT, STEAL
 2830 DATA SHY, SLY, STY, STEEL
 2840 DATA HEAL, TEST, LAST, TASTY
 2850 DATA HASTY, SHIRT, SIR, HIT
 2860 DATA SIT, NOT, HAD, TAR

5. Vector Race Game

General Description

This program is meant to show how vectors can be used to direct the position of a car round a race track. General instructions are seen when the program is started. The names of the players are requested, and a warning is given that the vectors are 'cumulative' and indicate 'speed' and direction. This means that the next position is determined by your previous vector added to your currently chosen vector. The game is single key stroke entry, except when putting in your names, when the return key must be used. The 'delete' has not been allowed when selecting your vector . . . i.e. get it right. The vectors are displayed vertically without the enclosing brackets. You might wish to enhance this.

When playing the game the person whose go it is has up to 15 seconds before the race track gives way to a text screen displaying their previous vector and asking for the next. The 15 seconds may be shortened by entering any key. The track of the car is left on the race track as a blue and yellow echo and though it is theoretically possible, as the program is written, to crash on the yellow, no third year has yet done so.

Detailed Description

Lines 10–260 The main structure of the program: nothing surprising. The only important point is that $CAR = CAR + 1$ must go at line 170 and not earlier or later as procedure 'Get Vector' assumes CAR values 0 and 1 rather than 1 and 2.

270–550 The race track is drawn.

560–710 The 'X' and 'Y' values in line 600 will hold the previous position of the car. The 'A\$' array holds the names of the contestants – indeed it is used for several functions.

720–840 If you change the start positions, then you will need to change lines (390–400) and 710.

850–1230 Information is displayed on this text screen inviting the next player to enter their vector. The vector is validated and restricted to a number between -9 and 9 . The vector is collected as a single character and translated into a number subsequently.

1240–1360 The car is checked for off limits by a colour test i.e. not green or blue. The collision check sees if the cars are within 2 pixels of each other. On the basis of experiments, this seems acceptable, but all that needs to be changed are the values of 2 in line 1340 if you are not satisfied. The victory check assumes that each car has done at least 10 moves. It is possible to get round in 10 moves so the quick reverse freak has nothing to gain. The victory check makes certain that your final position is within reasonable limits.

1370–1460 The new position of the car is displayed and a blue line used to join it to the old one. The moves are counted to stop someone cheating by reversing straight over the finish line (see above).

1470–1510 The victory check.

1520–1980 The instructions here are free to change. I have noticed that they do not say that the vectors are limited from -9 to 9 , but when I have loaded the game for the class I have dropped a word in their ears and they seem to have no problems.

1990–2780 Self evident. The collision, crash, victory and play again procedures.

2790–2900 A checkered flag border for the text screens.

Educational Notes

I have used this to teach vectors to low ability third years. The idea is developed from an old one which we do with pencil and paper and it helps to play the game on paper first to acquaint them with the concept of vectors. I do not recommend the game as a panacea for actual teaching, but once they have mastered the simple paper track, the entertainment value of the game shows its worth.

Because we do not have enough machines, they play in groups of four a mini-league between them. It *would* be possible to do a single class demonstration, but then some of the messages would need to be changed.

Program Listing

```
10 *****
20 * THE VECTOR RACE GAME *
25 *
30 * WRITTEN FOR THE DRAGON *
35 *
40 * BY RAY NEW 1983 *
50 *****
60 GOSUB 1550 'INTRO
70 GOSUB 590 'INITIALISE
80 GOSUB 300 'DRAW SCREEN
90 GOSUB 750 'START POSITIONS
100 CAR=0
110 FOR I=1 TO 255
120 IF INKEY$<>"" THEN I=255
130 SOUND I,1
140 NEXT I
150 VT=CAR:GOSUB 880 'GET VECTOR
160 SCREEN 1,0
170 CAR=CAR+1
180 GOSUB 1270 'CHECK LIMITS
190 GOSUB 1400 'DRAW CAR
200 IF CRASH=0 AND VICTORY=0 AND COLLIDE=0 AND CAR
<>2 THEN110
210 IF CRASH=0 AND VICTORY=0 AND COLLIDE=0 THEN100
220 IF COLLIDE=1 THEN GOSUB 2020:GOTO250
230 IF CRASH=1 THEN GOSUB 2220:GOTO250
240 GOSUB 2380 'VICTORY
250 GOSUB 2660 'FINAL
260 END
270 REM #
280 REM DRAW RACE TRACK
290 REM #
300 PMODE1,1
310 COLOR2,4
320 PCLS1
330 SCREEN1,0
340 LINE(0,0)-(255,191),PSET,B
350 LINE(40,40)-(215,151),PSET,B
360 LINE(10,10)-(245,181),PSET,B
370 PAINT(42,42),4,2
380 PAINT(2,2),4,2
390 DRAW"BM3,100;E4H4"
400 DRAW"BM49,100;H4E4"
410 DRAW"BM57,49"
420 DR=0
430 DRAW"D4F4E4U4BR4NR4D3NR6D4R8" 'VE
440 DRAW"BR4NR4B4R8" 'C
450 DRAW"BR4R8L4D8" 'T
460 DRAW"BR4B4R8D8NL8" 'D
470 DRAW"BR4U8R8D4L8F4R4" 'R
```

```

480 DRAW"BD4L8D8U4R8U4D8" 'A
490 DRAW"BD4L8D8R8" 'C
500 DRAW"BD4L8D4NR8D4R8" 'E
510 DRAW"BM93,81"
520 IF DR=0 THEN DR=1: DRAW"BM93,73": GOTO430
530 IF DR=1 THEN DR=2: DRAW"BM129,97": GOTO430
540 COLOR 3,2
550 RETURN
560 REM #
570 REM INITIALISE
580 REM #
590 CAR=0
600 MOVES=0: X1=0: Y1=0: X2=0: Y2=0
610 CRASH=0
620 VICTORY=0
630 COLLIDE=0
640 DIM PS(5), SPEED(5)
650 A$(0)=" FIRST"
660 A$(1)="SECOND"
670 FOR I=1 TO 4
680 READ PS(I)
690 NEXT I
700 RETURN
710 DATA 20,96,30,96
720 REM #
730 REM START POSITIONS
740 REM #
750 X=PS(1): Y=PS(2)
760 X1=X: Y1=Y
770 CAR=1
780 GOSUB 1400 'DRAW CAR
790 X1=X
800 X=PS(3): Y=PS(4)
810 X2=X: Y2=Y
820 CAR=2
830 GOSUB 1400 'DRAW CAR
840 RETURN
850 REM #
860 REM GET VECTORS
870 REM #
880 CLS
890 PRINT
900 GOSUB 2820 'DRAW BORDER
910 PRINT@39,"vector race game";
920 PRINT@71,"-----";
930 VC=2*VT
940 CNT=0
950 PRINT@131,"IT'S ";A$(VT+4);"'S TURN TO MOVE";
960 PRINT@195,"YOUR CURRENT SPEED IS ";SPEED(VC+1)
/10;
970 PRINT@249,SPEED(VC+2)/10;
980 NUMBER=0

```



```

990 PRINT@259+CNT*64,"YOUR VECTOR PLEASE ";
1000 PRINT@293+CNT*64,A$(CNT);" NUMBER      ";
1010 B$=""
1020 PLAY"T50AGA"
1030 A$=INKEY$
1040 IF (A$<"0"OR A$>"9")AND A$<>"-" THEN 1030
1050 IF B$="-"AND A$="-"THEN A$=""
1060 B$=B$+A$
1070 IF B$<>"-"THEN NUMBER=1
1080 PRINTA$;
1090 IF NUMBER=0 THEN1030
1100 CNT=CNT+1
1110 A$(1+CNT)=B$
1120 B$=""
1130 SPEED(VC+CNT)=SPEED(VC+CNT)+10*VAL(A$(1+CNT))
1140 PS(VC+CNT)=PS(VC+CNT)+.5*SPEED(VC+CNT)*((-1)^
CNT)*-1
1150 IF CNT<2 THEN 980
1160 X=PS(VC+1)
1170 Y=PS(VC+2)
1180 SOUND 25,3
1190 PRINT@451,"PRESS ANY KEY TO CONTINUE";
1200 IF INKEY$=""THEN1200
1210 SOUND 225,3
1220 RETURN
1230 RETURN
1240 REM #
1250 REM CHECK LIMITS
1260 REM #
1270 IF X<0 THEN X=0
1280 IF Y<0 THEN Y=0
1290 IF X>255 THEN X=255
1300 IF Y>191 THEN Y=191
1310 IF PPOINT(X,Y)<>1 AND PPOINT(X,Y)<>3 THEN CRA
SH=1
1320 IF CAR=1 THEN LA=ABS(X-X2) ELSE LA=ABS(X-X1)
1330 IF CAR=1 THEN LB=ABS(Y-Y2) ELSE LB=ABS(Y-Y1)
1340 IF LA<2 AND LB<2 THEN COLLIDE=1
1350 IF MOVES>20THEN GOSUB 1500
1360 RETURN
1370 REM #
1380 REM PLOT CAR POSITION
1390 REM #
1400 MOVES=MOVES+1
1410 IF CAR=1 THEN LINE(X1,Y1)-(X,Y),PSET:X1=X:Y1=
Y
1420 IF CAR=2 THEN LINE(X2,Y2)-(X,Y),PSET:X2=X:Y2=
Y
1430 IF CAR=1 THEN DRAW"C2NR2NL2NU2ND2":PSET(X,Y,1
)
1440 IF CAR=2 THENDRAW"C2L2U2R4D4L4":PSET(X,Y,1)
1450 DRAW"C3"

```

```

1460 RETURN
1470 REM #
1480 REM VICTORY CHECK
1490 REM #
1500 IF X>10AND X<40 AND Y>50 AND Y<96 THEN VICTOR
Y=1:WINNER=CAR
1510 RETURN
1520 REM #
1530 REM INTRO
1540 REM #
1550 CLS
1560 GOSUB 2820 'DRAW BORDER
1570 PRINT@39,"vector race game";
1580 PRINT@71,"-----";
1590 PRINT@97,"THIS IS A GAME FOR TWO PLAYERS";
1600 PRINT@129,"YOU WILL EACH CONTROL A VECTOR";
1610 PRINT@161,"CAR ... CAR 1 IS A CROSS AND ";
1620 PRINT@193,"CAR 2 IS A SQUARE.";
1630 PRINT@225,"THE SPEED OF THE CAR IS GIVEN";
1640 PRINT@257,"BY A VECTOR !!";
1650 PRINT@289,"FOR EXAMPLE .. SPEED VECTOR 3";
1660 PRINT@321,"-3";
1670 PRINT@353,"WILL MOVE THE CAR TO A NEW";
1680 PRINT@385,"POSITION 3 TO THE RIGHT AND";
1690 PRINT@417,"3 DOWN.";
1700 PRINT@449,"PRESS ANY KEY TO CONTINUE";
1710 SOUND 25,3
1720 IF INKEY$=""THEN1720
1730 SOUND 225,3
1740 CLS
1750 GOSUB 2820 'DRAW BORDER
1760 PRINT@39,"vector race game";
1770 PRINT@71,"-----";
1780 PRINT@107,"warning";
1790 PRINT@164,"SPEEDS ARE CUMULATIVE !";
1800 PRINT@225,"YOUR TRACK WILL BE LEFT IN";
1810 PRINT@259,"BLUE ON THE SCREEN.";
1820 PRINT@291,"FIRST PLAYER'S NAME PLEASE";
1830 PRINT@328,"";
1840 PLAY"T30AGA"
1850 LINE INPUT A$(4)
1860 IF LEN(A$(4))>8 THEN A$(4)=LEFT$(A$(4),8)
1870 POKE 1375,198
1880 PRINT@355,"SECOND PLAYER'S NAME PLEASE";
1890 PRINT@392,"";
1900 PLAY"T30AGA"
1910 LINE INPUT A$(5)
1920 IF LEN(A$(5))>8 THEN A$(5)=LEFT$(A$(5),8)
1930 POKE1439,198
1940 PRINT@451,"PRESS ANY KEY TO CONTINUE";
1950 SOUND 25,3
1960 IF INKEY$=""THEN1960

```

```

1970 SOUND 225,3
1980 RETURN
1990 REM #
2000 REM COLLIDE
2010 REM #
2020 FOR SD=160 TO 1 STEP -10
2030 SOUND SD,1
2040 IF DC=1 THEN DRAW"C1":DC=0 ELSE DRAW"C4":DC=1
2050 DRAW"BM57,121;NR8D8R8" 'C
2060 DRAW"BR4U8R8D8NL8" 'O
2070 DRAW"BR4NU8R8" 'L
2080 DRAW"BR4NU8R8" 'L
2090 DRAW"BR4NU8" 'I
2100 DRAW"BR4R8U4L8U4R8" 'S
2110 DRAW"BR4D8" 'I
2120 DRAW"BR4U8R8D8NL8" 'O
2130 DRAW"BR4U8F8U8" 'N
2140 NEXT SD
2150 PLAY"DDD"
2160 IF CAR=1 THEN WINNER=2 ELSE WINNER=1
2170 GOSUB 2380 'VICTORY
2180 RETURN
2190 REM #
2200 REM CRASH
2210 REM #
2220 FOR SD=160 TO 1 STEP -10
2230 SOUND SD,1
2240 IF DC=1 THEN DRAW"C1":DC=0 ELSE DRAW"C4":DC=1
2250 DRAW"BM57,121;NR8D8R8" 'C
2260 DRAW"BR4U8R8D4L8F4R4" 'R
2270 DRAW"BR4U8R8D4NL8D4" 'A
2280 DRAW"BR4R8U4L8U4R8" 'S
2290 DRAW"BR4D8U4R8U4D8" 'H
2300 NEXT SD
2310 PLAY"ABC"
2320 IF CAR=1 THEN WINNER=2 ELSE WINNER=1
2330 GOSUB 2380 'VICTORY
2340 RETURN
2350 REM #
2360 REM VICTORY
2370 REM #
2380 IF COLLIDE=0 AND CRASH=0 THEN PLAY"T20ABCDEFGH
FEDCBA" ELSE GOTO 2490
2390 FOR SD=160 TO 1 STEP -10
2400 SOUND SD,1
2410 IF DC=1 THEN DRAW"C1":DC=0 ELSE DRAW"C4":DC=1
2420 DRAW"BM57,121;NU8E4F4NU8" 'W
2430 DRAW"BR4NU8" 'I
2440 DRAW"BR4U8F8NU8" 'N
2450 DRAW"BR4U8F8NU8" 'N
2460 DRAW"BR4U8NR8D4NR7D4R8" 'E
2470 DRAW"BR4U8R8D4L8F4R4" 'R

```

```

2480 NEXT SD
2490 CLS
2500 GOSUB 2820 'DRAW BORDER
2510 PRINT@39,"vector race game";
2520 PRINT@71,"-----";
2530 PRINT@138,"well done";
2540 PRINT@196,"CAR";WINNER;"HAS WON THIS RACE";
2550 PRINT@263,"CONGRATULATIONS TO ";
2560 PRINT@330,A$(WINNER+3);
2570 PRINT@451,"PRESS ANY KEY TO CONTINUE";
2580 SOUND 25,3
2590 DU$=INKEY$
2600 IF INKEY$=""THEN2600
2610 SOUND 225,3
2620 RETURN
2630 REM #
2640 REM FINAL
2650 REM #
2660 CLS
2670 GOSUB 2820 'DRAW BORDER
2680 PRINT@39,"vector race game";
2690 PRINT@71,"-----";
2700 DU$=INKEY$
2710 PRINT@229,"DO YOU WANT ANOTHER GO ?";
2720 SOUND 25,3
2730 IK$=INKEY$
2740 IF IK$<>"Y"AND IK$<>"N" THEN 2730
2750 SOUND 225,3
2760 PLAY"ADA"
2770 IF IK$="Y" THEN RUN
2780 RETURN
2790 REM #
2800 REM DRAW BORDER
2810 REM #
2820 FOR I=0 TO 31
2830 POKE1024+I,198
2840 POKE1504+I,198
2850 NEXTI
2860 FOR I=0 TO 15
2870 POKE1024+I*32,198
2880 POKE1055+I*32,198
2890 NEXTI
2900 RETURN

```


6. Rotation

General Description

This is an enormously frustrating game to be played against the computer. The user must key in a two number vector which will manifest itself as a line on the screen. Each of your moves will be mirrored by the computer. Your objective is to locate the centre and force the computer to meet your own line. The user takes the purple line (the start point on the left, if using a black and white screen).

Detailed Description

Lines 80–90 Set up graphics area.

100–200 Fancy pattern.

220–340 Display instructions.

350–370 Perform subroutines to display grid, input move from keyboard, make moves and check if centre is located.

410–550 Subroutine to set up start grid and random starting points.

590–770 Subroutine to make moves and check if it hit the computer's line.

780–820 Indicates the hit, shows the score when finished.

830–870 Finish or play again.

Educational Notes

This is essentially an intelligence test which appears to leave the majority of adults floundering in grave despair. For some reason (!) most youngsters seem to get the hang of it fairly quickly.

Program Listing

```
10 REM*****
20 REM*          ROTATION          *
30 REM*          *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*          *
60 REM* BY LAURENCE OWEN. 1983 *
70 REM*****
80 PCLEAR4
90 PMODE3:PCLS:SCREEN1,1:C=7
100 REM *****FANCY PATTERN*****
110 FOR X=8 TO 255 STEP 16
120 IF C=7 THEN C=8 ELSE C=7
130 COLORC
140 LINE(X,191)-(128,0),PSET
150 LINE(X,0)-(128,191),PSET
160 IF X>191 THEN GOTO190
170 LINE(0,X)-(255,96),PSET
180 LINE(255,X)-(0,96),PSET
190 NEXT
200 FOR D=1 TO 3000:NEXT
210 REM **** INTRODUCTION ****
220 CLS
230 PRINT@10,"*****"
240 PRINT@42,"* ROTATION *"
250 PRINT@74,"*****"
260 PRINT@130,"YOU INPUT A VECTOR TO MOVE;      THE
  VECTOR MUST BE TWO NUMBERS."
270 PRINT@160,"THE COMPUTER WILL REFLECT YOUR  MOV
  E. WHEN YOU THINK YOU KNOW  WHERE THE CENTRE IS,Y
  OU MUST      FORCE THE COMPUTER'S LINE TO      MEET YO
  URS."
280 PRINT@324,"YOU ARE THE PURPLE LINE"
290 PRINT@389,"PRESS ANY KEY TO ENTER"
300 PRINT@425,"YOUR NEXT MOVE"
310 PRINT@489,"PRESS ANY KEY";
320 SCREEN0,1
330 C$=INKEY$
340 A$=INKEY$:IFA$=""THEN GOTO340
350 GOSUB410
360 GOSUB590
370 GOTO360
380 REM *****
390 REM ** SET UP START GRID **
400 REM *****
410 PCLS:COLOR6
420 FOR X=0 TO 248 STEP 8
430 LINE(X,0)-(X,191),PSET
440 LINE(0,X)-(248,X),PSET
450 NEXT
460 PX=RND(8)*8+20:CY=RND(8)*8+164
```

```

470 PY=RND(20)*8+20:CY=RND(20)*8+20
480 CIRCLE(CX,CY),4,8
490 CIRCLE(PX,PY),4,7:PAINT(PX,PY),7,7
500 SCREEN1,1
510 SOUND 250,5
520 MOVES=0
530 FOR D=1TO100:NEXT:C$=INKEY$
540 A$=INKEY$:IFA$=""THEN540
550 RETURN
560 REM *****
570 REM ** REQUEST NEXT MOVE **
580 REM *****
590 MOVES=MOVES+1
600 SCREEN0:CLS:PLAY"T20;ADA"
610 IF OUT=1 THEN OUT=0:PRINT@129,"NO GOOD,YOU'RE
OFF THE SCREEN!"
620 PRINT@192:INPUT" ENTER VECTOR (X,Y)";X,Y
630 X=INT(X*2+0.5)/2:Y=-INT(Y*2+0.5)/2
640 PA=PX+X*8:PB=PY+Y*8
650 CA=CX-X*8:CB=CY-Y*8
660 REM** CHECK IF LEGAL MOVE **
670 IF PA<0 OR PA>247 OR PB<0 OR PB>191 THEN OUT=1
:GOTO600
680 IF CA<0 OR CA>247 OR CB<0 OR CB>191 THEN OUT=1
:GOTO600
690 REM** MAKE MOVES **
700 SCREEN1:PLAY"BABBAGE"
710 COLOR7:LINE(PX,PY)-(PA,PB),PSET
720 COLOR8:LINE(CX,CY)-(CA,CB),PSET
730 PX=PA:PY=PB:CX=CA:CY=CB
740 REM** CHECK IF HIT **
750 IF CX=PX AND CY=PY THEN GOTO780
760 A$=INKEY$:IFA$=""THEN GOTO760
770 RETURN
780 FOR N=1 TO 3
790 CIRCLE(CX,CY),2*N,N+5:SOUND 12*N,2
800 NEXT
810 A$=INKEY$:IFA$="" THEN GOTO810
820 SCREEN0:CLS:PRINT@100,"NUMBER OF MOVES TAKEN =
";MOVES
830 PRINT@200,"CONTINUE? (Y/N)"
840 A$=INKEY$:IF A$<>"Y" AND A$<>"N" THEN GOTO830
850 IF A$="Y" THEN GOTO 350
860 PRINT@294,"THANKS FOR PLAYING!"
870 END

```


7. Building Blocks

General Description

This remarkable program enables the user to create 3-D objects on a 2-D screen. It allows you to construct a picture with building blocks, whose positions are determined by the specifications you give to the computer. Use the L key to expand your creation to the Left, R if you want it developed to the Right and so on. You can choose one of two colour sets and you can change the colour of the blocks used in the chosen colour set. If you want to see the computer draw the shape for you, simply press P for Playback.

Detailed Description

Line 80 Reserves the graphics area and sets the mode.

90 Clear string space; MAX = Maximum number of operations.

100–160 Set up variables.

170–180 Sets start colour of first block, and chosen colour set.

190–200 Perform subroutines to initialise block and to display block on the screen.

210 Imitates keyboard input, for playback.

220–370 Input from keyboard.

380–390 Perform subroutine to display Help screen (i.e. list of commands).

400–430 Add move to C\$(LT) so computer can playback picture later. Also, set colour of block to colour chosen.

470–640 Subroutine to initialise block.

680–820 Subroutine to display block on screen. Also checks if 'out of bounds' and gives audible warning.

860–1000 Subroutine to display Help screen.

1010–1300 Instructions, and input choice of colour set.

Educational Notes

This program is valuable for its ability to bring geometric examples to life. Art teachers will find it useful as an aid to demonstrating the power and techniques of perspective, and computer students will waste hours trying to demonstrate that they have missed their vocation.

Program Listing

```
10 REM*****
20 REM*   BUILDING BLOCKS   *
30 REM*   *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*   *
60 REM* BY LAURENCE OWEN. 1983 *
70 REM*****
80 PCLEAR4:PMODE3
90 CLEAR 1000:MAX=500
100 DIM XA(4),YA(4),XB(4),YB(4)
110 DIM XC(3,3),YC(3,3),C$(MAX)
120 W=128:H=96:G=1
130 M=65
140 GOTO1020
150 Q=1:X=-0.5:B=4.5:Y=-0.5
160 LT=1:S=1
170 CL=2
180 COLOR 2,4+SCR:PCLS
190 GOSUB470      ' INIT CUBE
200 GOSUB680      ' PUT CUBE
210 IF LT<G THEN KB$=C$(LT):GOTO 250
220 G=G+1
230 IF LT>MAX THEN LT=MAX
240 KB$=INKEY$
250 IFKB$="0" THENB=ABS(B-Q):GOTO400
260 IFKB$="I" THENB=B+Q:GOTO400
270 IFKB$="L" THENY=Y-Q:GOTO400
280 IFKB$="R" THENY=Y+Q:GOTO400
290 IFKB$="U" THENX=X-Q:GOTO400
300 IFKB$="D" THENX=X+Q:GOTO400
310 IFKB$=" " THENXC=XC+1:C$(LT)=KB$:LT=LT+1:GOTO2
10
320 IFKB$="2" THENCL=2+SCR:GOTO400
330 IFKB$="3" THENCL=3+SCR:GOTO400
340 IFKB$="1" THENCL=1+SCR:GOTO400
350 IFKB$="P" THENG=G-1:GOTO150
360 IFKB$="C" THEN G=1:GOTO150
370 IFKB$="Q" THEN CLS:SCREEN0:END
380 IFKB$="" THEN240ELSE GOSUB860
```

```

390 GOTO240
400 C$(LT)=KB$:LT=LT+1
410 XC=XC-1:IFXC>.5THEN210
420 COLORCL
430 GOTO 190
440 REM *****
450 REM **      INIT CUBE      **
460 REM *****
470 XA(0)=Y/B
480 XA(1)=XA(0)
490 XA(2)=(Y+Q)/B
500 XA(3)=XA(2)
510 XA(4)=XA(0)
520 YA(0)=X/B
530 YA(1)=(X+Q)/B
540 YA(2)=YA(1)
550 YA(3)=YA(0)
560 YA(4)=YA(0)
570 BB=B/(B+Q):XC=1
580 FORN=0TO4
590 XA(N)=XA(N)*M
600 XB(N)=XA(N)*BB
610 YA(N)=YA(N)*M
620 YB(N)=YA(N)*BB
630 NEXT
640 RETURN
650 REM *****
660 REM **      PUT CUBE      **
670 REM *****
680 SCREEN1,SCR
690 FORN=0TO3
700 XC(N,0)=XA(N+1)+W:YC(N,0)=YA(N+1)+H
710 XC(N,1)=XA(N)+W:YC(N,1)=YA(N)+H
720 XC(N,2)=XB(N)+W:YC(N,2)=YB(N)+H
730 XC(N,3)=XB(N+1)+W:YC(N,3)=YB(N+1)+H
740 IF XC(N,0)>255 OR XC(N,0)<0 THEN OUT=1
750 IF YC(N,0)>191 OR YC(N,1)<0 THEN OUT=1
760 NEXT
770 IF OUT=1 THEN OUT=0:SOUND1,6:GOTO820
780 FOR I=0TO3
790 FOR J=0TO2
800 LINE(XC(I,J),YC(I,J))-(XC(I,J+1),YC(I,J+1)),PS
ET
810 NEXT J,I
820 RETURN
830 REM *****
840 REM **      HELP DISPLAY      **
850 REM *****
860 CLS:SCREEN0
870 PRINT@35,"L..LEFT","R..RIGHT"
880 PRINT@67,"U..UP","D..DOWN"
890 PRINT@99,"O..OUT","I..IN"

```

```

900 PRINT@163,"SPACEBAR..OMITS NEXT BLOCK"
910 PRINT@227,"P..PLAYBACK"
920 PRINT@259,"C..CLEAR AND RE-START"
930 PRINT@291,"Q..QUIT"
940 PRINT@355,"1..GREEN(CYAN)"
950 PRINT@387,"2..YELLOW(MAGENTA)"
960 PRINT@419,"3..BLUE(ORANGE)"
970 PRINT@483,"PRESS ANY KEY TO CONTINUE";
980 A$=INKEY$:IFA$=""THEN980
990 SCREEN1,SCR
1000 RETURN
1010 REM ***** INTRODUCTION *****
1020 CLS
1030 PRINT@6,STRING$(19,"#")
1040 PRINT@38,"# BUILDING BLOCKS #"
1050 PRINT@70,STRING$(19,"#")
1060 PRINT@128," THIS PROGRAM ALLOWS YOU TO"
1070 PRINT" BUILD 3-D OBJECTS ON A 2-D"
1080 PRINT" SCREEN.TO BUILD THE OBJECT,YOU"
1090 PRINT" USE BLOCKS WHICH YOU CAN MOVE"
1100 PRINT" UP,DOWN,LEFT,RIGHT,IN AND OUT."
1110 PRINT:PRINT" YOU CAN SELECT ONE OF TWO"
1120 PRINT" DIFFERENT COLOUR SETS AND"
1130 PRINT" ALSO CHANGE THE COLOUR OF THE"
1140 PRINT" BLOCKS AS YOU BUILD."
1150 PRINT@483,"PRESS ANY KEY TO CONTINUE";
1160 IF INKEY$=""THEN1160
1170 CLS
1180 PRINT:PRINT" TO MOVE,PRESS THE FIRST LETTER"
1190 PRINT" OF THAT DIRECTION:"
1200 PRINT" E.G. TO GO LEFT,PRESS 'L'."
1210 PRINT:PRINT" IF YOU MOVE OFF THE SCREEN,THE"
1220 PRINT" COMPUTER WILL GIVE A WARNING"
1230 PRINT" SOUND - YOU MUST REVERSE YOUR"
1240 PRINT" DIRECTION."
1250 PRINT:PRINT" TO SEE THE COMPUTER DRAW YOUR"
1260 PRINT" OBJECT,PRESS 'P' FOR PLAYBACK."
1270 PRINT@449," IF YOU ARE READY TO BEGIN,"
1280 PRINT@480," SELECT COLOUR SET (0 OR 1)";
1290 A$=INKEY$:IF A$<>"0" AND A$<>"1" THEN 1290
1300 SCR=VAL(A$):GOTO150
1310 END

```


8. 3-D Prisms

General Description

This program has been designed to allow the user to rotate a 3-D prism on the screen. You can create your own prism by altering the data statement in line 1610 (see detailed notes below). The user can choose either irregular sided prisms or regular prisms of up to twelve sides.

Detailed Description

Line 80 Reserve graphics area; clear screen.

90–210 Fancy title.

230 User-defined function to convert degrees to radians.

240 Display instructions, input choice of regular/irregular.

250–350 Asks for number of sides, e.g. a triangular prism has three sides. Asks for angle for each step of rotation.

360–370 Sets resolution and colour set of graphics screen, and displays 'Help' screen.

380–400 Converts angle to radians and sets up variables.

410–420 If irregular prisms selected, performs subroutine to load start co-ordinates from data; else performs subroutine to compute start co-ordinates for regular prisms.

430 Performs subroutine to transfer shape through Z-plane.

440–570 Display each end of prism.

580–660 Join both ends of prism together.

670–770 Get input from keyboard.

780–850 Rotates about X-axis using rotation matrix.

$$X' = X$$

$$Y' = Y \cos \theta + Z \sin \theta$$

$$Z' = Z \cos \theta - Y \sin \theta$$

860–930 Rotates about Y-axis using rotation matrix.

$$X' = X \cos \theta - Z \sin \theta$$

$$Z' = Z \cos \theta + X \sin \theta$$

$$Y' = Y$$

940–1010 Rotate about Z-axis using matrix.

$$X' = X \cos \theta + Y \sin \theta$$

$$Y' = Y \cos \theta - X \sin \theta$$

$$Z' = Z$$

1020–1080 Transfers shape through Z-plane.

1090–1230 Help display.

1240–1400 Instructions.

1410–1470 Loads data for irregular-sided prisms.

1480–1550 Computes data for regular-sided prisms.

1610 Data for irregular prism.

1620 Alternative set of data: this will only be used if line 1610 is deleted or REM'd.

Note: To change shape of prism, change line 1610. The data at line 1610 is in the format

<u>X,Y</u>	,	<u>X,Y</u>	,	<u>X,Y</u>	,	<u>X,Y</u>
First corner		Second corner		Third corner		Fourth corner

Plot shape on graph paper with origin going through middle of the object.

Choose values for co-ordinates carefully, so that screen limits are not exceeded.

Educational Notes

Whilst this program has an obvious role in clarifying descriptions in geometry lesson, it's also a wonderful demonstration of the potential of computer graphics!

Program Listing

```

10 REM*****
20 REM*      3-D PRISMS      *
30 REM*                               *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*                               *
60 REM* BY LAURENCE OWEN 1983 *
70 REM*****
80 PCLEAR4:CLS
90 REM***** FANCY TITLE *****
100 FOR I=169 TO 329 STEP32

```

```

110 FOR J=0 TO 12
120 READ K
130 PRINT@I+J,CHR$(128+K);
140 NEXT J,I
150 DATA 14,12,12,12,15,15,15,15,14,12,12,12,15
160 DATA 31,31,31,27,5,15,15,15,31,31,31,27,5
170 DATA 15,14,12,31,5,14,12,13,31,5,15,31,5
180 DATA 15,31,31,31,5,31,31,7,31,5,15,31,5
190 DATA 14,12,12,31,5,15,15,15,31,4,12,31,5
200 DATA 31,31,31,31,7,15,15,15,31,31,31,31,7
210 FORD=1T05000:NEXT
220 REM*****
230 DEF FNR(X)=X*ATN(1.0)/45
240 GOSUB1250 ' INSTRUCTIONS
250 CLS
260 IF SHAPE=2 THEN PRINT@5,"IR";
270 PRINT@7,"REGULAR SIDED PRISMS"
280 PRINT@64,"HOW MANY SIDES (2-";
290 IF SHAPE=1 THEN PRINT"12)"; ELSE PRINT"6)";
300 INPUT NS
310 IF NS<2 OR NS>12 THEN 250
320 IF NS>6 AND SHAPE=2 THEN 250
330 PRINT@128,"ROTATE IN STEPS OF (1-360)";
340 INPUT AN
350 IF AN<1 OR AN>360 THEN 250
360 PMODE4:COLOR0,5:PCLS
370 GOSUB 1100 'DISPLAY OPTIONS
380 AR=FNR(AN)
390 DIM C(2,NS,3)
400 W=128:H=96:S=30
410 REM**GET START POINTS R/I**
420 ON SHAPE GOSUB 1500,1430
430 GOSUB1030 ' TRANSFER
440 REM***** DRAW SURFACES ****
450 PCLS
460 FORU=1 TO 2
470 FORT=1 TO NS-1
480 X1=W+C(U,T,1)*S
490 Y1=H+C(U,T,2)*S
500 X2=W+C(U,T+1,1)*S
510 Y2=H+C(U,T+1,2)*S
520 LINE(X1,Y1)-(X2,Y2),PSET
530 NEXTT
540 X1=W+C(U,1,1)*S
550 Y1=H+C(U,1,2)*S
560 LINE(X1,Y1)-(X2,Y2),PSET
570 NEXTU
580 REM***DRAW BOTH TOGETHER***
590 FOR T=1 TO NS
600 X1=W+C(1,T,1)*S
610 Y1=H+C(1,T,2)*S
620 X2=W+C(2,T,1)*S

```

```

630 Y2=H+C(2,T,2)*S
640 LINE(X1,Y1)-(X2,Y2),PSET
650 NEXTT
660 SOUND200,4
670 REM***** GET NEXT MOVE *****
680 KB$=INKEY$:IFKB$=""THEN680
690 IFKB$="X"THENGOSUB790'XAXIS
700 IFKB$="Y"THENGOSUB870'YAXIS
710 IFKB$="Z"THENGOSUB950'ZAXIS
720 IFKB$="S"THEN S=S+5:IFS=45 THEN S=10
730 IFKB$="B"THEN COLOR5,0
740 IFKB$="W"THEN COLOR0,5
750 IFKB$="H"THENGOSUB1100'HELP
760 IFKB$="Q"THEN CLS:SCREEN0:END
770 GOTO 450
780 REM***** X-AXIS SUB*****
790 FOR U=1 TO 2
800 FOR T=1 TO NS
810 Y=C(U,T,2):Z=C(U,T,3)
820 C(U,T,2)=(Y*COS(AR))+(Z*SIN(AR))
830 C(U,T,3)=(Z*COS(AR))-(Y*SIN(AR))
840 NEXTT,U
850 RETURN
860 REM***** Y-AXIS SUB*****
870 FOR U=1 TO 2
880 FOR T=1 TO NS
890 Z=C(U,T,3):X=C(U,T,1)
900 C(U,T,1)=(X*COS(AR))-(Z*SIN(AR))
910 C(U,T,3)=(Z*COS(AR))+(X*SIN(AR))
920 NEXTT,U
930 RETURN
940 REM***** Z-AXIS SUB*****
950 FOR U=1 TO 2
960 FOR T=1 TO NS
970 Y=C(U,T,2):X=C(U,T,1)
980 C(U,T,2)=(Y*COS(AR))-(X*SIN(AR))
990 C(U,T,1)=(X*COS(AR))+(Y*SIN(AR))
1000 NEXTT,U
1010 RETURN
1020 REM***** TRANSFER SUB*****
1030 FOR T=1 TO NS
1040 C(2,T,1)=C(1,T,1)
1050 C(2,T,2)=C(1,T,2)
1060 C(2,T,3)=C(1,T,3)+2
1070 NEXTT
1080 RETURN
1090 REM*** HELP DISPLAY SUB***
1100 SCREEN0:CLS
1110 PRINT@43,"3D-PRISMS";
1120 PRINT@97,"X....ROTATE ON X-AXIS";
1130 PRINT@129,"Y....ROTATE ON Y-AXIS";
1140 PRINT@161,"Z....ROTATE ON Z-AXIS";

```



```

1150 PRINT@225,"B....CHANGE TO WHITE ON BLACK";
1160 PRINT@257,"W....CHANGE TO BLACK ON WHITE";
1170 PRINT@321,"S....CHANGE THE SCALE";
1180 PRINT@385,"Q....QUIT";
1190 PRINT@417,"H....DISPLAY THIS PAGE";
1200 PRINT@490,"PRESS ANY KEY";
1210 KB$=INKEY$:IFKB$=""THEN1210
1220 SCREEN1
1230 RETURN
1240 REM*** INSTRUCTIONS SUB***
1250 CLS6
1260 PRINT@41," INSTRUCTIONS ";
1270 PRINT@97," THIS PROGRAM ALLOWS YOU TO ";
1280 PRINT@129," ROTATE A 3-D PRISM. ";
1290 PRINT@161," YOU CAN DRAW EITHER REGULAR ";
1300 PRINT@193," OR IRREGULAR SIDED PRISMS. ";
1310 PRINT@257," IF YOU WISH , YOU CAN MAKE ";
1320 PRINT@289," YOUR OWN IRREGULAR PRISMS BY ";
1330 PRINT@321," CHANGING THE DATA IN THE LAST";
1340 PRINT@353," LINE OF THIS PROGRAM. ";
1350 PRINT@385," THE FORMAT IS X,Y,X,Y,..ETC ";
1360 PRINT@449,"REGULAR (1) OR IRREGULAR (2) ?";
1370 KB$=INKEY$:IFKB$=""THEN1370
1380 IFKB$<"1" ORKB$>"2"THEN1370
1390 SHAPE=ASC(KB$)-48
1400 RETURN
1410 REM***SAVE START POINTS***
1420 REM*** IRREGULAR SIDES ***
1430 FOR T=1 TO NS
1440 READ C(1,T,1),C(1,T,2)
1450 C(1,T,3)=-1
1460 NEXTT
1470 RETURN
1480 REM***SAVE START POINTS***
1490 REM*** REGULAR SIDES ***
1500 FOR T=1 TO NS
1510 C(1,T,1)=COS(ATN(1)*8*T/NS)*1.425
1520 C(1,T,2)=SIN(ATN(1)*8*T/NS)*1.425
1530 C(1,T,3)=-1
1540 NEXTT
1550 RETURN
1560 REM**MAKE YOUR OWN PRISM**
1570 REM** DATA - X,Y,X,Y,....**
1580 REM** DELETE OR REM THE **
1590 REM** NEXT LINE, FOR AN **
1600 REM** ALTERNATIVE PRISM **
1610 DATA -1.25,1,1.25,1,.5,0,1.5,-1,-.75,-.5,.25,
-.25
1620 DATA -1,-1,-1,1,1,1,1,-1,0,-.5,-1.5,-1.5

```


9. N-Particle

General Description

The gravitational attraction between any two masses is given by the formula:

$$F = \frac{G * m1 * m2}{D * D}$$

In the program the forces on each mass due to all the other masses are added up so that its new position after time 't' may be estimated. To minimise cumulative errors, all masses are moved at the same time – i.e. all the forces on all the masses and all their positions (calculated using the old positions), are calculated before they are all moved, (Thus, at least:

Force of Mass A on Mass B = Force Mass B on Mass A
i.e. at least the momentum is conserved even for long sampling times.)

A further refinement in the program is that rather than using the instantaneous acceleration (which will always be too little when masses are approaching and too large when they are separating), the mean of the two latest and the previous acceleration is used (except, of course, for the first calculation).

Detailed Description

The main program consists of three nested loops.

The innermost loop takes the mass at a time (Ms, for example) and its data, and goes round the loop (N – 1) times to calculate the total forces exerted on Ms by the other (N – 1) masses. (This loop having, of course, the 'OLD' positions of all the masses.)

The next loop out feeds masses one at a time to the inner loop, and when each emerges with its X and Y acceleration calculates its 'NEW' velocity (P and Q) components and co-ordinates (stored in ALSO (S,O) and ALSO (S, 1)).

When all the masses have been processed, the outermost loop uses a mini-loop (1580–1610) to update all the information used by the other loops, and returns to the start.

Lines 10–570 Descriptions, instructions and set up arrays, etc.

580–1010 Accepts data and stores it in DAT (10, 5). (Notice the sampling time may be changed if greater accuracy is required.)

1020–1070 The outer loop.

1080 $S = 0$. This resets the count for the centre loop – i.e. starts it all again with the new data (updated in the lower half of the loop).

1090–1550 The centre loop. The top half of this loop takes one mass at a time, M_s and gives the variables M , X , Y , U and V its co-ordinates and velocity from DAT, then feeds those variables to the inner loop (it also resets the count with $H = 0$).

1230–1360 The inner loop. This inner loop finds the force due to the H mass on the mass M_s for all the masses – i.e. each time around, it calculates the force on M_s due to another mass and adds it to the previous total. B and C hold the running total of the X and Y acceleration.

1370–1450 ACC holds acceleration. The old values are transferred to E and F , before ACC is updated (with B and C). B and C are then recalculated using B and E or C and F (to get the average over the time interval ' t ').

K is 0 the first time round and thereafter 0.5.

1460–1530 P and Q are the new components of velocity calculated using:

$$[V = u + ft]$$

They are stored directly in DAT. The new positions are calculated using:

$$[S = ut + \frac{1}{2} ft^2]$$

and stored in the array ALSO until all the particles have been treated.

1540–1670 Mini-loop updates all masses by transferring ALSO to DAT.

Educational Notes

In the classroom this program will prove invaluable for the opportunities it offers teachers to provide students with a lucid demonstration of the laws of gravity. If restricted to two masses

with the initial momentum at zero, the masses can orbit in quite stable ellipses about their common centre of gravity. This, of course, is the acid test!

The program also has the useful capacity to demonstrate – or, more precisely, underline – the conservation of momentum even when it goes wrong – i.e. when a collision occurs.

Program Listing

```

10 *****
20 *           N-PARTICLE           *
25 *                                           *
30 * WRITTEN FOR THE DRAGON *
35 *                                           *
40 * BY           RAY NEW           1983 *
50 *****
60 TITLE$="#####N-PARTICLE#####"
70 P1$="T20GAG"
80 DIM ACC(10,1),DAT(10,4),ALSO(10,1)
90 S=0:U=0:V=0:H=0:I=0:J=0:D=0
100 B=0:N=0:P=0:Q=0:K=0:T=0:E=0:F=0
110 T=1
120 REM *****
130 REM *           INSTR/DESC           *
140 REM *****
150 CLS
160 PRINTTITLE$;
170 PRINT@65," THIS PROGRAM PLOTS THE PATHS";
180 PRINT@97,"OF ANY NUMBER OF PARTICLES";
190 PRINT@129,"MOVING UNDER THEIR MUTUAL";
200 PRINT@161,"GRAVITATIONAL ATTRACTION.";
210 PRINT@193," IT REQUIRES,FOR EACH PARTICLE,";
220 PRINT@225,"mass,start CO-ORDINATES,speed,";
230 PRINT@257,"AND angle.";
240 PRINT@289," THE SHORTER THE SAMPLING";
250 PRINT@321,"INTERVAL(t),THE MORE ACCURATE";
260 PRINT@353,"THE APPROXIMATIONS USED.(AND"
270 PRINT@385,"THE SLOWER THE PLOT).SIMILARLY";
280 PRINT@417,"THE MATHS BREAKS DOWN IF THEY";
290 PRINT@449,"APPROACH EACH OTHER TOO CLOSELY";
300 PRINT@483,"PRESS ANY KEY TO CONTINUE";
310 PLAY P1$
320 IF INKEY$=""THEN320
330 CLS
340 PRINTTITLE$;
350 PRINT@65," NEWTON'S THIRD LAW IS HOWEVER";
360 PRINT@97,"ALWAYS OBEYED SO momentum AT";
370 PRINT@129,"LEAST IS CONSERVED.";
380 PRINT@161," CLEARLY THE INITIAL TOTAL";

```



```

390 PRINT@193,"MOMENTUM MUST BE ZERO IF THE";
400 PRINT@225,"PARTICLES ARE TO STAY ON THE";
410 PRINT@257,"SCREEN.";
420 PRINT@289," THE UNITS USED ARE ARBITRARY.";
430 PRINT@483,"PRESS ANY KEY TO CONTINUE";
440 PLAY P1$
450 IF INKEY$="" THEN 450
460 CLS
470 PRINT TITLE$;
480 PRINT@99,"EXAMPLE FOR TWO PARTICLES"
490 PRINT@163,"PARTICLE 1          PARTICLE 2"
500 PRINT@225,"    MASS=100          MASS=1"
510 PRINT@257,"    XPOS=230          XPOS=230"
520 PRINT@289,"    YPOS=120          YPOS=80"
530 PRINT@321,"    SPEED=10          SPEED=30"
540 PRINT@353,"    ANGLE=270          ANGLE=90"
550 PRINT@483,"PRESS ANY KEY TO CONTINUE";
560 PLAY P1$
570 IF INKEY$="" THEN 570
580 REM *****
590 REM *      ACCEPT DATA      *
600 REM *****
610 CLS
620 PRINT TITLE$;
630 PRINT@68,"TO CHANGE 'T' PRESS 'T' ";
640 PRINT@100,"OTHERWISE PRESS ANY KEY"
650 PLAY P1$
660 IK$=INKEY$
670 IF IK$="" THEN 660
680 PLAY P1$
690 IF IK$="T" THEN PRINT@164,"'T' IS PRESENTLY 1.
":INPUT "    PLEASE ENTER NEW VALUE";T
700 PRINT
710 PLAY P1$
720 INPUT " ENTER NO. OF PARTICLES N=";R
730 T=T/5
740 PRINT@483,"PRESS ANY KEY TO CONTINUE";
750 PLAY P1$
760 IF INKEY$="" THEN 760
770 CLS
780 PRINT TITLE$;
790 PRINT@65,"ENTER DATA FOR";
800 PRINT " MASS NUMBER" S+1
810 PLAY P1$
820 INPUT " MASS=";M
830 PLAY P1$
840 INPUT " XPOS=";X
850 PLAY P1$
860 INPUT " YPOS=";Y
870 PLAY P1$
880 INPUT " SPEED=";V
890 PLAY P1$

```



```

900 INPUT " ANGLE(DEGS)=";A
910 A=(A-90)*3.14159/180
920 DAT(S,0)=M*1000
930 DAT(S,1)=X
940 DAT(S,2)=Y
950 DAT(S,3)=V*COS(A)
960 DAT(S,4)=V*SIN(A)
970 S=S+1
980 PRINT@4B3,"PRESS ANY KEY TO CONTINUE";
990 PLAY P1$
1000 IF INKEY$="" THEN 1000
1010 IF S<R THEN 770
1020 REM *****
1030 REM * OUTER LOOP - RESTART *
1040 REM *****
1050 PMODE 4,1
1060 PCLS
1070 SCREEN 1,1
1080 S=0
1090 REM *****
1100 REM * CENTRE LOOP *
1110 REM * ONE MASS AT A TIME *
1120 REM *****
1130 M=DAT(S,0)
1140 X=DAT(S,1)
1150 Y=DAT(S,2)
1160 U=DAT(S,3)
1170 V=DAT(S,4)
1180 H=0
1190 IF X>255 OR X<0 OR Y>191 OR Y<0 THEN 1210
1200 PSET(X,Y,5)
1210 B=0
1220 C=0
1230 REM *****
1240 REM * INNER LOOP *
1250 REM * ADDS FORCE DUE TO *
1260 REM * EACH OTHER MASS *
1270 REM *****
1280 IF H=S THEN 1350
1290 N=DAT(H,0)
1300 I=DAT(H,1)
1310 J=DAT(H,2)
1320 D=(SQR((I-X)*(I-X)+(J-Y)*(J-Y)))^3
1330 B=B+N*(I-X)/D
1340 C=C+N*(J-Y)/D
1350 H=H+1
1360 IF H<R THEN 1280
1370 REM *****
1380 REM * END OF INNER LOOP *
1390 REM *****
1400 E=ACC(S,0)
1410 F=ACC(S,1)

```

```

1420 ACC(S,0)=B
1430 ACC(S,1)=C
1440 B=B+K*(B-E)
1450 C=C+K*(C-F)
1460 P=U+B*T
1470 Q=V+C*T
1480 DAT(S,3)=P
1490 DAT(S,4)=Q
1500 ALSO(S,0)=X+U*T+.5*B*T*T
1510 ALSO(S,1)=Y+V*T+.5*C*T*T
1520 S=S+1
1530 IF S<R THEN 1130
1540 REM *****
1550 REM * END OF CENTRE LOOP *
1560 REM *****
1570 S=0
1580 DAT(S,1)=ALSO(S,0)
1590 DAT(S,2)=ALSO(S,1)
1600 S=S+1
1610 IF S<R THEN 1580
1620 K=.5
1630 GOTO 1080
1640 REM *****
1650 REM * END OF OUTER LOOP *
1660 REM *****
1670 END

```

10. Anagram

General Description

This program generates anagrams from given data, randomly accessing the data and jumbling the letters. It invites a response from the child and checks for certain letter combinations. Those it considers most unlikely it marks with an arrow (↑). There are limited sound effects to hold the attention of the player.

It finally invites the player to judge whether the word made is an 'English' word. Deliberately, there are no facilities to rub-out errors and all attempts made by the player are saved for display to the teacher when the program finishes. The teacher is needed to correct and instruct the player when the 'game' has finished and the final display is marked with '?'s against words the computer thinks suspect and with '*'s against those that the player thinks suspect.

In theory, progress on from the display page is controlled by the teacher who will need to type in the '@' character.

You are first presented with an introduction page. This is followed by letters being 'beeped' across the screen for the anagram. After each attempt a brief opportunity is offered to give up on the anagram and progress to the menu page, whence the game can be finished, replayed or results displayed. Enter a '?' to view the words made so far.

It is important to read the educational notes before typing in the program as several decisions must be taken concerning the data. The game seems to hold the attention of players for about 15 minutes.

Detailed Description

Lines 5–90 Title display.

100–350 Main structure of the program. The final block of program is jumped to at line 310, otherwise the program is

really a simple series of steps through procedures. Certain variables have to be initialised.

360–500 All jumbling of the letters is done on string handling routines. WORD\$ is the unmixed word and NWORD\$ is the anagram.

510–700 This routine marches the letters of the anagram across the screen to a respectable beeping sound. The small delay is to give each letter time to sound its 'beep'. Play with the sound if you like.

710–810 Routine to randomly select a word.

820–1100 This routine itself calls a number of other sub-routines. The last letter pair used is created and checked against letter pairs in the Test Pairs Array. The players' word is stored for later display with a flag set if the computer does not like the letter combination.

1110–1160 This routine, called from the one above, determines if the last letter entered (TEST\$) was a vowel. A flag (HFLAG) is set if a vowel is found.

1170–1240 This routine checks to see whether the letter chosen is in the anagram. If it is not, the selected letter will generate no screen response. There is also a check that the same letter in the anagrams is not used twice. This is done by substituting the character '9' into the word once the letter in its position has been used.

1250–1310 This is the heart of the checking section of the program. The educational section of the notes will give you a guide on the data to be included. The routine checks letter pairs against acceptable pairs of consonants, or pairs of vowels, held in the Test Pairs Array.

1320–1390 This allows the player to decide if the word is an 'English' word or not and adds a suitable marker '*' at the end of the word if the player rejects the word made.

1400–1570 Introduction.

1580–1760 This displays the end of the game menu.

1770–1890 This routine displays the results. Return to the menu should be via the @ key.

1900–2000 Displays the words entered so far, on the anagram building screen.

2010–2300 Test Pairs Array.

It is very important that these are read carefully before typing in the program and the DATA in particular. The DATA and Test Pairs supplied is test data to show that the program works, but will not provide the best use of the program. The teacher using the program must decide (i) the number of words available to be mixed up, (ii) the words themselves and (iii) the letter pairs that will be allowed.

The words themselves must be chosen with care. You may wish to concentrate on the letter pairs 'ST', 'CT', 'LT'. It would then be advisable to choose short words for the data which include these letters, and other letters when combined with S, T etc. would not be acceptable. Take e.g. 'BOAST'. By limiting the acceptable letter pairs you will test for 'BT', 'BS', 'SB' etc. The longer words provide greater problems for computer analysis, as the chances of three consonants being put together and being accepted by the computer increase. E.g.: 'SST' will currently pass the pairs test as 'SS' and 'ST' both separately pass the test. It is up to the user to decide whether to write a three letter check procedure, but I feel that back checking has to finish arbitrarily anyway, so I have stopped at letter pairs. If the teacher wishes to avoid this problem, the anagram words must again be chosen with care.

It is partly for this reason, and the educational advantages of making the player decide whether the word he has used is an 'English' word for himself, that such a routine has been included.

I suggest that you keep several copies of the program with different sets of data or spool the data in from tape if you wish to adapt the program in that manner. In this way you can develop a structured series of programs which in themselves will form a learning program.

I will of course be pleased to hear in detail of any program of learning that is developed. If you don't follow this idea, the program will stand alone – but will not be as useful as it could be.

Program Listing

```
5 REM *****TITLE DISPLAY*****
10 WORD$=" ANAGRAM TESTER "
11 CLS
12 FOR P=1TO15
13 RW$=RIGHT$(WORD$,P)
14 LW$=LEFT$(WORD$,P)
15 RW$=STRING$(16-P," ")+RW$
16 LW$=LW$+STRING$(16-P," ")
17 PRINT RW$;LW$;
18 NEXT P
19 M1$="03L4.FL8EL4.DL8EL4FGA"
20 M2$="FB-AGFEDEC"
21 M3$="04DP16LBCL4.03AGL8AL1F"
22 PLAY "T4"+M1$+M2$+M1$+M3$
30 REM*****
40 REM* THE ANAGRAM PROGRAM *
50 REM* *
60 REM* WRITTEN FOR THE DRAGON *
70 REM* *
80 REM* BY LAURENCE OWEN 1983 *
90 REM*****
100 REM
110 REM *****MAIN STRUCTURE*****
120 REM
130 CLEAR 5000:DIM PR$(26),KWORD$(42)
140 GOSUB2040 'LOAD PAIRS ARRAY
150 GOSUB1430 ' START
160 CLS:SIGN=0
170 FOR I=1TO42:KWORD$(I)="":NEXT
180 KW=1
190 GOSUB740 ' GETWORD
200 LL=LEN(WORD$)
210 GOSUB390 ' JUMBLE(WORD$)
220 GOSUB540 ' DISPLAY
230 GOSUB850 ' GETIN
240 IF SIGN THEN KWORD$(KW)=KWORD$(KW)+" ?"
250 GOSUB1350 ' CHOOSE
260 SIGN=0
270 KW=KW+1
280 IF KW=43 GOTO 1610
290 PRINT@448,"IF YOU'VE FINISHED,PRESS 'Y' NOW";
300 FOR D=1TO200:FI$=INKEY$
310 IF FI$="Y" THEN 1610
320 NEXT
330 PRINT@160:PRINT@224:PRINT@256:PRINT@320:PRINT@
448
340 GOTO230
350 END
```

```

360 REM *****
370 REM ** JUMBLE ROUTINE **
380 REM *****
390 TEMP$="":OWORD$=WORD$
400 FOR N=1 TO 10
410 X=LEN(WORD$)
420 Y=RND(X)
430 TEMP$=TEMP$+MID$(WORD$,Y,1)
440 L$=LEFT$(WORD$,Y-1)
450 R$=RIGHT$(WORD$,X-Y)
460 WORD$=L$+R$
470 IF WORD$="" THEN 490
480 NEXT
490 NWORD$=TEMP$
500 RETURN
510 REM *****
520 REM ** DISPLAY ROUTINE **
530 REM *****
540 CLS
550 Y=LEN(NWORD$)
560 W=2:X=0
570 FOR N=1TO10
580 X=X+1
590 L$=MID$(NWORD$,X,1)
600 FOR Z=1 TO W
610 PRINT@38+Z,L$
620 SOUND50+8*Z,1
630 PRINT@38+Z," "
640 NEXT Z
650 W=W+2
660 PRINT@70+Z,L$
670 SOUND50,4
680 IF X=Y THEN 700
690 NEXT N
700 RETURN
710 REM *****
720 REM ** GETWORD ROUTINE **
730 REM *****
740 RESTORE
750 X=RND(10)
760 FOR Y=1 TO X
770 READ WORD$
780 NEXT Y
790 DATA MASSIVE,GANTRY,TRIBUTE,TRYING,SAUCER
800 DATA VISION,MIXTURE,METAL,CATTLE,THROUGH
810 RETURN
820 REM *****
830 REM ** GETIN ROUTINE **
840 REM *****
850 X=1:CONWORD$=NWORD$
860 PRINT@161,"MAKE YOUR WORD"
870 FOR M=1 TO 10

```



```

880 IWORD$=CONWORD$
890 ICHAR$=INKEY$: IF ICHAR$="" THEN 890
900 IF ICHAR$="?" GOSUB 1930
910 IF ASC(ICHAR$)=13 THEN RETURN
920 GOSUB 1200 ' INCHECK
930 IF GFLAG=0 THEN 890 ELSE PRINT@225+X, ICHAR$;
940 SOUND 50,2
950 KWORD$(KW)=KWORD$(KW)+ICHAR$
960 PAIR$=RIGHT$(PAIR$,1)+ICHAR$
970 IFX=1 THEN X=X+1: STOR$=ICHAR$: GOTO 880
980 TEST$=ICHAR$
990 GOSUB 1140 ' VOWEL
1000 IF HFLAG THEN F1=1 ELSE F1=0
1010 TEST$=LEFT$(PAIR$,1)
1020 GOSUB 1140 ' VOWEL
1030 IF HFLAG THEN F2=1 ELSE F2=0
1040 FFLAG=1
1050 IF (F1 AND F2) OR (F1=0 AND F2=0) THEN GOSUB 1
280 ' CHECK
1060 IF FFLAG=0 THEN PRINT@257+X, CHR$(94): SIGN=1: S
OUND 1,1
1070 STOR$=ICHAR$: X=X+1
1080 IF LL=X-1 THEN 1100
1090 NEXT M
1100 RETURN
1110 REM *****
1120 REM ** VOWEL ROUTINE **
1130 REM *****
1140 P=INSTR(1,"AEIOU",TEST$)
1150 IF P=0 THEN HFLAG=0 ELSE HFLAG=1
1160 RETURN
1170 REM *****
1180 REM ** INCHECK ROUTINE **
1190 REM *****
1200 GFLAG=1
1210 P=INSTR(1,IWORD$, ICHAR$)
1220 IF P=0 THEN GFLAG=0 ELSE IWORD$=LEFT$(IWORD$,P-
1)+"9"+RIGHT$(IWORD$,LL-P)
1230 CONWORD$=IWORD$
1240 RETURN
1250 REM *****
1260 REM ** CHECK ROUTINE **
1270 REM *****
1280 STOR=ASC(STOR$)-64
1290 P=INSTR(1,PR$(STOR),PAIR$)
1300 IF P=0 THEN FFLAG=0 ELSE FFLAG=1
1310 RETURN
1320 REM *****
1330 REM ** CHOOSE ROUTINE **
1340 REM *****
1350 PRINT@321,"AN ENGLISH WORD?";
1360 WE$=INKEY$: IF WE$="" THEN 1360

```



```

1370 IF WE$="N" THEN KWORD$(KW)=KWORD$(KW)+"*":PRI
NT" NO":SOUND1,6:ELSE PRINT" YES":PLAY"T50ADA"
1380 FOR D=1TO200:NEXT
1390 RETURN
1400 REM *****
1410 REM ** START ROUTINE **
1420 REM *****
1430 CLS
1440 PRINT@6,"THE ANAGRAM TESTER"
1450 PRINT@37,"*****";
1460 PRINT@66,"THIS PROGRAM IS MEANT TO TEST"
1470 PRINT@97,"THE ABILITY TO CREATE LETTER"
1480 PRINT@129,"PAIRS FROM A VALID ANAGRAM."
1490 PRINT@194,"YOU WILL SEE LOTS OF LETTERS"
1500 PRINT@225,"IN A MESS. TRY TO GET THEM "
1510 PRINT@257,"INTO A WORD."
1520 PRINT@322,"THE COMPUTER WILL SHOW A"
1530 PRINT@353,"LITTLE ARROW ^^^ IF IT THINKS"
1540 PRINT@385,"A LETTER IS VERY SILLY."
1550 PRINT@484,"PRESS ANY KEY TO START";
1560 R$=INKEY$:IFR$=""THEN1560
1570 RETURN
1580 REM
1590 REM *****END OF GAME*****
1600 REM
1610 CLS6:PRINT@6," call your teacher ";
1620 PRINT@108,"OPTIONS";
1630 PRINT@161,"LIST WORDS AND RESULTS (1)";
1640 PRINT@225,"JUMBLE ANOTHER WORD (2)";
1650 PRINT@289,"START THE PROGRAM AGAIN (3)";
1660 PRINT@353,"END THE PROGRAM (4)";
1670 PRINT@452,"ENTER 1,2,3 OR 4 ONLY >>":SCREEN0
,1
1680 R$=INKEY$:IFR$=""THEN1680
1690 IFR$<"1" OR R$>"4" THEN1680
1700 PRINTR$;:FORD=1TO100:NEXT
1710 IFR$="1" THEN GOSUB1800:GOTO1610
1720 IFR$="3" THEN 150
1730 IFR$="2" THEN 160
1740 SCREEN0,1
1750 TU$="03CDEFGAB04CC03BAGFEDC":PLAY "T16"+TU$+T
U$
1760 CLS: PRINT@234,"!! BYE BYE !!":END
1770 REM *****
1780 REM ** RESULT ROUTINE **
1790 REM *****
1800 CLS:PRINT@2,"# ";NWORD$;" #";:PRINT@15,">> #
";OWORD$;" #";
1810 FOR X=1 TO 42 STEP3
1820 PRINT@32*(X+2)/3, KWORD$(X);TAB(11) KWORD$(X+
1);TAB(22) KWORD$(X+2);
1830 IF KWORD$(X+2)="" THEN X=42

```

```

1840 NEXT
1850 PRINT@485,"let your teacher look";
1860 SCREEN0,1
1870 CN$=INKEY$:IFCN$=""THEN1870
1880 IF ASC(CN$)<>64 THEN 1870
1890 RETURN
1900 REM *****
1910 REM ** VIEW ROUTINE **
1920 REM *****
1930 FOR VN=1 TO KW+1
1940 FOR VM=0 TO 9
1950 PRINT@181+32*VM,KWORD$(VN+VM)
1960 FORD=1TO100:NEXT
1970 NEXT VM
1980 FORD=1TO200:NEXT
1990 NEXT VN
2000 RETURN
2010 REM *****
2020 REM ** TEST PAIRS ARRAY **
2030 REM *****
2040 PR$(1) ="AI,AU"
2050 PR$(2) ="BB,BL,BR,BS,BY"
2060 PR$(3) ="CH,CK,CL,CR,CT"
2070 PR$(4) ="DG,DM,DR,DS,SY"
2080 PR$(5) ="EA,EE,EI,EU"
2090 PR$(6) ="FF,FL,FR,FS,FT,FY"
2100 PR$(7) ="GG,GH,GL,GN,GR,GS,GY"
2110 PR$(8) ="HR,HS,HT"
2120 PR$(9) ="IA,IE,IO"
2130 PR$(10)=""
2140 PR$(11) ="KK,KN,KS,KY"
2150 PR$(12) ="LC,LD,LF,LG,LK,LL,LM,LP,LR,LS,LT,LY"
2160 PR$(13) ="MB,MC,MM,MP,MS,MY"
2170 PR$(14) ="NC,ND,NG,NK,NN,NP,NS,NT,NY"
2180 PR$(15) ="OA,OI,OO,OU"
2190 PR$(16) ="PH,PL,PN,PP,PR,PS,PT,PY"
2200 PR$(17)=""
2210 PR$(18) ="RB,RC,RD,RF,RG,RH,RK,RL,RM,RN,RR,RS,
RT,RY"
2220 PR$(19) ="SC,SH,SK,SL,SM,SN,SP,SQ,SS,ST,SW,SY"
2230 PR$(20) ="TH,TL,TR,TS,TT,TW,TY"
2240 PR$(21) ="UA,UE,UI,UO"
2250 PR$(22) ="VS,VV,VY"
2260 PR$(23) ="WH,WS"
2270 PR$(24) ="XT,XY"
2280 PR$(25)=""
2290 PR$(26) ="ZY,ZZ"
2300 RETURN

```

11. Fraction Towers

General Description

This program is intended mainly for use with younger children. The instructions will be displayed on the screen at the start of the program. Press any key to start the game. The screen will contain a series of fraction towers, all of which will be blank except the top one. The object is to place increasingly large fractions down the fraction tower.

Detailed Description

Lines 80–130 Display title page.

170–250 Display instructions.

290–300 Update count of fractions.

310–360 Read in next 'starting fraction' and check for terminator.

370–420 Build tower outline.

430–520 Start game.

560–580 Display fraction in box.

590–600 Display query in next box.

610–700 Input next fraction.

710–750 Validate fraction; check for numerator and denominator.

760–790 Set N = numerator, D = denominator.

800–830 Check N and D are numeric.

850 Check N and D are not decimals.

860–870 Check N/D less or equal to last fraction.

880 Check N/D more than one.

890–940 Valid fraction, increase height position, store fraction.

950–960 Test for final fraction.

970–990 Enter final fraction.

1000–1110 Congratulate successful player.

- 1120-1130 Test for further towers.
- 1140-1200 Invite player to play again.
- 1210-1260 Error message.
- 1270-1300 Show reason for invalid input.
- 1310-1360 Congratulations on a successful game.
- 1370-1420 Thank player and end program.

Educational Note

This program should really only be used once you have taught fraction towers in the classroom. It provides an entertaining medium for experimenting with fraction towers. I suspect the program is better used with primary children rather than secondary children, though I have used it with first years in secondary school.

I found that a group of four or five round a keyboard and monitor happily discussed amongst themselves what the next fraction tower ought to be.

Program Listing

```

10 REM *****
20 REM *   FRACTION   TOWERS   *
30 REM *               *
40 REM * WRITTEN FOR THE DRAGON*
50 REM *               *
60 REM *BY ALUN HUTCHINSON 1983*
70 REM *****
80 CLS
90 REM *** TITLE PAGE
100 PRINT@70,"*****"
110 PRINT@102,"* FRACTION TOWERS *"
120 PRINT@134,"*****"
130 PRINT@452,"PRESS ANY KEY TO START"
140 A$=INKEY$
150 IFA$=""THEN140
160 PLAY "T2004C"
170 REM *** INSTRUCTIONS
180 CLS
190 PRINT@100,"YOU ARE GOING TO BUILD"
200 PRINT@132,"FRACTION TOWERS."
210 PRINT@196,"EACH FRACTION THAT YOU"
220 PRINT@228,"CHOOSE MUST BE BIGGER"
230 PRINT@260,"THAN THE ONE BEFORE AND"
240 PRINT@292,"MUST BE LESS THAN 1."

```



```

250 PRINT@451,"PRESS ANY KEY TO CONTINUE"
260 A$=INKEY$
270 IFA$=""THEN260
280 CLS
290 REM *** 'R' = FRACTIONS READ
300 R=R+1
310 REM *** READ NEXT FRACTION INTO F$
320 REM *** AND EQUIVALENT DECIMAL INTO ANS
330 READ F$,ANS
340 DATA 1/10,0.1,1/4,0.25,4/5,0.8,END,0
350 REM *** TEST FOR END OF DATA
360 IF F$="END" THEN 1370
370 REM *** BUILD TOWER OUTLINE
380 FOR A=0TO4
390 PRINT@96*A+5, "*****"
400 PRINT@96*A+37, "*"
410 PRINT@96*A+69, "*****"
420 NEXTA
430 PRINT@310,"PRESS"
440 PRINT@340,"SPACEBAR"
450 PRINT@372,"TO START"
460 PLAY"T20ABCDEFG"
470 A$=INKEY$
480 IFA$=""THEN470
490 PRINT@310, " "
500 PRINT@340, " "
510 PRINT@372, " "
520 PLAY"T2005C"
530 REM *** 'H' = HEIGHT
540 REM *** 'P' = POSITION OF '/' IN FRACTION
550 H=0:P=2
560 REM *** DISPLAY FRACTION IN APPROPRIATE BOX
570 PRINT@96*H+43-P,F$
580 PRINT@96*H+47,"*"
590 REM *** DISPLAY '?' IN NEXT BOX
600 PRINT@(H+1)*96+42,"? *"
610 PRINT@150," NEXT"
620 PRINT@182,"FRACTION"
630 PRINT@214," (X/Y)"
640 REM *** INVITE INPUT OF NEXT FRACTION
650 PRINT@276," ";
660 INPUTNF$
670 REM *** CLEAR INSTRUCTIONS
680 PRINT@369," "
690 PRINT@408," "
700 PRINT@439," "
710 REM *** VALIDATE FRACTION ENTERED
720 P=INSTR(1,NF$,"/")
730 IF P=0 THEN1220
740 IF P=LEN(NF$) THEN 1220
750 IF P=1 THEN 1220
760 REM *** 'N' = NUMERATOR

```

```

770 N=VAL(LEFT$(NF$,P-1))
780 REM *** 'D' = DENOMINATOR
790 D=VAL(RIGHT$(NF$,LEN(NF$)-P))
800 IF D<=0 THEN1220
810 IF D>999 THEN1220
820 IF N<=0 THEN1220
830 IF N>999 THEN1220
840 REM *** MORE VALIDATION OF INPUT
850 IF N<>INT(N) OR D<>INT(D) THEN ER$=" NO DECIM
ALS":GOTO1280
860 IF N/D<ANS THEN ER$=" TOO SMALL":GOTO1280
870 IF N/D=ANS THEN ER$=" THE SAME":GOTO1280
880 IF N/D>=1 THEN ER$="BELOW 1 PLEASE":GOTO1280
890 REM *** FRACTION ENTERED IS OK
900 PLAY"O4T20CCCCBBB"
910 H=H+1
920 F$=NF$
930 ANS=N/D
940 PRINT@276," "
950 REM *** TEST FOR COMPLETE TOWER
960 IF H<>4 THEN 570
970 REM *** ENTER FINAL FRACTION IN LAST BOX
980 PRINT@427-P,F$
990 PRINT@431,"*"
1000 REM *** TOWER COMPLETE
1010 PRINT@84,"WELL DONE"
1020 PRINT@145," "
1030 PRINT@177," "
1040 PRINT@209," "
1050 PLAY"T2004CDEFGAB"
1060 FORI=1TO100:NEXT
1070 PRINT@180,"YOU HAVE"
1080 PRINT@212,"COMPLETED"
1090 PRINT@244,"THE TOWER"
1100 PLAY"T2004BAGFEDC"
1110 FORI=1TO200:NEXT
1120 REM *** TEST FOR FURTHER TOWERS
1130 IF R>2THEN1310
1140 PRINT@339,"DO YOU WANT"
1150 PRINT@372,"ANOTHER ?"
1160 PRINT@404,"(YES/NO)"
1170 PRINT@435," ";
1180 INPUTYN$
1190 IFYN$="YES"OR YN$="Y" THEN 280
1200 GOTO1370
1210 REM *** ERROR MESSAGE
1220 PRINT@375,"ERROR"
1230 PRINT@408,"ON"
1240 PRINT@439,"INPUT"
1250 PLAY "T2001CCCD"
1260 GOTO650
1270 REM *** SHOW REASON FOR REJECTION

```

```

1280 PRINT@369,ER$
1290 PLAY "T2001EEEF"
1300 GOTO650
1310 FORI=1TO3000:NEXTI
1320 REM *** END OF GAME
1330 CLS
1340 PRINT@72,"CONGRATULATIONS"
1350 PRINT@129,"YOU COMPLETED ALL THREE TOWERS"
1360 GOTO1380
1370 CLS
1380 PRINT@260,"THANK YOU FOR PLAYING"
1390 PLAY"T2003CDEFGAB"
1400 FORI=1TO3000:NEXTI
1410 CLS
1420 END

```

12. Stock Market

General Description

This long program combines game playing with a simulation. The game is basically a Stock Market game in which you trade with a restricted number of companies and try to make the goal of £100,000 from the £10,000 that you start with. The simulation begins when you are able to set the parameters of the game and play against your own parameters or those chosen for you by the teacher.

When the game begins, you are given the introductory messages, and then you may select the parameters option. Type any key to avoid this, and the simulation moves straight into the stock market panel of information. To control the game there is single key entry as specified except that share purchases must be entered with the 'Enter' key. The parameters, if chosen, are also single key entry and must be entered as required including '+' or '-' signs as requested, decimal points if asked for and double digit numbers if required. This is part of the data validation.

You will then see the companies displayed. The first column is the actual market valuation in pence, and the second column is the last price change in pence. Then follow two more columns. The first is your actual share holding and the second is the value of the shares you hold.

Under the line at the bottom is displayed the cash you currently have at your disposal, the week of the simulation, and the total of the valuations of your share holdings.

Information messages are displayed in the remaining area of the screen. Once you have chosen a share in which to trade, you may not return to that share in the current week. You may process the shares in any order. The final messages are a bit trite and you may like to improve on them.

Detailed Description

Lines 10–410 These give the main structure of the program.

420–440 This overworked subroutine clears the message area at the bottom of the screen.

450–630 This simply takes in details from the keyboard of the proposed share purchase and validates the input. Single key entry is validated, up to six digits, or the Enter key terminates entry. There is a share limit of 50,000, so that six digits allows the user to deliberately exceed the limit and re-try if a wrong key is pressed.

640–700 This simulation is generous. Overdrafts are granted from time to time, but only during a boom and when the cash you request is less than £5,000 (line 660).

710–740 There is a share limit of 50,000 to inhibit crazy single company speculation. This is not offered as a start parameter but the value in line 730 is all that need be changed.

750–780 This routine inhibits large sales after week 8. The disaster will only occur once – once is enough, but should you wish to transfer more than £15,000 then you will lose it to the unscrupulous broker.

790–830 This routine combines bank charges and brokerage fees. Both may be set as start of simulation parameters.

840–910 This routine is called any time a transaction involves a change of screen status and has to be called before the screen is re-displayed. The subroutine 'Total' is performed within this to update the total value of your holdings.

920–1040 This routine displays the question on the screen and if you type '0' will set a flag which will send you on to the next week, otherwise selecting the company in which you wish to trade. The reply is restricted to the number of firms.

The conditions are carefully thought out to prevent too quick a win. If you want the disasters to reoccur then leave out the flag from the condition. I think it is fundamentally unfair to hit a person when he's down, so they get hit just once. You will find that there are times when for several games no disasters occur and at other times they seem to pile up.

1140–1250 The unscrupulous broker routine.

1260–1500 This routine is meant to teach the folly of heading without worry into debt. If the overdraft is withdrawn then the procedure searches your share holdings one by one (lines 1380–1400) for your largest holding, sells that at a suitable charge and

sees if you are still in debt.

1510–2290 These are the details of the disasters and could be changed at will or left out if you did not like them. The only point to register is that they are called from 'DISASTERS' and the bad disasters recursively call the disasters themselves in case your bad luck can be compounded in some way.

2300–2400 This routine prints the headings. The screen only just holds all the information shown. PRINT USING in line 2360 ensures that only five characters of the company names are printed.

2410–2430 prints the share holdings. As in the previous routine, here is another example of the power of the PRINT USING command.

2440–2530 This routine works out the value of your holding in the last column and calls the share and totals display sub-routines.

2540–2560 Totals display routine.

2570–2700 The key line here is 2630. This determines the outcome of the game. The TREND of each economic cycle, the length of which you may set as a parameter at the beginning, combined with the programmed success of each company called 'LUCK' and the economic bias you give the program called 'BIAS', determines whether a share will move up or down. The RANGE in line 2640 is another parameter you have control of. This decides the maximum possible increase or decrease in the value of a share. A smaller range provides a difficult simulation. Its default is 80, the maximum.

2710–2760 This routine decides whether the TREND for an economic cycle is 'up' or 'down' – 'bull' or 'bear' – and also sets the range of the TREND.

2770–2820 Totals the value of holdings.

2830–2860 This routine is included to prevent you renegeing on a purchase or sale decision. If you wish to allow a 'second bite at the cherry', then lines 160 and 1860 can be removed.

2870–3060 If you choose not to set your own parameters, the default values are set up here.

3070–3270 Fancy title. If you're typing the program in yourself, you may want to leave out these lines!

3280–3560 Introduction to the stock market.

3570–4000 Apart from typing BREAK, once you are engaged in parameter choosing, you must see it through to the end. The parameters are displayed one by one, and keyboard entry is

required exactly as displayed. This removes the need for subsequent validation and guarantees the program cannot crash from choosing strange parameters. They won't get past the keyboard. Thus it is a little tedious, and in general I suggest only an adult does this or produces suitable parameters on a sheet for a student.

4010–4070 Subroutine to get and validate keyboard input.

4080–4260 End of game messages.

Educational Notes

There is no doubt that even played as a straight game, the fourth and fifth years have enjoyed this immensely. Profit and loss, disasters and trading all become quickly apparent. The more interesting aspects lie in planning parameters and playing against them, plotting on graph paper or in note form what you are doing. For example: if you select just two firms and an economic cycle of 6, plot the share values on the graph paper as you sell or buy and see if a cycle becomes evident and how one might predict the onset of another cycle. Try setting a series of different share ranges combined with different share minima, and draw a conclusion as to the combination of events most likely to be profitable. Keep a table of your chosen minima and ranges and final victory totals. Of course you will run into 'disasters' and you can explain the need to produce more than one run of the simulation for better statistical reliability.

I offer a brokerage of 00%, but I felt there always had to be interest charges on bank overdrafts and thus the minimum rate there of 10%. I feel the simulation needs to run for 26 weeks, but you may change this in the initialisation.

Those youngsters who have tried to play against their own freely chosen parameters have been surprised to find out that usually they worsen their overall performance . . . and thus direction here is essential to avoid irritation. Good luck, this game/simulation is very compelling!

Program Listing

```
10 REM*****
20 REM*      STOCK MARKET      *
30 REM*      *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*      *
60 REM* BY LAURENCE OWEN 1983 *
70 REM*****
80 CLS
90 GOSUB2880      ' INITIALISE
100              ' & INTRODUCE
110 GOSUB2310     ' HEADINGS
120 FOR WKS=1 TO 26
130 GOSUB2580     ' SHARE-CALC
140 GOSUB2780     ' TOTAL
150 GOSUB2450     ' TABLE
160 GOSUB2840     ' SETTLE
170 FM=FIRMS
180 GOSUB1060     ' DISASTERS
190 DECIDED=0
200 SHARES=0
210 HOPE=0
220 GOSUB930      ' QUESTION
230 IF DECIDED THEN GOSUB 800:GOTO350
240 SALE=1
250 GOSUB460      ' SALE DETAILS
260 IF SHARES>0 THEN GOSUB650      ' CASHCHECK
270 IF SALE<>1 THEN 240
280 MAX=0
290 GOSUB720      ' SHARE LIMIT
300 GOSUB760      ' LARGE SALE
310 GOSUB800      ' COMMISSION
320 GOSUB850      ' UPDATE
330 C=HOPE
340 GOSUB2420     ' SHARE DISPLAY
350 GOSUB2550     ' TOTALS DSPLY
360 FM=FM-1
370 IF FM=0 OR DECIDED=1 THEN390
380 GOTO190
390 IF CASH>=100000 THEN 410
400 NEXT
410 GOTO 4130     ' END OF GAME
420 REM*****CLEAR MSG-BOARD***
430 PRINT@448,STRING$(63," ");
440 RETURN
450 REM*****SALE DETAILS SUB***
460 TEST=1
470 GOSUB430      ' CLEAR MSG
480 PRINT@448,"";:PRINT USING"%      %";COY$(HOPE);:P
RINT" SHARES. ( + BUY - SELL) ";
490 NUM$="":T1$="+":T2$="-":GOSUB4020
```



```

500 NUM$=NUM$+KB$
510 XS=0:T1$="0":T2$="9":TS=1
520 PRINT@494,"HOW MANY ? ";
530 GOSUB4020
540 IF ASC(KB$)=13 THEN TS=0:GOTO560
550 NUM$=NUM$+KB$:XS=XS+1:IFXS<6 THEN530
560 SHARES=VAL(NUM$):TS=0
570 IF SHARES=0 THEN MSG$="TRY AGAIN!":GOTO610
580 IF SHARES<0 THEN SOUND20,2 ELSE SOUND128,2
590 IF SHARES<0 AND ABS(SHARES)>HELD(HOPE) THEN MS
6$="WE ARE NOT FOOLS!":GOTO610 ELSE 620
600 IF SHARES<>INT(SHARES) THEN MSG$="WHOLE SHARES
ONLY PLEASE"ELSE620
610 TEST=0:GOSUB430:PRINT@484,MSG$;:FORD=1TO4000:N
EXT
620 IF TEST=0 GOTO460
630 RETURN
640 REM*****CASH CHECK SUB***
650 IF SHARES*MRKT(HOPE)/(100*(1-BC)) <= CASH THEN
RETURN
660 IF CASH-SHARES*MRKT(HOPE)/100 < -5000 OR TREND
< 0 THEN SALE=0
670 GOSUB430 ? CLEAR MSG
680 IF SALE THEN PRINT@452,"OVERDRAFT GRANTED !":S
OUND 200,2 ELSE PRINT@452,"NO OVERDRAFT TODAY":SOU
ND30,2
690 FORD=1TO2000:NEXT
700 RETURN
710 REM*****SHARE LIMIT SUB***
720 GOSUB430 ? CLEAR MSG
730 IF SHARES+HELD(HOPE) > 50000 THEN PRINT@452,"M
AXIMUM SHAREHOLDING=50000":MAX=1:SOUND80,2:FORD=1T
O5000:NEXT
740 RETURN
750 REM*****LARGE SALE SUB***
760 MONEY=ABS(SHARES*MRKT(HOPE)/100)
770 IF D1<1 AND MONEY+RND(5000) > 25000 AND WKS>8
AND MRKT(HOPE)>2*LOW(HOPE) THEN GOSUB1150 ? BROKER
780 RETURN
790 REM*****COMMISSION SUB***
800 CMMN=ABS(SHARES*MRKT(HOPE)/100)*BC
810 CASH=CASH-CMMN
820 IF CASH<0 THEN CASH=CASH+INT(CASH*LRATE)
830 RETURN
840 REM*****UPDATE SUB***
850 D1=D1-1
860 IF MAX AND D1<>1000 THEN SHARES=50000-HELD(HOP
E):HELD(HOPE)=50000
870 CASH=INT(CASH-SHARES*MRKT(HOPE)/100)
880 IF MAX=0 THEN HELD(HOPE)=HELD(HOPE)+SHARES
890 PRICE(HOPE)=HELD(HOPE)*MRKT(HOPE)/100
900 GOSUB 2780 ? TOTAL

```

```

910 RETURN
920 REM*****QUESTION SUB***
930 GOSUB430 ' CLEAR MSG
940 PRINT@449,"TYPE COMPANY NUMBER":PRINT" OR '0'
TO CONTINUE ";
950 KB$=INKEY$:IFKB$=""THEN950
960 IFKB$<"0" ORKB$>RIGHT$(STR$(FIRMS),1)THEN950
970 SOUND60,4
980 HOPE=VAL(KB$)
990 IF FLAG(HOPE)<>0 THEN930
1000 PRINTKB$;:SOUND120,4
1010 IF KB$="0" THEN DECIDED=1
1020 FORD=1T0500:NEXT
1030 FLAG(HOPE)=1
1040 RETURN
1050 REM*****DISASTERS SUB****
1060 IF CASH<-4000-RND(3000) THEN GOSUB1270 '
FORECLOSE
1070 IF CASH<0 AND TREND<0 THEN GOSUB1270 ' F
ORECLOSE
1080 IF D2=0 AND TREND>0.07 AND WKS<6+RND(6) AND C
ASH<8000+RND(4000) THEN GOSUB1520 ' LAWYER
1090 IF D3=0 AND TREND>0.03 AND WKS>9 AND CASH+TTL
>30000+RND(5000) THEN GOSUB1770 ' HOSPITAL
1100 IF D4=0 AND WKS>13 AND CASH+TTL>50000 AND TRE
ND>-0.04 THEN GOSUB1930 ' PROBATE
1110 IF D5=0 AND WKS>6 AND WKS<15 AND CASH<5000 AN
D TREND<-0.03 THEN GOSUB2060 ' POOLS
1120 IF D6=1 THEN GOSUB2220 ' BANKRUPT
1130 RETURN
1140 REM*****BAD BROKER SUB****
1150 FOR XB=1 TO 6
1160 GOSUB430 ' CLEAR MSG
1170 FORD=1T050:NEXT:SOUND20,2
1180 PRINT@452,"YOUR BROKER IS A ROGUE!!"
1190 FORD=1T050:NEXT:SOUND160,2
1200 NEXT XB
1210 PRINT@448,"HE'S OFF TO THE BAHAMAS WITH THECA
SH & COMMISSION ON THIS SALE";
1220 IF SHARES<0 THEN HELD(HOPE)=HELD(HOPE)+SHARES
ELSE CASH=CASH-INT(MRKT(HOPE)*SHARES/100)
1230 D1=1001:SHARES=0
1240 FORD=1T07000:NEXT
1250 RETURN
1260 REM*****FORECLOSE SUB****
1270 FOR XB=1T06
1280 GOSUB430 ' CLEAR
1290 FORD=1T050:NEXT:SOUND220,2
1300 PRINT@449,"BANK FORECLOSES YOUR OVERDRAFT!";
1310 FORD=1T050:NEXT:SOUND220,2
1320 NEXT

```

```

1330 PRINT@480,"THE BANK WILL SELL YOUR SHARES";:F
ORD=1TO4000:NEXT
1340 GOSUB430      ' CLEAR
1350 PRINT@448,"A BLOCK AT A TIME AND CHARGE YOU 3
5% FOR ITS EFFORTS!";
1360 FORD=1TO7000:NEXT
1370 Y=1
1380 FOR X=2 TO FIRMS
1390 IF PRICE(Y) < PRICE(X) THEN Y=X
1400 NEXT
1410 IF PRICE(Y) = 0 THEN D6=1
1420 HELD(Y)=0
1430 CASH=CASH + INT(PRICE(Y)*.65)
1440 PRICE(Y)=0:C=Y
1450 GOSUB2420      ' SHARE DISPLAY
1460 GOSUB2780      ' TOTAL
1470 GOSUB2550      ' TOTALS DSPLY
1480 IF D6=1 OR CASH>0 THEN1500
1490 GOTO 1370
1500 RETURN
1510 REM*****LAWYER SUB****
1520 GOSUB430      ' CLEAR MSG
1530 SOUND 16,8
1540 PRINT@448,"A LAWYER REQUESTS YOUR PRESENCE! D
O YOU GO ? (Y/N) ";
1550 KB$=INKEY$:IFKB$=""THEN1550
1560 IFKB$<>"Y"ANDKB$<>"N"THEN1550
1570 PRINTKB$;:IF KB$="N" THEN D2=1
1580 FORD=1TO1000:NEXT
1590 IF D2 THEN GOSUB430:PRINT@459,"VERY WISE!"; E
LSE GOSUB1630      ' CHOICE
1600 FORD=1TO2000:NEXT
1610 RETURN
1620 REM*****CHOICE SUB****
1630 GOSUB430      ' CLEAR MSG
1640 X=RND(0)
1650 IF X>.5 THEN PRINT@448,"YOU INHERIT ANOTHER
";:D2=1
1660 X=4000+RND(4000)
1670 IFD2=1 THEN PRINTX:CASH=CASH+X
1680 IFD2=0 THEN PRINT@448,"YOU ARE REQUIRED TO BR
IBE THE";
1690 IFD2=0 THEN PRINT@480,"FRAUD SQUAD! TRY ";X;
:CASH=CASH-X
1700 FORD=1TO7000:NEXT
1710 GOSUB2550      ' TOTALS DSPLY
1720 FORD=1 TO3000:NEXT
1730 D2=1
1740 GOSUB1060      ' DISASTERS
1750 RETURN
1760 REM*****HOSPITAL SUB****

```



```

1770 GOSUB430      ? CLEAR
1780 PRINT@448,"YOU FELL OFF A LADDER AND ARE INHO
SPITAL, IN A COMA FOR 6 WEEKS!";
1790 FORD=1T07000:NEXT
1800 FOR X=1T06
1810 WKS=WKS+1
1820 FORD=1T01000:NEXT
1830 GOSUB2580      ? SHARECALC
1840 GOSUB2780      ? TOTAL
1850 GOSUB2450      ? TABLE
1860 GOSUB2840      ? SETTLE
1870 IF WKS>26 THEN 4130 ?FINAL
1880 SOUND80,4
1890 NEXT
1900 D3=1
1910 RETURN
1920 REM*****PROBATE SUB****
1930 GOSUB430      ? CLEAR MSG
1940 MONEY=RND(10000)
1950 PRINT@452,"PROBATE MUST NOW BE PAID";
1960 PLAY"T302GGL8GL4GA+L8AL4AL8GL4GL8G-L4G":GOSUB
430
1970 PRINT@448,"PROBATE WAS JUDGED AT ";CASH+MONEY
;
1980 PRINT@480,"YOU EXCEEDED YOUR CREDIT LIMIT";
1990 FORD=1T05000:NEXT
2000 CASH=-MONEY
2010 GOSUB2550      ? TOTALS DSPLY
2020 D4=1:FORD=1T03000:NEXT
2030 GOSUB1060      ? DISASTERS
2040 RETURN
2050 REM*****POOLS SUB****
2060 GOSUB430      ? CLEAR MSG
2070 PRINT@448,"YOU'VE A 2ND DIVIDEND POOLS WIN!";
2080 MONEY=4000+RND(10000)
2090 FORD=96T0192:SOUND0,1:NEXT
2100 GOSUB430      ? CLEAR MSG
2110 HTM$(1)="DARLINGTON":HTM$(2)="BRISTOL R.":HTM
$(3)=" SWANSEA":HTM$(4)=" WOLVES "
2120 ATM$(1)="LIVERPOOL":ATM$(2)=" IPSWICH ":ATM$(
3)="TOTTENHAM":ATM$(4)=" MAN.UTD "
2130 PRINT@456,"YOU COLLECT ";MONEY;
2140 HD=RND(4):AD=RND(4)
2150 PRINT@480,HTM$(HD);" DREW WITH ";ATM$(AD);"! "
;
2160 FORD=1T05000:NEXT
2170 CASH=CASH+MONEY
2180 GOSUB2550      ? TOTALS DSPLY
2190 FORD=1T03000:NEXT:D5=1
2200 RETURN
2210 REM*****BANKRUPT SUB****
2220 GOSUB430      ? CLEAR MSG

```



```

2230 PRINT@456,"YOU ARE BANKRUPT!";
2240 FORD=151T00STEP-15:FORP=1T05
2250 SOUND0,1:NEXTP,D
2260 PRINT@483,"TRY WORKING FOR A LIVING";
2270 FORD=1T05000:NEXT
2280 GOTO 4130 ' END OF GAME
2290 RETURN
2300 REM*****HEADINGS SUB****
2310 CLS
2320 PRINT"COMPANY COST LAST YOU VALUE";
2330 PRINT" NAME (P) MOVE HOLD POUNDS";STRIN
G$(32,"-");
2340 FOR X=1T0FIRMS
2350 A$=RIGHT$(STR$(X),1)
2360 PRINT@64+32*X, A$;".":PRINT USING"% %";COY
$(X)
2370 NEXT
2380 PRINT@352,STRING$(32,"-")
2390 PRINT@416,STRING$(32,"-")
2400 RETURN
2410 REM*****SHARE DISPLAY****
2420 PRINT@71+32*C," ";:PRINT USING"#### +### ####
## ####";MRKT(C),CHANGE(C),HELD(C),PRICE(C);
2430 RETURN
2440 REM*****TABLE SUB****
2450 FOR X=1 TO FIRMS
2460 PRICE(X)=HELD(X)*MRKT(X)/100
2470 C=X
2480 GOSUB2420 ' SHARE DISPLAY
2490 IF CHANGE(X)<0 THEN SOUND1,4 ELSE SOUND 120,4
2500 FOR D=1T0100:NEXT
2510 NEXT
2520 GOSUB2550 ' TOTALS DSPLY
2530 RETURN
2540 REM*****TOTALS DISPLAY****
2550 PRINT@384,"";:PRINT USING"CASH #### (WEEK
##) ####";CASH,WKS,TTL;
2560 RETURN
2570 REM*****SHARECALC SUB****
2580 GOSUB430:PRINT@490,"PLEASE WAIT";
2590 X=1:CYCLE=CYCLE+1
2600 IF CYCLE>PERIOD THEN GOSUB2720 ' TREND
2610 FOR Z=1 TO FIRMS
2620 Y=RND(0)
2630 IF Y>BIAS+TREND+LUCK(Z) THEN X=-X
2640 TEMP=MRKT(Z)+X*RND(RANGE)
2650 IF TEMP<=LOW(Z) THEN 2620
2660 CHANGE(Z)=TEMP-MRKT(Z)
2670 MRKT(Z)=TEMP
2680 PRICE(Z)=INT(HELD(Z)*MRKT(Z)/100)
2690 NEXT
2700 RETURN

```

```

2710 REM*****TREND SUB****
2720 YY=RND(0):Y=1
2730 IFYY>0.5 THEN Y=-Y
2740 TREND=Y*INT(10*RND(0))/100
2750 CYCLE=0
2760 RETURN
2770 REM*****TOTAL SUB****
2780 TTL=0
2790 FOR X=1 TO FIRMS
2800 TTL=TTL+PRICE(X)
2810 NEXT
2820 RETURN
2830 REM*****SETTLE SUB****
2840 FOR X= 0 TO FIRMS
2850 FLAG(X)=0:NEXT
2860 RETURN
2870 REM*****INITIALISE SUB****
2880 CLEAR 5000
2890 FIRMS=8:CASH=10000:WKS=1
2900 TTL=0:CYCLE=0:PERIOD=5
2910 TREND=0:BIAS=0.4:SHARES=0
2920 RANGE=80:BC=0.12:LRATE=0.3
2930 CMMN=0:HOPE=0:C=0
2940 FOR X=1 TO FIRMS
2950 READ MKT(X),LOW(X),COY$(X)
2960 GOSUB2720 TREND
2970 LUCK(X)=TREND
2980 NEXT
2990 DATA 83,50,MINES
3000 DATA 38,25,CARS
3010 DATA 90,40,OIL
3020 DATA 60,50,TIN
3030 DATA 110,70,BANKS
3040 DATA 30,10,GOLD
3050 DATA 45,10,TEA
3060 DATA 80,25,RADIO
3070 REM*****INTRODUCTION****
3080 CLS0
3090 FOR I=64 TO 223
3100 READ CC:CO=RND(7)
3110 PRINT@I,CHR$(128+16*CO+CC);
3120 NEXT
3130 FOR I=288 TO 447
3140 READ CC:CO=RND(7)
3150 PRINT@I,CHR$(128+16*CO+CC);
3160 NEXT
3170 DATA 0,0,0,0,7,15,15,11,0,15,15,15,15,0,7,15,
15,11,0,7,15,15,11,0,15,0,0,15,0,0,0,0
3180 DATA 0,0,0,0,15,0,0,12,0,0,5,10,0,0,15,0,0,15
,0,15,0,0,12,0,15,0,1,14,0,0,0,0
3190 DATA 0,0,0,0,15,15,15,15,0,0,5,10,0,0,15,0,0,
15,0,15,0,0,0,0,15,15,15,0,0,0,0,0

```

```

3200 DATA 0,0,0,0,3,0,0,15,0,0,5,10,0,0,15,0,0,15,
0,15,0,0,3,0,15,0,4,11,0,0,0,0
3210 DATA 0,0,0,0,13,15,15,14,0,0,5,10,0,0,13,15,1
5,14,0,13,15,15,14,0,15,0,0,15,0,0,0,0
3220 DATA 0,0,15,2,1,15,0,7,15,15,11,0,15,15,15,11
,0,15,0,0,15,0,15,15,15,15,0,15,15,15,15,0
3230 DATA 0,0,15,15,15,15,0,15,0,0,15,0,15,0,0,15,
0,15,0,1,14,0,15,0,0,0,0,5,10,0,0
3240 DATA 0,0,15,5,10,15,0,15,15,15,15,0,15,15,15,
10,0,15,15,15,0,0,15,15,15,10,0,0,5,10,0,0
3250 DATA 0,0,15,0,0,15,0,15,0,0,15,0,15,0,0,15,0,
15,0,4,11,0,15,0,0,0,0,0,5,10,0,0
3260 DATA 0,0,15,0,0,15,0,15,0,0,15,0,15,0,0,15,0,
15,0,0,15,0,15,15,15,15,0,0,5,10,0,0
3270 FORD=1T07000:NEXT
3280 CLS6
3290 PRINT@33,"          YOU HAVE INHERITED";
3300 PRINT@65,"910,000 POUNDS.  BUT TO EARN";
3310 PRINT@97,"THE 900,000 YOU MUST PROVE";
3320 PRINT@129,"YOURSELF BY CONVERTING THE";
3330 PRINT@161,"SMALLER SUM OF 10,000 INTO";
3340 PRINT@193,"100,000 IN 26 WEEKS ON THE";
3350 PRINT@225,"STOCK MARKET.";STRING$(17," ");
3360 PRINT@289,"          EVERYONE WANTS YOUR";
3370 PRINT@321,"MONEY SO BE WISE.  YOUR BROKER";
3380 PRINT@353,"CHARGES 12% ON ALL DEALS OF";
3390 PRINT@385,"SALE OR PURCHASE OF SHARES.";
3400 PRINT@417,"BANKS CHARGE 30% ON OVERDRAFTS";
3410 PRINT@449,"AND TEND TO WITHDRAW SUDDENLY!";
3420 CL$=INKEY$
3430 KB$=INKEY$:IFKB$=""THEN3430
3440 CLS6
3450 PRINT@65,"          YOU MAY SET THE PARAMETERS";
3460 PRINT@97,"OF THE GAME , AND THIS ALLOWS";
3470 PRINT@129,"YOU TO EXPLORE SOME OF THE";
3480 PRINT@161,"FACTORS GOVERNING SUCCESS AND";
3490 PRINT@193,"THE PROFITABILITY OF STOCK";
3500 PRINT@225,"TRADING.";STRING$(22," ");
3510 PRINT@321,"          IF YOU WISH TO SET YOUR";
3520 PRINT@353,"OWN PARAMETERS  PRESS 'P'. ";
3530 PRINT@385,STRING$(30," ");
3540 PRINT@417,"IF NOT , PRESS ANY OTHER KEY.";
3550 KB$=INKEY$:IFKB$=""THEN3550
3560 IFKB$<>"P" THEN 110
3570 REM*****PARAMETERS****
3580 CLS:PRINT
3590 PRINT"HOW MANY FIRMS          (1-8)  ";
3600 T1$="1":T2$="8":GOSUB4020
3610 FIRMS=VAL(KB$)
3620 PRINT"HOW LONG ECONOMIC CYCLE (3-9)  ";
3630 T1$="3":T2$="9":GOSUB4020
3640 PERIOD=VAL(KB$)

```



```

3650 PRINT"RANGE SHARES CAN MOVE (30-79) ";
3660 T1$="3":T2$="8":GOSUB4020
3670 R1$=KB$
3680 T1$="0":T2$="9":GOSUB4020
3690 RANGE=VAL(R1$+KB$)
3700 PRINT"ECONOMIC CYCLE BIAS-% (30-49) ";
3710 T1$="3":T2$="4":GOSUB4020
3720 R1$=KB$
3730 T1$="0":T2$="9":GOSUB4020
3740 BIAS=VAL(R1$+KB$)/100
3750 PRINT"BROKERS COMMISSION-% (00-29) ";
3760 T1$="0":T2$="2":GOSUB4020
3770 R1$=KB$:T2$="9":GOSUB4020
3780 BC=VAL(R1$+KB$)/100
3790 PRINT"OVERDRAFT RATE-% (10-39) ";
3800 T1$="1":T2$="3":GOSUB4020
3810 R1$=KB$
3820 T1$="0":T2$="9":GOSUB4020
3830 LRATE=VAL(R1$+KB$)/100
3840 PRINT:PRINT"FOR EACH FIRM, YOU MUST DECIDE: -"
3850 PRINT"1. IF IT IS PROFITABLE GENERALLY";
3860 PRINT"2. ITS LOWEST SHARE VALUE"
3870 FOR X=1 TO FIRMS
3880 PRINT@384:PRINT@448:PRINT@416
3890 PRINT@385,CD$(X)
3900 PRINT@419,"PROFIT RATING (-9TO+9) ";
3910 T1$="+":T2$="-":GOSUB4020
3920 R1$=KB$
3930 T1$="0":T2$="9":GOSUB4020
3940 LUCK(X)=VAL(R1$+KB$)/100
3950 PRINT@451,"LOWEST PRICE (00-99) ";
3960 GOSUB4020:R1$=KB$:GOSUB4020
3970 LOW(X)=VAL(R1$+KB$)
3980 MRKT(X)=LOW(X)+RND(RANGE)
3990 NEXT
4000 GOTO110
4010 REM *****REPLY SUB****
4020 KB$=INKEY$:IFKB$=""THEN4020
4030 IF TS THEN IF ASC(KB$)=13 THEN TS=0:GOTO4070
4040 IFKB$<T1$ ORKB$>T2$THEN4020
4050 IFT1$="+"ANDKB$=","THEN4020
4060 PRINTKB$;
4070 RETURN
4080 REM*****CONGRATS SUB****
4090 CLS:FORD=1TO1000:NEXT:CLS
4100 PRINT@38," WELL DONE! ";:PRINT@70," Y
OU HAVE MADE THE ";:PRINT@102," MAGIC TOTAL AND
";:PRINT@134," CAN INHERIT THE ";:PRINT@166,"
REST OF THE LOOT ";
4110 RETURN
4120 REM*****END OF GAME****
4130 GOSUB430 'CLEAR MSG

```



```

4140 PRINT@450,"*** END OF SIMULATION ***";:FORD
=1T02000:NEXT
4150 IF CASH>100000 THEN GOSUB4090:GOTO4230
4160 FOR D=1T05000:NEXT
4170 CLS2
4180 PRINT@37,"          USELESS          ";:PRINT@69,ST
RING$(21," ");:PRINT@101," YOU HAVE FAILED ";
4190 PRINT@133," AND DO NOT GET THE ";:PRINT@165,
" REST OF THE LOOT! ";
4200 IF CASH+TTL>100000 THEN PRINT@229," SELL UP N
EXT TIME !! ";
4210 IF CASH>90000 THEN PRINT@227," AT LEAST YOU C
AME CLOSE!";
4220 IF CASH<10000 THEN PRINT@229," YOU CAN'T EVEN
KEEP ";:PRINT@261," WHAT YOU'RE GIVEN ";
4230 PRINT@390," PLAY AGAIN (Y/N) ";
4240 KB$=INKEY$:IFKB$=""THEN4240
4250 IFKB$<>"Y" AND KB$<>"N"THEN4240
4260 PRINTKB$;:IFKB$="Y"THEN RUN ELSE PLAY"T304B03
B":END

```

13. Four Stroke Engine

General Description

This program is a graphic demonstration of how a four stroke engine works. Once running, the program continues until you press 'S'. It only takes a few seconds to load and is great fun to watch. To develop the program you may wish to add labels to spark plug, valves, piston etc, but as it stands the valves open and shut, and airflow is indicated with arrows. The display may be frozen using Shift and @ , as in any Dragon program.

Detailed Description

Lines 10–220 Main structure, perpetual motion until 'S' is pressed.

230–480 The effective speed of the demonstration can be controlled by making the value of ST smaller in line 350. This in turn varies the number of 'frames' into the PT array.

490–610 The update of the graphical display including the piston and cam shaft.

620–720 This routine decides which valve to open and close as well as which of the four strokes is to be displayed.

730–830 Draws the cylinder.

840–1100 Opens and closes the valves, and draws the spark.

1110–1380 These routines display one of the four stroke phases.

Educational Notes

This is a good substitute for 'chalk and talk'. If you decide not to label the various parts of the engine then you can use the program to elicit responses as to what is happening from the class.

Program Listing

```
10 *****
20 *   FOUR STROKE ENGINE   *
25 *                         *
30 *   WRITTEN FOR THE DRAGON *
35 *                         *
40 *   BY   RAY NEW   1983   *
50 *****
60 PMODE 4,1
70 PCLS
80 DIM B(24,90)
90 GET(0,0)-(24,90),B
100 SCREEN 1,1
110 GOSUB 260 'INITIALISE
120 FOR I=0 TO FRAMES-1
130 P=0
140 GOSUB 520 'PLOT
150 IF I=TDC OR I=BDC THEN GOSUB 650 'STROKE
160 PUT(169,57)-(191,147),B,PSET
170 PUT(160,119)-(200,170),B
180 P=1
190 GOSUB 520 'PLOT
200 IF INKEY$="S" THEN END
210 NEXT I
220 GOTO 120
230 REM #
240 REM INITIALISE
250 REM #
260 S=0
270 LINE(50,18)-(197,18),PSET
280 DRAW"BM53,15;U5NL3R3BR3D5U3R5U2D5BR3U5NR5D2NR4
D3R5"
290 DRAW"BR14U5G3R5"
300 DRAW"BD2BR11R5U3L5U2R5BR3R6L3D5BR6U5R5D3L4F2R2
BR3U5R5D5NL5"
310 DRAW"BR3U5D3R2NE3F2R1BR3U5NR5D2NR4D3R5"
320 DRAW"BR11U5NR5D2NR4D3R5BR3U5F5U5"
330 DRAW"BR3NR5D5R5U2L2R2D2BR3NU5BR3U5F5U5BR3NR5D2
NR4D3R5"
340 GOSUB 760 'CYLINDER
350 ST=.261
360 FR=INT(2*3.14159/ST)
370 C=20
380 R=90
390 TDC=0
400 BDC=INT(FR/2)
410 DIM PT(FR)
420 FOR I=0 TO FR-1
430 PT(I)=INT(C*SIN(I*ST+ST*.5))
440 NEXT I
450 PT(I)=PT(0)
```

```

460 J=1
470 I=0
480 RETURN
490 REM #
500 REM PLOT
510 REM #
520 X=180:Y=150
530 M=I+P+6
540 N=INT(M/FR)
550 M=M-FR*N
560 X1=X+PT(I+P):Y1=Y+PT(M)
570 Y2=R+PT(M)
580 LINE(X,Y)-(X1,Y1),PSET
590 LINE(X1,Y1)-(X,Y2),PSET
600 LINE(X-12,Y2-6)-(X+12,Y2),PSET,BF
610 RETURN
620 REM #
630 REM STROKE
640 REM #
650 S=S+1
660 PUT(20,85)-(110,110),B,PSET
670 ON S GOSUB 1140,1300,1360,1230
680 IF S=4 THEN S=0
690 IF S=0 OR S=3 THEN GOSUB 960
700 IF S=0 OR S=1 THEN GOSUB 870
710 IF S=2 OR S=3 THEN GOSUB 1060
720 RETURN
730 REM #
740 REM CYLINDER
750 REM #
760 DRAW"BM168,115;U70R24D70"
770 DRAW"BM170,45;U4L20BU4R24D8"
780 DRAW"BM130,39;R12H4D8E4"
790 PAINT(140,40):PAINT(140,38)
800 DRAW"BM190,45;U4R20BU4L24D8"
810 DRAW"BM218,39;R12H4D8E4"
820 PAINT(228,40):PAINT(228,38)
830 RETURN
840 REM #
850 REM IN VALVE
860 REM #
870 IFS=0THEND1$="C0":D2$="C5"ELSE D1$="C5":D2$="C0"
"
880 DRAWD1$
890 DRAW"BM171,45;R2BD6"
900 DRAWD2$
910 DRAW"L2"
920 DRAW"C5"
930 REM #
940 REM OUT VALVE
950 REM #

```



```

960 IFS=3THEND1$="C0":D2$="C5"ELSE D1$="C5":D2$="C0"
"
970 DRAW D1$
980 DRAW"BM189,45;L2BD6"
990 DRAW D2$
1000 DRAW"R2"
1010 DRAW"C5"
1020 RETURN
1030 REM #
1040 REM SPARK
1050 REM
1060 IF S=2THEN D1$="C5"ELSE D1$="C0"
1070 DRAW D1$
1080 DRAW"BM180,46;G3R6G3"
1090 DRAW"C5"
1100 RETURN
1110 REM #
1120 REM DISPLAY COMPRESSION
1130 REM #
1140 DRAW"BM20,90;R5L5D5R5BR3U5R5D5L5"
1150 DRAW"BR8U5F3E3D5BR3U5R5D3L4BD2BR7"
1160 DRAW"U5R5D3L4F2R2"
1170 DRAW"BR3U5R5L5D2R4L4D3R5"
1180 DRAW"BR3R5U3L5U2R5BR3BD5R5U3L5U2R5BR3D5BR3U5R
5D5L5BR8U5F5U5"
1190 RETURN
1200 REM #
1210 REM DISPLAY INDUCTION
1220 REM #
1230 DRAW"BM28,95;U5D5BR3U5F5U5BD5BR3"
1240 DRAW"U5R2F2D1G2L2BR7U5D5R5U5BR3R5L5D5R5BR6U5L
3R6"
1250 DRAW"BR3D5BR3U5R5D5L5R5BR3U5F5U5"
1260 RETURN
1270 REM #
1280 REM DISPLAY POWER
1290 REM #
1300 DRAW"BM44,95;U5R5D3L5D2BR8U5R5D5L5BR8"
1310 DRAW"U5D5E3F3U5BR3R5L5D2R4L4D3R5BR3U5R5D3L4F2
R2"
1320 RETURN
1330 REM #
1340 REM DISPLAY EXHAUST
1350 REM #
1360 DRAW"BM36,95;U5R5L5D2R4L4D3R5BR3E5G2NH2F3BR3U
5D2R5U2D5BR3U5R5D2L5R5D3"
1370 DRAW"BR3U5D5R5NU5BR3R5U3L5U2R5BR3R6L3D5"
1380 RETURN

```

14. Attributes

General Description

This program tries to provide a forum for you to develop question and answer programs for the classroom. Two possible uses for this program are given here. The science data was developed for less able fourth years in a girls' school. The geography data is included to show the program's adaptability.

The program starts with an instruction page on which the scoring and rules are displayed. You start with 50 points, and lose a small number of points for the help that you require – the largest loss being 5 points for a fruitless guess. Entry is by typing the appropriate letter followed by the ENTER key. If you guess correctly, then you are awarded an extra 50 points and given the choice to play again with your new total.

Adaptability

For those who wish to develop their own data, the following points should be noted:

Line 660 This contains the number of items of data in the program. If this is changed it **MUST** be changed to an **EVEN** number.

Lines 740–810 These contain the eight questions needed for the program and must be altered according to the program's needs.

Lines 900–930 These contain the introduction and must also be changed according to the program's needs.

Data Codes

The clue to these data codes is in line 730. Each question has a number equivalent to the question. Thus question one has value

1 while question eight has value 128. 'PARIS' has a code 1 because it is a capital and code 4 because it is on a river. Add these together and the code for 'PARIS' is 5 . . . QED.

Detailed Description

Lines 10–140 Remarks and calls the introduction and initialisation routines.

150–320 Main body of program because this is where the screen is constantly updated according to the question asked.

330–370 Works out whether or not the answer to the question being asked is YES or NO.

380–420 Inputs your guess and sees if it is right or wrong.

430–470 Updates your score.

480–550 Checks to see what question number you are asking for.

560–610 Delay subroutine.

620–850 Sets the questions and selects a random city for you to guess.

860–1050 Gives you an introduction to the game.

1060–1250 Lets you select whether you want general help, special help or to return to the game.

1260–1380 General help routine.

1390–1600 Special help routine.

1620–2090 Various routines to end the game.

2100–2220 Asks whether or not you wish to play again and acts accordingly.

Educational Notes

This program should be used in small groups, a maximum of four, round a single machine. The structure of the program lends itself to the less able, unless for the more able you are prepared to generate a large and challenging amount of data. The only point that the users of the program must realise, is to use the general help page for the set of answers. Looking at the science data, you can understand why: a sulphur rod and sulphur powder have some different physical characteristics and so sulphur on its own cannot be a suitable response. Similarly the ruler must be wooden!

Program Listing

```
10 REM+++++++
20 REM+      ATTRIBUTES      +
30 REM+      +
40 REM+ WRITTEN FOR THE DRAGON +
50 REM+      +
60 REM+ BY GAVIN J.CRADLE 1983 +
70 REM+++++++
80 DIM QUESTION$(8),CLUE$(8)
90 CLEAR
100 A=50
110 SCORE=A
120 CLS
130 GOSUB 620
140 GOSUB 860
150 REM+++++++
160 REM      screen display
170 REM+++++++
180 CLS
190 PRINT HINT$
200 PRINT "X=TO MAKE A GUESS,Z=TO GIVE UP","H=TO A
SK FOR HELP,A=ASK QUESTION";
210 PRINT STRING$(32,"=");
220 FOR X=1 TO 8
230 PRINT X;QUESTION$(X);CLUE$(X)
240 NEXT X
250 PRINT "YOUR SCORE IS ";SCORE
260 PRINT @416,"WHAT DO YOU WISH TO DO ?"
270 INPUT Q$
280 IF Q$="H" THEN GOSUB 1060
290 IF Q$="X" THEN 380
300 IF Q$="Z" THEN 1750
310 IF Q$="A" THEN GOSUB 480:GOSUB 330:GOTO 150
320 GOTO 260
330 REM+++++++
340 REM  analyse and results
350 REM+++++++
360 IF VLUE AND BV(QUESTION) THEN CLUE$(QUESTION)=
" yes" ELSE CLUE$(QUESTION)=" no"
370 RETURN
380 REM.....guess.....
390 INPUT "ENTER YOUR GUESS >";GUESS$
400 SOUND 50,1
410 IF GUESS$=WORD$ THEN 1840 ELSE PRINT " <wrong>
LOSE 5 POINTS":GOSUB 560:GOSUB 430
420 GOTO 150
430 REM.....score.....
440 IF GUESS$<>WORD$ THEN SCORE=SCORE-5:GOTO 460
450 IF GUESS$<>WORD$ AND HELP<>1 THEN SCORE=SCORE-
1
460 IF SCORE<1 THEN 1610
```



```

470 GOTO 150
480 REM+++++
490 REM          validate
500 REM+++++
510 PRINT @448,"PRESS THE APPROPRIATE NUMBER."
520 B$=INKEY$
530 IF B$<"1" OR B$>"8" THEN 520
540 QUESTION$=VAL(B$)
550 RETURN
560 REM+++++
570 REM          delay
580 REM+++++
590 FOR DE=1 TO 2000
600 NEXT DE
610 RETURN
620 REM+++++
630 REM          initialisation
640 REM+++++
650 RESTORE
660 HOW MANY=20
670 MESSAGE$="ITS COLOUR IS"
680 ST=RND(HOW MANY)
690 FOR TM=1 TO ST
700 READ WORD$,VLUE,HINT$
710 NEXT TM
720 HINT$=MESSAGE$+" "+HINT$
730 BV(1)=1:BV(2)=2:BV(3)=4:BV(4)=8:BV(5)=16:BV(6)
=32:BV(7)=64:BV(8)=128
740 QUESTION$(1)="DOES IT FLOAT IN WATER>"
750 QUESTION$(2)="DOES IT DISSOLVE>"
760 QUESTION$(3)="IS IT A CONDUCTOR>"
770 QUESTION$(4)="DOES IT SNAP IF BENT>"
780 QUESTION$(5)="WILL IT REACT WITH ACID>"
790 QUESTION$(6)="CAN A BUNSEN LIGHT IT>"
800 QUESTION$(7)="WILL A BUNSEN MELT IT>"
810 QUESTION$(8)="IS IT EASY TO SCRATCH>"
820 FOR X=1 TO 8
830 CLUE$(X)=" ???"
840 NEXT X
850 RETURN
860 REM+++++
870 REM          introduction
880 REM+++++
890 CLS
900 PRINT TAB(8);"SCIENCE TESTER"
910 PRINT "THIS PROGRAM SEES IF YOU CAN"
920 PRINT "WORK OUT WHAT THE ITEM IS FROM"
930 PRINT "ONLY A FEW CLUES."
940 PRINT "YOU ARE CHARGED 1 PT. FOR EACH"
950 PRINT "QUESTION THAT YOU ASK."
960 PRINT "YOU ARE CHARGED 5 PTS. FOR"
970 PRINT "SPECIAL HELP,GENERAL HELP IS"

```

```

980 PRINT "free!"
990 PRINT "TYPE 'H' FOR SPECIAL & GENERAL"
1000 PRINT " HELP."
1010 PRINT "REMEMBER TO USE help BEFORE ANY"
1020 PRINT "GUESSES."
1030 PRINT
1040 PRINT "  press any key to carry on."
1050 IF INKEY$="" THEN 1050 ELSE RETURN
1060 REM+++++++
1070 REM  help menu
1080 REM+++++++
1090 CLS
1100 PRINT TAB(12);"help"
1110 PRINT
1120 PRINT "PRESS 'S' FOR SPECIAL."
1130 PRINT "PRESS 'G' FOR GENERAL."
1140 PRINT "PRESS 'R' TO RETURN TO GAME."
1150 PRINT
1160 PRINT "REMEMBER THAT YOU ARE CHARGED 3"
1170 PRINT "POINTS FOR EVERY PIECE OF "
1180 PRINT "SPECIAL HELP THAT YOU RECEIVE."
1190 PRINT
1200 PRINT "WHAT IS YOUR CHOICE ?"
1210 INPUT Q$
1220 IF Q$="S" THEN 1390
1230 IF Q$="G" THEN 1260
1240 IF Q$="R" THEN 150
1250 GOTO 1060
1260 REM.....general help.....
1270 CLS
1280 PRINT TAB(8);"general help"
1290 PRINT "HERE ARE THE POSSIBLE ANSWERS:-"
1300 RESTORE
1310 FOR N=1 TO HOW MANY
1320 READ AN$,BV,CN$
1330 PRINT AN$,
1340 NEXT N
1350 PRINT "PRESS ANY KEY TO RETURN TO MENU"
1360 IF INKEY$="" THEN 1360
1370 CLS
1380 GOTO 1090
1390 REM....special.help.....
1400 CLS
1410 PRINT TAB(9);"special help"
1420 PRINT "WHICH QUESTION DO YOU WANT THE  HELP F
OR ?"
1430 FOR X=1 TO 8
1440 PRINT X;QUESTION$(X)
1450 NEXT X
1460 GOSUB 480
1470 W=BV(QUESTION)
1480 RESTORE

```

```

1490 CLS
1500 PRINT QUESTION$(QUESTION)
1510 PRINT "FOR ALL THESE BELOW THE ANSWER IS yes
"
1520 FOR X=1 TO HOW MANY
1530 READ HELP$,Q,Z$
1540 IF W AND Q THEN PRINT HELP$,
1550 NEXT X
1560 PRINT @416,"PRESS ANY KEY TO RETURN TO MENU."
1570 IF INKEY$="" THEN 1570
1580 SCORE=SCORE-3
1590 CLS
1600 GOTO 1090
1610 REM+++++++
1620 REM end game routines
1630 REM+++++++
1640 CLS 0
1650 FOR SO=255 TO 200 STEP -1:SOUND SO,1:NEXT SO
1660 FOR SO=200 TO 255:SOUND SO,1:NEXT SO
1670 PRINT @77,"YOU";
1680 PRINT @141,"HAVE";
1690 PRINT @205,"RUN";
1700 PRINT @269,"OUT";
1710 PRINT @333,"OF";
1720 PRINT @397,"POINTS.";
1730 GOSUB 560
1740 GOTO 2100
1750 REM.....surrender.....
1760 CLS 0
1770 FOR TM=1 TO 30
1780 PRINT @235,"CHICKEN";
1790 PRINT @235,"chicken";
1800 NEXT TM
1810 GOSUB 560
1820 GOSUB 1940
1830 GOTO 2100
1840 REM.....win.....
1850 CLS 3
1860 PRINT @138,"congratulations!";
1870 PRINT @237,"YOU ARE";
1880 PRINT @301,"CORRECT!";
1890 PRINT @416,"IF YOU CHOOSE TO PLAY AGAIN,YOU"
1900 PRINT "START WITH 50 extra POINTS!!"
1910 A=SCORE+50
1920 GOSUB 560
1930 GOTO 2100
1940 REM.....close.....
1950 CLS
1960 PRINT @10,"YOU HAVE LOST!"
1970 PRINT
1980 FOR X=1 TO 8
1990 PRINT X;QUESTION$(X);

```

```

2000 PRINT CLUE$(X)
2010 NEXT X
2020 PRINT
2030 PRINT "THE ANSWER IS ";WORD$
2040 PRINT
2050 PRINT "      press any key to carry on"
2060 IF INKEY$="" THEN 2060
2070 QUESTION=0
2080 A=50
2090 RETURN
2100 REM.....final.....
2110 CLS
2120 PRINT @128,"DO YOU WANT TO PLAY AGAIN (Y/N)"
2130 INPUT Q$
2140 IF Q$="Y" AND GUESS$=WORD$ THEN 110 ELSE IF Q
$="Y" THEN RUN
2150 IF Q$="N" THEN GOSUB 2170
2160 GOTO 2140
2170 REM.....the end.....
2180 CLS
2190 PRINT @231,"GOOD-BYE FOR NOW."
2200 GOSUB 560
2210 CLS
2220 END
2230 REM.....data.....
2240 DATA WOODEN RULER,169,BROWN
2250 DATA SAND,0,YELLOW
2260 DATA SALT,2,WHITE
2270 DATA SUGAR,98,WHITE
2280 DATA POLYTHENE SHEET,225,TRANSPARENT
2290 DATA IRON BAR,20,GREY
2300 DATA PLASTICINE,224,ANY COLOUR
2310 DATA GLASS ROD,200,TRANSPARENT
2320 DATA PERSPEX VISOR,233,TRANSPARENT
2330 DATA GOLD COIN,132,YELLOW
2340 DATA GRAPHITE ROD,172,BLACK
2350 DATA BRASS BAR,20,YELLOW
2360 DATA SULPHUR ROD,232,YELLOW
2370 DATA DIAMOND,32,TRANSPARENT
2380 DATA MARBLE,152,ANY COLOUR
2390 DATA CANDLEWAX,233,ANY COLOUR
2400 DATA LEAD PIPE,148,GREY
2410 DATA CHALK POWDER,16,WHITE
2420 DATA ICE BLOCK,203,TRANSPARENT
2430 DATA PAVING SLAB,136,GREY

```


THE FOLLOWING LINES SHOULD BE CHANGED FOR THE PROGRAM TO USE THE GEOGRAPHY DATA.

```
660 HOW MANY=24
670 MESSAGE$="THE CONTINENT IS"
740 QUESTION$(1)="IS IT A CAPITAL CITY>"
750 QUESTION$(2)="IS IT ON A COASTLINE>"
760 QUESTION$(3)="IS IT ON A RIVER>"
770 QUESTION$(4)="IS IT A HOT COUNTRY>"
780 QUESTION$(5)="DO THEY SPEAK ENGLISH>"
790 QUESTION$(6)="DO THEY HAVE A MONARCH>"
800 QUESTION$(7)="IS IT A CITY STATE>"
810 QUESTION$(8)="WAS IT A BRITISH COLONY>"
900 PRINT TAB(9);"WORLD CITIES."
920 PRINT "WORK OUT WHAT THE CITY IS FROM"
2230 REM.....data.....
2240 DATA PARIS,5,EUROPE
2250 DATA HONG KONG,219,ASIA
2260 DATA CANBERRA,185,AUSTRALASIA
2270 DATA LONDON,53,EUROPE
2280 DATA MOSCOW,5,EUROPE
2290 DATA LOS ANGELES,18,NORTH AMERICA
2300 DATA CALCUTTA,142,ASIA
2310 DATA RIO DE JANEIRO,6,SOUTH AMERICA
2320 DATA CANTON,12,ASIA
2330 DATA TEHERAN,9,ASIA
2340 DATA LUSAKA,137,AFRICA
2350 DATA WEST BERLIN,4,EUROPE
2360 DATA FLORENCE,4,EUROPE
2370 DATA HO CHI MIN CITY,12,ASIA
2380 DATA CHICAGO,148,NORTH AMERICA
2390 DATA NEW YORK,150,NORTH AMERICA
2400 DATA BUDAPEST,5,EUROPE
2410 DATA CRAKOW,4,EUROPE
2420 DATA CAIRO,15,AFRICA
2430 DATA KINGSTON,187,CENTRAL AMERICA
2440 DATA STOCKHOLM,37,EUROPE
2450 DATA COLOGNE,4,EUROPE
2460 DATA MONTE CARLO,99,EUROPE
2470 DATA THE VATICAN,65,EUROPE
```

15. Interference

General Description

This program will prove invaluable to physics teachers trying to explain the thorny problems of interference to A-level classes. The program produces two sets of overlapping semi-circular waves as a simulation of two-slit diffraction. The simulation produces clearly identifiable interference patterns which can be used in parallel with verbal explanation. The user instructions go to some lengths to point students in the right direction, and ask the user to supply sample values for slit separation and wavelength.

Detailed Description

Lines 10–430 The main program.

70–250 Heading page with a colourful border.

260–340 Various procedures are called from the introduction to re-run.

350–430 Checks for re-run and if not selected gives the user 10 seconds to change their minds before calling the 'goodbye' procedure.

440–720 Instructions.

730–950 This requests user data, the slit separation(D) and wavelength(L). The scaling factor of 16 is introduced so that the values of wavelength may be more realistic for visible light.

960–1020 Slits are drawn.

1030–1100 The incident beam is assumed to be a parallel one. The spacing between waves is L .

1110–1230 Waves from the left-hand slit are drawn by lines 1160–1190. Right-hand slit waves are drawn by lines 1200–1220.

1240–1500 This procedure draws lines in the direction of constructive interference.

1510-1710 Another set of waves to be superimposed on the first?

1720-1800 Calculation of ARCSIN, this could be written more elegantly using

$$\text{ARCSIN } X = \text{ARCTAN} \left\{ \frac{X}{\sqrt{1-X^2}} \right\}$$

1810-2050 Writing on Hi-Res screen.

2060-2230 Do you wish to start again?

2240-2310 Goodbye.

Educational Notes

In the classroom, this simulation could effectively replace existing audio-visual demonstrations. If used with A-Level groups (with a well-adjusted TV monitor), measurements of angle can be made directly from the screen to verify (or otherwise) the students' predictions.

For the best results, I would suggest a slit separation of less than 1000mm and wavelengths between 200 and 300mm. For example $D=800$, $L=230$ mm.

Program Listing

```
10 *****
20 *          INTERFERENCE          *
25 *                                           *
30 * WRITTEN FOR THE DRAGON *
35 *                                           *
40 * BY      RAY NEW      1983 *
50 *****
60 CLS
70 PRINT@68,"I";
80 PRINT@102,"N":PRINT@136,"T";
90 PRINT@170,"E":PRINT@204,"R";
100 PRINT@238,"F":PRINT@272,"E";
110 PRINT@306,"R":PRINT@340,"E";
120 PRINT@374,"N":PRINT@408,"C";
130 PRINT@442,"E";
140 FOR T=1 TO 3
150 SOUND25,3
160 FOR I=0 TO 63
170 SET(I,0,RND(9)-1)
180 SET(I,31,RND(9)-1)
190 NEXT I
```



```

200 SOUND225,3
210 FOR I=0 TO 31
220 SET (0,I,RND(9)-1)
230 SET (63,I,RND(9)-1)
240 NEXT I
250 NEXT T
260 GOSUB 470 'HELP
270 GOSUB 760 'DATA
280 Q$="N"
290 PMODE4,1:PCLS:SCREEN1,1
300 GOSUB 990 'SLITS
310 GOSUB 1060 'BEAM
320 GOSUB 1140 'WAVES
330 GOSUB 1300 'MAXIMA
340 GOSUB 1540 'RERUN
350 IF IK$="Y" THEN PMODE4,1:SCREEN1,1:GOTO310
360 GOSUB 2090 'START AGAIN ?
370 IF SA$="Y" THEN 270
380 FOR I=1 TO 1500
390 IF INKEY$ <> "" THEN I=1500:LC$="Y"
400 NEXT I
410 IF LC$="Y" THEN 270
420 GOSUB 2270 'GOODBYE
430 END
440 REM #
450 REM HELP
460 REM #
470 CLS
480 DU$=INKEY$
490 PRINT"-----INTERFERENCE-----";
500 PRINT@33,"DO YOU WANT INSTRUCTIONS (Y/N)";
510 IK$=INKEY$
520 IF IK$="" THEN 510
530 IF IK$="N" THEN 720
540 CLS
550 PRINT"-----INTERFERENCE-----";
560 PRINT@33,"THIS PROGRAM PRODUCES A";
570 PRINT@64,"GRAPHICAL REPRESENTATION OF AN";
580 PRINT@96,"INTERFERENCE PATTERN PRODUCED BY";
590 PRINT@128,"A PAIR OF COHERENT LIGHT SOURCES";
600 PRINT@161,"YOU WILL HAVE TO SUPPLY VALUES";
610 PRINT@192,"OF SLIT SEPARATION BETWEEN 60 NM";
620 PRINT@224,"AND 1280 NM, AND OF WAVELENGTH";
630 PRINT@256,"BETWEEN 60 NM AND 600 NM.";
640 PRINT@289,"YOU MAY VIEW THE WAVES PRODUCED";
650 PRINT@320,"FROM EACH SLIT ALTERNATELY";
660 PRINT@352,"OR ONE SLIT AT A TIME.";
670 PRINT@385,"IF YOU WISH TO STUDY THE EFFECT";
680 PRINT@416,"ON THE PATTERN OF CHANGING THE";
690 PRINT@448,"WAVELENGTH, YOU WILL BE GIVEN THE";
700 PRINT@480,"OPPORTUNITY AT THE END.";
710 IF INKEY$="" THEN 710

```



```

720 RETURN
730 REM #
740 REM DATA
750 REM #
760 CLS
770 PRINT"-----INTERFERENCE-----";
780 PRINT@32,"SLIT SEPARATION, IN NM";
790 INPUT DD$
800 DN=VAL(DD$)
810 IF DN<60 OR DN>1280 THEN PRINT@64," (OUT OF R
ANGE 60-1280)";:PRINT@32,STRING$(32," ");:GOTO780
820 PRINT@128,"WAVELENGTH OF LIGHT, IN NM";
830 INPUT LL$
840 LN=VAL(LL$)
850 IF LN<60 OR LN>600 THEN PRINT@160," (OUT OF
RANGE 60-600)";:PRINT@128,STRING$(32," ");:GOTO820
860 PRINT@224,"DO YOU WANT THE WAVES TO BE FROM";
870 PRINT@256,"EACH SLIT ALTERNATELY (Y/N)";
880 IK$=INKEY$
890 IF IK$<>"Y"AND IK$<>"N"THEN880
900 IF IK$="Y" THEN F1$="Y"
910 PRINT" ";IK$;
920 FORI=1TO300:NEXTI
930 L=(LN/16)
940 D=(DN/16)
950 RETURN
960 REM #
970 REM SLITS
980 REM #
990 LINE(0,150)-(124-D/2,160),PSET,BF
1000 LINE(132-D/2,150)-(124+D/2,160),PSET,BF
1010 LINE(132+D/2,150)-(255,160),PSET,BF
1020 RETURN
1030 REM #
1040 REM BEAMS
1050 REM #
1060 FOR ST=0 TO 41 STEP L
1070 LINE(0,191-ST)-(255,191-ST),PSET
1080 FOR I=1TO400:NEXTI
1090 NEXT ST
1100 RETURN
1110 REM #
1120 REM WAVES
1130 REM #
1140 C1=128-D/2
1150 C2=128+D/2
1160 FOR R=0 TO 140 STEP L
1170 CIRCLE(C1,150),R,5,1,.5,0
1180 IF F1$="Y" THEN 1210
1190 NEXTR
1200 FOR R=0 TO 140 STEP L
1210 CIRCLE(C2,150),R,5,1,.5,0

```

```

1220 NEXTR
1230 RETURN
1240 REM #
1250 REM MAXIMA
1260 REM CALCULATE CONSTRUCTIVE
1270 REM INTERFERENCE USING
1280 REM NL=DSINA
1290 REM #
1300 N=0
1310 II=0
1320 DU$=INKEY$
1330 SOUND150,2
1340 GOSUB 1840 'ASK IF WANTED
1350 IK$=INKEY$
1360 IF IK$<>"N"AND IK$<>"Y"THEN1350
1370 DRAW"C0":GOSUB1840:DRAW"C5"
1380 IF IK$="N"THEN1500
1390 B=N*L/D
1400 IF B>=1 THEN 1470
1410 N=N+1
1420 GOSUB 1750 'CALCULATE ARCSIN
1430 IF A<0 THEN A=0
1440 LINE(128,150)-(128+128*B,150-128*COS(A)),PSET
1450 LINE(128,150)-(128-128*B,150-128*COS(A)),PSET
1460 GOTO1390
1470 DU$=INKEY$
1480 SOUND150,2
1490 IF INKEY$=""THEN1490
1500 RETURN
1510 REM #
1520 REM RERUN
1530 REM #
1540 IF Q$="Y"THEN IK$="":RETURN
1550 Q$="Y"
1560 CLS
1570 PRINT"-----INTERFERENCE-----"
1580 PRINT@70,"D =";DN;"      L1 =";LN;
1590 PRINT@128,"DO YOU WISH TO SUPERIMPOSE A";
1600 DU$=INKEY$
1610 PRINT@160,"SECOND WAVELENGTH ? (Y/N)";
1620 IK$=INKEY$
1630 IF IK$<>"N"AND IK$<>"Y"THEN1620
1640 IF IK$="N"THEN1710
1650 PRINT@224,"SECOND WAVELENGTH, IN NM";
1660 INPUT L2$
1670 N2=VAL(L2$)
1680 IF N2<60 OR N2>600 THEN PRINT@256," (OUT OF
RANGE 60-600)";:PRINT@224,STRING$(32," ");:GOTO165
0
1690 L=(N2/16)
1700 FOR I=1TO500:NEXTI
1710 RETURN

```

```

1720 REM #
1730 REM CALCULATE ARCSIN
1740 REM #
1750 FOR I=II TO 1.57 STEP .05
1760 A1=SIN(I)
1770 II=I
1780 IF A1>=B THEN A=I-.025:I=1.57
1790 NEXT I
1800 RETURN
1810 REM #
1820 REM ASK IF EMPHASIS WANTED
1830 REM #
1840 DRAW"BM40,6;U5NR5D2NR4D3R5" ?E
1850 DRAW"BR3U5F3E3D5" ?M
1860 DRAW"BR3U5R5D2L5D3" ?P
1870 DRAW"BR8U5D2R5NU2D3" ?H
1880 DRAW"BR3U5R5D2NL5D3" ?A
1890 DRAW"BR3R5U3L5U2R5" ?S
1900 DRAW"BR3D5" ?I
1910 DRAW"BR3R5U3L5U2R5" ?S
1920 DRAW"BD5BR3U5NR5D2NR4D3R5"
1930 DRAW"BR11NU5BR3U5F5U5" ?IN
1940 DRAW"BR3R6L3D5" ?T
1950 DRAW"BR6U5NR5D2NR4D3R5" ?E
1960 DRAW"BR3U5R5D3L4F2R2" ?R
1970 DRAW"BR3U5NR5D2NR4D3" ?F
1980 DRAW"BR8U5NR5D2NR4D3R5" ?E
1990 DRAW"BR3U5R5D3L4F2R2" ?R
2000 DRAW"BR3U5NR5D2NR4D3R5" ?E
2010 DRAW"BR3U5F5U5" ?N
2020 DRAW"BD5BR3U5NR5D5R5" ?C
2030 DRAW"BR3U5NR5D2NR4D3R5" ?E
2040 DRAW"BR11U1BU2R5U2L5" ??
2050 RETURN
2060 REM #
2070 REM START AGAIN ?
2080 REM #
2090 CLS
2100 SA$="":LC$=""
2110 PRINT"-----INTERFERENCE-----";
2120 DU$=INKEY$
2130 PRINT@64,"DO YOU WISH TO START AGAIN (Y/N)";
2140 IK$=INKEY$
2150 IF IK$<>"Y"ANDIK$<>"N"THEN2140
2160 SA$=IK$
2170 IF SA$="Y"THEN2230
2180 CLS
2190 PRINT"-----INTERFERENCE-----";
2200 PRINT@64,"YOU HAVE 10 SECONDS TO CHANGE";
2210 PRINT@96,"YOUR MIND----ANY KEY WILL DO";
2220 FOR I=1TO1000:NEXT I
2230 RETURN

```

```

2240 REM #
2250 REM GOODBYE
2260 REM #
2270 CLS
2280 PRINT"-----INTERFERENCE-----";
2290 PRINT@76,"GOODBYE";
2300 FOR I=1TO1000:NEXTI
2310 RETURN

```


16. Meter Reader

General Description

This program has been designed to help students read electrical meters correctly. Its vivid graphical display and lucid instructions mean that, after initial preparation, this program can be used unsupervised.

Detailed Description

Lines 80–90 Reserves graphics area and string space; dimensions arrays for graphic text and meter scale ranges.

100–120 User-defined function to compute PI and initialise variables.

130–320 Data for DRAW command, to create text on high resolution graphics screen.

330–410 Introduction and main loop of program.

420–580 Subroutine to draw meter. This is performed during the Introduction display, and creates the meter display in pages 5–8 of the graphics area.

590–930 Subroutine to draw the scale. The meter is copied to pages 1–4 of the graphics area and the selected scale and subdivisions are added.

940–1090 Chooses the full scale deflection and subdivisions.

1100–1320 Draws the pointer. 1190 calculates the number of subdivisions on the chosen scale and 1200 converts into angle of deviation.

1330–1850 This subroutine deals with the computer-user interaction.

1860–1990 Fancy title.

2000–2270 Instructions. The subroutine to draw the meter is performed while the user is reading (hopefully) the instructions.

In the classroom this program could be used in conjunction with a real meter to teach scale reading. It also lends itself to being used in a remedial situation with small groups of students who either missed or misunderstood the original lesson.

To modify the program to suit particular meters, just change the DATA values in lines 1010 to 1040. Choose and test your changes carefully – there is a limited set of numbers available in the array of DRAW strings.

Program Listing

```
10 REM*****
20 REM*   THE METER READER   *
30 REM*   *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*   *
60 REM* BY LAURENCE OWEN 1983 *
70 REM*****
80 PCLEARB: CLEAR 1000
90 DIM N$(20), RANGE(3,3)
100 DEF FNP(X)=4.0*ATN(1.0)
110 PI=FNP(X)
120 SCORE=0: TRY=0
130 REM***SET UP GRAPHIC TEXT**
140 N$(0)="BR7L4U6R4D6BR1"
150 N$(1)="BR5U6BM+3,+6"
160 N$(2)="BM+3,-6R4D4L4D2R4BR1"
170 N$(3)="BR3R4U6L4BM+1,+3;R3BM+1,+3"
180 N$(4)="BM+7,-2;L5U4BR4D6BR2"
190 N$(5)="BR3R4U4L4U2R4BM+1,+6"
200 N$(6)="BM+3,-6D6R4U3L4BM+5,+3"
210 N$(7)="BM+3,-5;U1R4D6BR1"
220 N$(8)="BR7L4U6R4D3L4R4D3BR1"
230 N$(9)="BR7U6L4D3R4BM+1,+3"
240 N$(10)=N$(1)+N$(0)
250 N$(11)=N$(1)+N$(1)
260 N$(12)=N$(1)+N$(2)
270 N$(15)=N$(1)+N$(5)
280 PT$=N$(0)+"BR3BU3L1U1R1D1BD3"
290 N$(16)="BL8"+PT$+N$(2)  ^0.2
300 N$(17)="BL4"+PT$+N$(4)  ^0.4
310 N$(18)="BL4"+PT$+N$(6)  ^0.6
320 N$(19)=PT$+N$(8)         ^0.8
330 GOSUB1900  ^ INTRODUCTION
```

```

340 REM *****
350 REM ** MAIN LOOP **
360 REM *****
370 GOSUB970 'SET RANGES
380 GOSUB620 'DRAW SCALE
390 GOSUB1140 'DRAW POINTER
400 GOSUB1360 'USER ROUTINE
410 GOT0370
420 REM *****
430 REM ** DRAW METER **
440 REM *****
450 PMODE4,5:COLOR0,5:PCLS
460 LINE(0,180)-(256,180),PSET
470 CIRCLE(128,180),12,,.8,.5,1
480 PAINT(128,175)
490 CIRCLE(128,180),150,,1,.5,1
500 CIRCLE(128,180),180,,1,.5,1
510 LINE(0,108)-(116,180),PSET
520 LINE(255,108)-(140,180),PSET
530 PAINT(1,179):PAINT(254,179)
540 PAINT(1,1):PAINT(255,1)
550 A1$= "BM190,190R4U3L4U3R4BM+8,+6L4U6R4BD6"
560 A2$= "BR3U6R4D3L4BR3D1R1D2BR8L4U6R4BM-1,+3L3BR
9R3BM-3,+2R3"
570 DRAW A1$+N$(0)+A2$
580 RETURN
590 REM *****
600 REM ** DRAW SCALE **
610 REM *****
620 PMODE4,1:COLOR0,5:PCLS
630 FORI=1TO4
640 PCOPY I+4 TO I
650 NEXT
660 SCREEN1,1
670 FOR S=0 TO 5
680 A=PI*.75-((PI/2)/5)*S
690 R=150
700 X1=128+R*COS(A)
710 Y1=180-R*SIN(A)
720 R=158
730 X2=128+R*COS(A)
740 Y2=180-R*SIN(A)
750 LINE(X1,Y1)-(X2,Y2),PSET
760 REM *****PRINT NUMBERS*****
770 SU=S*SF/5
780 IF S<3 THEN DRAW"BL4"
790 IF SU<1 AND SU>0 THEN SU=SU*5+15
800 DRAW "BU3BL4"+N$(SU)
810 NEXT
820 REM *****DRAW SUB-DIVNS*****
830 FOR S=0 TO 5*SD
840 A=PI/4+((PI/2)/(5*SD))*S

```

```

850 R=150
860 X1=128+R*COS(A)
870 Y1=180-R*SIN(A)
880 R=154
890 X2=128+R*COS(A)
900 Y2=180-R*SIN(A)
910 LINE(X1,Y1)-(X2,Y2),PSET
920 NEXT
930 RETURN
940 REM *****
950 REM ** FSD AND SUB-DIVNS **
960 REM *****
970 FOR CHOICE=0 TO 3
980 FOR SIZE=0 TO 3
990 READ RANGE(CHOICE,SIZE)
1000 NEXT SIZE,CHOICE
1010 DATA 1,2,5,10
1020 DATA 5,5,5,10
1030 DATA 10,2,5,10
1040 DATA 15,3,6,6
1050 N=RND(4)-1:M=RND(3)
1060 SF=RANGE(N,0) 'SELECT FSD
1070 SD=RANGE(N,M) 'SUB-DIVN.
1080 RESTORE
1090 RETURN
1100 REM *****
1110 REM ** DRAW POINTER **
1120 REM *****
1130 REM EASY START
1140 IF SCORE<6 AND SD=3 THEN1170
1150 IF SCORE<6 AND SD=5 THENSD=1
1160 IF SCORE<6 AND SD=10THENS=2
1170 IF SCORE<3 THEN SD=1
1180 DRAW"BM240,190"+N$(SCORE)
1190 INC=(PI/2)/(5*SD)
1200 DEV=RND(5*SD)*INC ' DEVIATN
1210 A=PI*.75-DEV
1220 X1=128+15*COS(A)
1230 Y1=180-15*SIN(A)
1240 R=150
1250 X2=128+R*COS(A)
1260 Y2=180-R*SIN(A)
1270 LINE(124,180)-(X2,Y2),PSET
1280 LINE(132,180)-(X2,Y2),PSET
1290 PAINT(X1,Y1)
1300 FL$=INKEY$
1310 KB$=INKEY$:IFKB$=""THEN1310
1320 RETURN
1330 REM *****
1340 REM ** USER **
1350 REM *****
1360 EFFORT=0

```



```

1370 CLS:SCREEN0
1380 PRINT@453,"YOUR SCORE IS :- ";
1390 PRINT@470,SCORE;
1400 A=(DEV/(PI/2))*SF 'ANSWER
1410 TRY=TRY+1
1420 EFFORT=EFFORT+1
1430 IF EFFORT<4 THEN 1520
1440 REM *****HAD 3 ATTEMPTS*****
1450 PRINT@164,"THE CORRECT ANSWER IS ";A;
1460 FORD=1T03000:NEXT
1470 REM *****SHOW THE METER*****
1480 SCREEN1
1490 FORD=1T08000:NEXT
1500 RETURN
1510 REM ***ASK THE QUESTION***
1520 PRINT@37,"WHAT IS THE READING";
1530 INPUT B
1540 IF B>SF+1 THEN 1370
1550 BI=INT(B*100+.5)
1560 AI=INT(A*100+.5)
1570 IF (AI-BI)^2<3 THEN 1700
1580 REM *****WRONG ANSWER*****
1590 PRINT@135,"SORRY ";B;" IS WRONG";
1600 IF EFFORT<3 THEN 1620
1610 FORD=1T03000:NEXT:GOTO1370
1620 PRINT@258,"PRESS ANY KEY TO LOOK AGAIN";
1630 FORD=1T0 2000
1640 KB$=INKEY$:IFKB$=""THEN1680
1650 SCREEN1
1660 KB$=INKEY$:IFKB$=""THEN1660
1670 GOTO1370
1680 NEXT
1690 GOTO1370
1700 REM *****RIGHT ANSWER*****
1710 PRINT@232,B;" IS RIGHT !";
1720 SCORE=SCORE+1
1730 PRINT@470,SCORE;
1740 FORD=1T03000:NEXT
1750 IF SCORE<12 THEN RETURN
1760 REM*****FINISHED?*****
1770 CLS6
1780 PRINT@163,"YOU HAVE SCORED ";SCORE;"POINTS";
1790 PRINT@232,"FROM ";TRY;" TRIES";
1800 PRINT@355,"DO YOU WANT TO START AGAIN?";
1810 KB$=INKEY$:IFKB$=""THEN1810
1820 IFKB$<>"Y"ANDKB$<>"N"THEN1810
1830 IFKB$="N" THEN CLS:PRINT@235,"FAREWELL":END
1840 SCORE=0:TRY=0
1850 RETURN

```

```

1860 REM *****
1870 REM **      INTRODUCTION      **
1880 REM *****
1890 REM *****TITLE*****
1900 CLS0
1910 PRINT@232,"the";CHR$(128);"meter";CHR$(128);"
reader";
1920 FORD=1TO3000:NEXT
1930 FORI=0 TO 15
1940 PRINT@263+I,CHR$(128);CHR$(126);
1950 FORD=1TO30:NEXT
1960 SOUND200,2
1970 NEXT
1980 PRINT@279,CHR$(128);
1990 FORD=1TO1000:NEXT
2000 REM *****INSTRUCTIONS*****
2010 CLS6
2020 PRINT@32," THE COMPUTER WILL DRAW THE FACE";
2030 PRINT"OF AN ELECTRIC METER WITH THE"
2040 PRINT"POINTER AT A RANDOM POSITION."
2050 PRINT@160," THE FULL SCALE DEFLECTION WILL"
2060 PRINT"BE      1  5  10 OR 15      UNITS"
2070 PRINT"DIVIDED INTO FIVE SECTIONS. EACH";
2080 PRINT"SECTION WILL BE DIVIDED INTO"
2090 PRINT" 2  3  5  6 OR 10 SUB-DIVISIONS";
2100 PRINT@352,"YOU HAVE THREE CHANCES TO READ"
2110 PRINT"THE METER. IF UNSUCCESSFUL, YOU"
2120 PRINT"WILL BE TOLD THE CORRECT READING";
2130 GOSUB450      'DRAW METER
2140 PRINT@482," PRESS ANY KEY TO CONTINUE  ";
2150 FL$=INKEY$
2160 KB$=INKEY$:IFKB$=""THEN2160
2170 CLS6
2180 PRINT@33," AFTER TWELVE CORRECT ANSWERS ";
2190 PRINT@65," YOU WILL BE TOLD HOW WELL ";
2200 PRINT@97," YOU HAVE DONE.              ";
2210 PRINT@161," WHEN YOU ARE READY TO GIVE ";
2220 PRINT@193,"YOUR READING,PRESS ANY KEY TO ";
2230 PRINT@225,"MOVE FROM THE METER DISPLAY. ";
2240 PRINT@482," PRESS ANY KEY TO CONTINUE  ";
2250 FL$=INKEY$
2260 KB$=INKEY$:IFKB$=""THEN2260
2270 RETURN

```

17. Wordsquares

General Description

This is the traditional wordsquare puzzle played on the computer. You are given half a dozen topics whose words you may wish to have hidden in the wordsquare. You select the topic and wait while the computer generates the wordsquare. When the computer is ready it displays the wordsquare and you have to move a cursor with the four cursor control keys around the screen picking out the letters. When you think that you have found a letter you may use the space bar to change its colour. You will notice that the letter is 'docked' from the word displayed on the right which you had to find. Of course it is possible to cheat and uncover every letter, in which case you will inevitably score the points for the letters found, but I am sure none of you would be that unsporting. You will notice that the final score is given in terms of letters found rather than words.

Detailed Description

Lines 110–280 Display title page, and get key press.

290 (and 690) Test for 'quit'.

310–600 Initialise arrays. X and Y give direction of word. T\$ gives six topics. HW\$ gives 6×10 hidden words.

610–720 Player selects topic from list.

790–810 Initialise arrays of letters. NL gives number of letters to be found.

840–1220 Routine to place the ten words for the chosen topic, in a 15×15 square using one of eight random directions with the word starting at a random point within the square.

1240–1310 Displays the wordsquare, filling gaps with random letters.

1330–1350 Prints list of words to be found.

1370–1390 Places cursor in random position.

- 1410-1470** Flashes cursor.
- 1510-1560** Checks players response.
- 1580-1610** Checks for valid movement.
- 1630-1640** Stops cursor when player selects spacebar to register a 'hit'.
- 1650-1710** Checks for a valid hit. If valid blanks out letter(s) from word list.
- 1740-1830** Congratulate player on successful game.
- 1860-1890** Show score to player who quit.
- 1900-1980** End of game.

Educational Notes

The educational value of wordsquare hunts is dubious. It familiarises youngsters with the words in the subject under study but that is probably the limit. I think that this program is more for use at home for general educative entertainment. You are free, of course, to change the data.

Program Listing

```

30 REM*****
40 REM*      WORDSQUARES      *
50 REM*      *                *
60 REM* WRITTEN FOR THE DRAGON *
70 REM*      *                *
80 REM*BY ALUN HUTCHINSON  1983*
90 REM*****
100 REM
110 REM*** DISPLAY TITLE PAGE
120 CLS
130 FORI=0T031:POKE1024+I,42:NEXT
140 FORI=32T0480STEP32:POKE1024+I,42:NEXT
150 FORI=0T032:POKE1504+I,42:NEXT
160 PLAY"T1503CDEFGFEDEFE"
170 PRINT@74,"wordsquares"
180 PRINT@195,"USE ARROWS TO MOVE TO THE"
190 PRINT@227,"LETTER YOU WANT AND PRESS"
200 PRINT@259,"SPACEBAR IF THE LETTER IS"
210 PRINT@291,"THE ONE THAT YOU WANT TO"
220 PRINT@323,"MAKE A WORD WITH."
230 PRINT@385,"PRESS 'Q' TO QUIT AT ANY TIME"
240 PRINT@453,"PRESS ANY KEY TO BEGIN"

```



```

250 FORI=31TO479STEP32:POKE1024+I,42:NEXT
260 A$=INKEY$
270 IFA$=""THEN260
280 PLAY"L1204F"
290 IFA$="Q"THEN1930
300 CLS
310 REM*** INITIALISE ARRAYS
320 CLEAR1000
330 DIMC$(16,16),HW$(6,10)
340 REM*** X,Y = DIRECTION
350 FORI=1TO8:READX(I):NEXT
360 DATA1,1,0,-1,-1,-1,0,1
370 FORI=1TO8:READY(I):NEXT
380 DATA0,1,1,1,0,-1,-1,-1
390 REM*** T$ = TOPICS
400 FORI=1TO6:READT$(I):NEXT
410 DATAMUSIC,PHYSICS,ELECTRONICS,COMPUTER STUDIES
,MATHS,SPORT
420 REM*** HW$ = HIDDEN WORDS
430 FORI=1TO6
440 FORJ=1TO10
450 READHW$(I,J)
460 NL(I)=NL(I)+LEN(HW$(I,J))
470 NEXTJ
480 NEXTI
490 DATABEETHOVEN,STRAVINSKY,CLARINET,TROMBONE,REC
ORDER
500 DATATCHAIKOVSKY,CONCERTO,ORCHESTRA,SYMPHONY,PR
OKOFIEV
510 DATAELECTRICITY,NEUTRON,MECHANICS,ELASTICITY,M
OMENTS
520 DATAREFRACTION,INTERFERENCE,CALORIMETER,PENDUL
UM,ULTRASONICS
530 DATATRANSISTOR,INTEGRATED,CIRCUIT,TRANSDUCER,O
SCILLOSCOPE
540 DATAFEEDBACK,CAPACITOR,RESISTOR,CATHODE,SEMICO
NDUCTOR
550 DATAMICROCOMPUTER,ASSEMBLER,COMPILER,INTERPRET
ER,DISKETTES
560 DATAPROCESSOR,MICROCHIP,KEYBOARD,MAINFRAME,PRO
GRAMMER
570 DATATRIGONOMETRY,SUBTRACTION,ADDITION,PERCENTA
GE,COSINE
580 DATACALCULUS,GEOMETRY,ARITHMETIC,INTEGRATION,M
ATRIX
590 DATAFOOTBALL,STEEPLECHASE,HURDLE,BADMINTON,GYM
NASTICS
600 DATASWIMMING,OLYMPICS,BILLIARDS,BOBSLEIGH,CRIC
KET
610 REM*** DISPLAY TOPICS AND
620 REM*** INPUT SELECTION
630 PRINT@132,"WHICH TOPIC DO YOU WANT?"

```

```

640 FOR I=1TO6
650 PRINT@198+32*I,I;T$(I)
660 NEXT
670 A$=INKEY$
680 IFA$=""THEN670
690 IFA$="Q"THEN1930
700 A=VAL(A$)
710 IFA<10RA>6THEN670
720 PLAY"L803F"
730 REM*** NL = NO. OF LETTERS
740 REM*** V = COUNTDOWN
750 REM*** S&T GIVE POSITION OF
760 REM*** LETTER IN WORD LIST
770 REM*** X2&Y2 GIVE POSITION
780 REM*** OF LETTER IN GRID
790 NL=NL(A)
800 V=NL
810 DIMS(NL),T(NL),X2(NL),Y2(NL)
820 CLS
830 PRINT@230,"thinking of puzzle"
840 REM*** ROUTINE TO PLACE 10
850 REM*** WORDS IN 15X15 GRID
860 FOR N=1TO10
870 REM*** W$ = WORD TO PLACE
880 W$=HW$(A,N)
890 REM*** X,Y(P) = DIRECTION
900 P=RND(8)
910 Y=Y(P):X=X(P)
920 REM*** X4,Y4 = START POINT
930 X4=RND(15):Y4=RND(15)
940 U$=C$(X4,Y4)
950 IF U$="" THEN 1010
960 FORI=1TOLEN(W$)
970 IF MID$(W$,I,1)=U$ THEN X4=X4-((I-1)*X):Y4=Y4-
((I-1)*Y):GOTO1000
980 NEXTI
990 GOTO900
1000 IF X4<1 OR X4>15 OR Y4<1 OR Y4>15 THEN 900
1010 X1=X4:Y1=Y4
1020 FOR W=1TOLEN(W$)
1030 IF C$(X1,Y1)<>" " AND C$(X1,Y1)<>MID$(W$,W,1)
THEN 900
1040 X1=X1+X:Y1=Y1+Y
1050 IF X1<1 OR X1>15 OR Y1<1 OR Y1>15 THEN 900
1060 NEXTW
1070 REM*** WORD PLACED O.K.
1080 X1=X4:Y1=Y4
1090 PLAY"L1002A"
1100 N$=N$+STR$(N)
1110 PRINT@389,N$
1120 FOR K=1TOLEN(W$)
1130 C$(X1,Y1)=MID$(W$,K,1)

```

```

1140 REM*** RECORD LETTER POSN.
1150 E=E+1
1160 Y2(E)=Y1-1
1170 X2(E)=X1-1
1180 X1=X1+X:Y1=Y1+Y
1190 T(E)=N
1200 S(E)=K
1210 NEXTK
1220 NEXTN
1230 REM*** FILL UP WORDSQUARE
1240 CLS
1250 FOR X=1TO15
1260 FOR Y=1TO15
1270 IF C$(X,Y)<>" " THEN 1290
1280 C$(X,Y)=CHR$(RND(26)+64)
1290 PRINT@ (32*(X-1))+Y,C$(X,Y)
1300 NEXTY
1310 NEXTX
1320 REM*** PRINT WORD LIST
1330 FOR I=1TO10
1340 PRINT@32*I+18,HW$(A,I)
1350 NEXT
1360 REM*** SP = SCREEN POSITION
1370 X=RND(15)-1:Y=RND(15)-1
1380 SP=1025+32*X+Y
1390 O=PEEK(SP)
1400 IF O<64 THEN 1460
1410 REM*** FLASH SCREEN POSN.
1420 POKESP,O-64
1430 FOR I=1TO200:NEXT
1440 POKESP,O
1450 FOR I=1TO200:NEXT
1460 A$=INKEY$
1470 IFA$="" THEN 1400
1480 REM*** "Q" = QUIT
1490 REM*** " " = HIT
1500 REM*** CHR$(?) = MOVEMENT
1510 IFA$="Q" THEN 1850
1520 IFA$=" " THEN 1630
1530 IFA$=CHR$(8) THEN Y=Y-1
1540 IFA$=CHR$(9) THEN Y=Y+1
1550 IFA$=CHR$(10) THEN X=X+1
1560 IFA$=CHR$(94) THEN X=X-1
1570 REM*** CHECK POSN. ON GRID
1580 IF X<0 THEN X=0
1590 IF Y<0 THEN Y=0
1600 IF X>14 THEN X=14
1610 IF Y>14 THEN Y=14
1620 GOTO 1380
1630 IF O<64 THEN 1650
1640 POKESP,O-64
1650 S=0

```

```

1660 REM*** CHECK FOR VALID HIT
1670 FORQ=1TONL
1680 IFY2(Q)=Y AND X2(Q)=X THENV=V-1:Y2(Q)=-1:POKE
1041+32*T(Q)+S(Q),96:S=1:PLAY"L1604C"
1690 NEXTQ
1700 REM*** S=0 = MISS:S=1 = HIT
1710 IFS=0THENPLAY"L801D":POKESP,0:GOTO1380
1720 REM*** V=0 => FINISHED
1730 IFV>0THEN1380
1740 REM*** GAME SUCCESSFUL
1750 PLAY"T1003ABCABCDABCDE"
1760 FORI=1TO2000:NEXT
1770 CLS:PRINT@200,"congratulations"
1780 SCREEN0,1
1790 FORI=1TO2000:NEXT
1800 REM*** SHOW NO. OF LETTERS
1810 PRINT@420,"YOU FOUND ALL";NL
1820 PRINT@437,"LETTERS"
1830 SCREEN0,1
1840 GOTO1910
1850 CLS
1860 REM*** SHOW SCORE AFTER "Q"
1870 PLAY"T803GFEDGF"
1880 PRINT@192,"YOU FOUND";NL-V
1890 PRINT@205,"LETTERS OUT OF";NL
1900 REM*** END OF GAME
1910 FORI=1TO3000:NEXT
1920 PLAY"T803ACABCDABCDE"
1930 CLS
1940 PRINT@203,"BYE BYE"
1950 SCREEN0,1
1960 FORI=1TO2000:NEXT
1970 PLAY"T402BC"
1980 CLS

```


18. Simon

General Description

This program is a computer version of the SIMON memory game available in High Street shops. I have included it because I like the way the programmer has handled the sound and display and I believe there is a place in any suite of programs for a memory tester such as this.

When you run the program you will see four coloured bars grouped in a square shape. There are numbers next to each bar (1–4), and these are the keys that control any bar – key 1 controls bar 1.

If you are not familiar with SIMON style games, then read on. The computer makes a sound and flashes a bar. You must then repeat exactly what the computer generated by pressing the key corresponding to the bar that flashed. To begin with the sounds and flashes are simple but quickly become more difficult.

The computer will tell you how many goes you survived.

Detailed Description

Lines 110–190 Display title page.

200–240 Initialise arrays. T() gives number of notes to be remembered; S() gives time delay in playing the notes.

280–360 Player selects level of play from 1 (easiest) to 5 (hardest).

420–520 Sets up display on screen.

550–630 Demonstrates the sound and flashing for all four colours in turn.

650 Sets delay for the chosen level.

680–720 Start of game.

740–770 Computer generates note and adds to its string.

800–870 Play computer's string.

890–1040 Player enters answer. Each note is played and the

appropriate bar flashed. After each note the answer is compared with the computer's string.

1060–1330 Routines for playing and flashing the chosen notes.

1340–1420 Congratulate successful player.

1430–1530 Shows score to unsuccessful player.

1560–1610 Invite player to play another game.

1620–1680 End of program.

Educational Notes

Young children in particular need assistance in developing retentive facilities and I think SIMON style games manage to do this. It avoids cramming useless facts into the mind, merely developing the ability to remember. I have found that first years enjoy it as an occasional end of lesson filler. It can hold the attention of a small group of youngsters for about 10 minutes.

Program Listing

```
30 REM*****
40 REM*      SIMON      *
50 REM*      *
60 REM*WRITTEN FOR THE DRAGON *
70 REM*      *
80 REM*BY ALUN HUTCHINSON 1983*
90 REM*****
100 REM
110 REM*** TITLE PAGE
120 CLS
130 SOUND200,2
140 PRINT@104,"*****"
150 PRINT@136,"*      SIMON      *"
160 PRINT@168,"*****"
170 FORI=1TO1000:NEXT
180 PLAY"T2005E"
190 PRINT@452,"PRESS ANY KEY TO BEGIN"
200 REM***INITIALISE ARRAYS
210 REM*** T=MAX NO. OF NOTES
220 REM*** S=TIME BETWEEN NOTES
230 FORI=1TO5:T(I)=5+5*I:NEXT
240 S(1)=800:S(2)=500:S(3)=300:S(4)=150:S(5)=50
250 A$=INKEY$
260 IFA$=""THEN250
270 CLS
280 PRINT@102,"SELECT LEVEL OF PLAY"
```

```

290 PRINT@141,"(1-5) "
300 A$=INKEY$
310 IFA$="" THEN300
320 REM*** A=LEVEL SELECTED
330 A=VAL(A$)
340 IFA<10RA>5 THEN300
350 PRINT@300,"LEVEL";A
360 SOUND160,1
370 FORI=1TO1000:NEXT
380 PRINT@452,"PRESS ANY KEY TO START"
390 SOUND100,1
400 A$=INKEY$
410 IFA$="" THEN400
420 REM*** SET UP DISPLAY
430 PMODE1:CLS8
440 SCREEN0,1
450 POKE1072,113
460 FORX=22TO42:SET(X,7,3):NEXT
470 POKE1223,114
480 FORY=8TO18:SET(21,Y,2):NEXT
490 POKE1424,115
500 FORX=22TO42:SET(X,20,1):NEXT
510 POKE1242,116
520 FORY=8TO18:SET(44,Y,4):NEXT
530 REM*** DEMONSTRATE THE FOUR
540 REM*** NOTES AND COLOURS
550 POKE1024,68
560 POKE1025,69
570 POKE1026,77
580 POKE1027,79
590 S=1000
600 GOSUB1070
610 GOSUB1140
620 GOSUB1210
630 GOSUB1280
640 REM*** SET S=LEVEL A DELAY
650 S=S(A)
660 REM*** ST$=COMPUTER SEQUENCE
670 REM*** TR=NO. OF TRIES
680 ST$="":TR=0
690 POKE1024,71
700 POKE1025,65
710 POKE1026,77
720 POKE1027,69
730 REM*** START OF LOOP
740 FORI=1TO2000:NEXT
750 TR=TR+1
760 R=RND(4)
770 ST$=ST$+STR$(R)
780 REM*** TEST FOR SUCCESS
790 IFTR>T(A) THEN1350
800 REM* PLAY COMPUTER SEQUENCE

```

```

810 FORK=1TOLEN(ST$)
820 R=VAL(MID$(ST$,K,1))
830 IFR=1GOSUB1070
840 IFR=2GOSUB1140
850 IFR=3GOSUB1210
860 IFR=4GOSUB1280
870 NEXTK
880 REM*** GU$=PLAYER SEQUENCE
890 GU$=""
900 REM*** PLAYER ENTERS NOTES
910 FORL=1TOTR
920 A$=INKEY$
930 IFA$<"1"ORA$>"4"THEN920
940 E=VAL(A$)
950 REM*** PLAY ENTERED NOTE
960 IFE=1GOSUB1070
970 IFE=2GOSUB1140
980 IFE=3GOSUB1210
990 IFE=4GOSUB1280
1000 GU$=GU$+STR$(E)
1010 REM*** TEST IF PLAYER SEQN.
1020 REM*** MATCHES COMPUTER SEQN
1030 IFGU$<>LEFT$(ST$,LEN(GU$))THEN1440
1040 NEXTL
1050 GOTO740
1060 REM*** PLAY & DISPLAY NOTE 1
1070 FORI=1TO10:NEXTI
1080 FORX=22TO42:SET(X,7,8):NEXTX
1090 FORX=22TO42:SET(X,7,3):NEXTX
1100 PLAY"L401G"
1110 FORI=1TOS:NEXTI
1120 RETURN
1130 REM*** PLAY & DISPLAY NOTE 2
1140 FORI=1TO10:NEXTI
1150 FORY=8TO18:SET(21,Y,8):NEXT
1160 FORY=8TO18:SET(21,Y,2):NEXT
1170 PLAY"L402G"
1180 FORI=1TOS:NEXTI
1190 RETURN
1200 REM*** PLAY & DISPLAY NOTE 3
1210 FORI=1TO10:NEXTI
1220 FORX=22TO42:SET(X,20,8):NEXT
1230 FORX=22TO42:SET(X,20,1):NEXT
1240 PLAY"L403G"
1250 FORI=1TOS:NEXT
1260 RETURN
1270 REM*** PLAY & DISPLAY NOTE 4
1280 FORI=1TO10:NEXT
1290 FORY=8TO18:SET(44,Y,8):NEXT
1300 FORY=8TO18:SET(44,Y,4):NEXT
1310 PLAY"L404G"
1320 FORI=1TOS:NEXT

```



```

1330 RETURN
1340 REM*** PLAYER SUCCESSFUL
1350 CLS
1360 PLAY"T1504ABCDEABCDEABCDE"
1370 PRINT@136,"CONGRATULATIONS"
1380 SCREEN0,1
1390 FORI=1TO2000:NEXT
1400 PRINT@324,"YOU BEAT ME AT LEVEL";A
1410 FORI=1TO3000:NEXT
1420 GOTO1550
1430 REM*** PLAYER UNSUCCESSFUL
1440 CLS
1450 SOUND55,4:SOUND40,5
1460 PRINT@105,"HARD LINES!"
1470 FORI=1TO2000:NEXT
1480 SOUND166,1
1490 TR=TR-1
1500 PRINT@293,"YOU MANAGED";TR
1510 IFTR=1THENPRINT@307,"TRY":GOTO1540
1520 IFTR>9THENPRINT@308,"TRIES":GOTO1540
1530 PRINT@307,"TRIES"
1540 FORI=1TO2000:NEXT
1550 SOUND150,2
1560 PRINT@418,"DO YOU WANT ANOTHER GAME?"
1570 PRINT@458,"(YES/NO)"
1580 A$=INKEY$
1590 IFA$="N"THEN1630
1600 IFA$="Y"THEN270
1610 GOTO1580
1620 REM*** END OF PROGRAM
1630 CLS
1640 PRINT@171,"BYE BYE"
1650 PLAY"T1004ACEGFEDCBA"
1660 FORI=1TO2000:NEXT
1670 CLS
1680 END

```

19. Hexadecimal-Binary

General Description

This program is for computer scientists only in that it is a time-based tester for binary to hexadecimal and viceversa. Hexadecimal as such is on very few of the syllabuses for 'O' level and even at 'A' level one could avoid learning it. Nevertheless for machine code work its knowledge is essential. There are calculators today that do decimal/hex conversions but few that do binary/hex conversions.

For the uninitiated this is how to do a conversion:

Consider the following binary number: 1100

Decimal for each binary position: 8421

This means the binary number is made up of $1 * 8$ and $1 * 4$, which is 12.

However hexadecimal digits are as follows (the decimal equivalents are underneath):

1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

This means that '1 1 0 0' can be written as 'C' in hexadecimal counting.

As hexadecimal counting is to base 16, the decimal number 28 is '1 C' in hex. Confused . . . ??? Of course not. Actually, like all counting systems it's easy when you use it regularly, and like all counting systems there are short cuts. If you use the program, you may need to find them to beat the clock!

Detailed Description

Lines 110–180 Display title page.

190–260 Initialise arrays of hexadecimal and binary number.

290–370 Choose hexadecimal to binary or binary to hexadecimal conversion.

450–510 Set constants for hexadecimal to binary.

520–560 Set constants for binary to hexadecimal.
570–700 Start game; initialise counts and begin timer.
710–950 Main loop of program. Generates a target (avoiding repeats); asks for an answer; validates answer and checks for correctness.
960–990 Pause after ten answers.
1000–1020 Check against previous best score.
1030–1060 Display current best score.
1080–1130 Invite player to play again.
1140–1190 Right answer subroutine.
1200–1250 Wrong answer subroutine.
1260–1320 End of program.
1330–1370 Time display subroutine.

Educational Notes

This program cannot really be considered ‘educational’ except in the practice it gives against the clock. However its graphics are fun, the clock nicely displayed and it exercises my dying grey matter as it will yours.

Program Listing

```

0 REM*****
40 REM*  HEXADECIMAL/BINARY  *
50 REM*                               *
60 REM*WRITTEN FOR THE DRAGON *
70 REM*                               *
80 REM*BY ALUN HUTCHINSON 1983*
90 REM*****
100 REM
110 REM***  TITLE PAGE
120 CLS
130 SOUND200,2
140 PRINT@100,"*****"
150 PRINT@132,"*  HEXADECIMAL-BINARY  *"
160 PRINT@164,"*****"
170 FORI=1TO1000:NEXT
180 PRINT@452,"PRESS ANY KEY TO BEGIN"
190 REM***INITIALISE ARRAYS
200 DIMH$(16),B$(16),T$(16),A$(16)
210 FORI=1TO16:READH$(I):NEXTI
220 DATA0,1,2,3,4,5,6,7
230 DATAB,9,A,B,C,D,E,F
  
```

```

240 FORI=1TO16:READB$(I):NEXTI
250 DATA0000,0001,0010,0011,0100,0101,0110,0111
260 DATA1000,1001,1010,1011,1100,1101,1110,1111
270 A$=INKEY$
280 IFA$=""THEN270
290 REM*** CHOICE OF DIRECTION
300 CLS
310 SOUND200,2
320 PRINT@99,"ENTER '1' TO CONVERT FROM"
330 PRINT@131,"HEXADECIMAL TO BINARY."
340 PRINT@195,"ENTER '2' TO CONVERT FROM"
350 PRINT@227,"BINARY TO HEXADECIMAL."
360 A$=INKEY$
370 IFA$=""THEN360
380 REM*** A=CONVERSION SELECTED
390 A=VAL(A$)
400 IFA<10RA>2THEN360
410 SOUND222,2
420 IFA=2THEN540
430 REM*** T$(I)=HEX VALUE
440 REM*** A$(I)=BINARY ANSWER
450 PRINT@326,"hexadecimal-binary"
460 FORI=1TO16:T$(I)=H$(I):A$(I)=B$(I):NEXTI
470 REM*** L$=LOWEST ACCEPTABLE
480 REM*** H$=HIGHEST ACCEPTABLE
490 REM*** D=NUMBER OF DIGITS
500 W$="BINARY":L$="0":H$="1":D=4
510 GOTO570
520 REM*** T$(I)=BINARY VALUE
530 REM*** A$(I)=HEX ANSWER
540 PRINT@326,"binary-hexadecimal"
550 FORI=1TO16:T$(I)=B$(I):A$(I)=H$(I):NEXTI
560 W$="HEXADECIMAL":L$="0":H$="F":D=1
570 FORI=1TO1000:NEXT
580 PRINT@452,"PRESS ANY KEY TO START"
590 A$=INKEY$
600 IFA$=""THEN590
610 SOUND150,1
620 REM*** PLAY THE GAME
630 CLS
640 REM*** R=RIGHT;W=WRONG
650 W=0:R=0
660 REM*** V=LAST QUESTION
670 V=0
680 REM*** SET & START TIMING
690 TIMER=0
700 START=TIMER
710 REM*** START LOOP
720 FORZ=1TO10
730 REM*** GENERATE TARGET
740 C=RND(16)
750 REM*** AVOID DUPLICATES

```



```

760 IFC=V THEN740
770 PRINT@105,"TRY NO.";Z
780 PRINT@164,"WHAT IS THE ";W$
790 PRINT@196,"EQUIVALENT OF ";T$(C)
800 E$=""
810 REM*** RECEIVE ANSWER
820 FORX=1TOD
830 GOSUB1340
840 Q$=INKEY$
850 REM*** CHECK FOR VALID KEY
860 IFQ$<L$ORQ$>H$ORQ$>"9"ANDQ$<"A"THEN830
870 E$=E$+Q$
880 REM*** DISPLAY ENTRY
890 PRINT@442,E$
900 NEXTX
910 GOSUB1340
920 REM*** CHECK FOR RIGHT/WRONG
930 IFE$=A$(C) THENGOSUB1150ELSEGOSUB1210
940 V=C
950 NEXTZ
960 REM*** PAUSE AFTER 10 TRIES
970 FORX=1T01000:NEXT
980 PLAY"T2002ABCABCABC"
990 FORX=1T02500:NEXT
1000 REM*** UPDATE RECORD SCORE
1010 IF R>RN THEN RN=R:RT=T
1020 IF R=RN AND T<RT THEN RT=T
1030 REM*** DISPLAY CURRENT BEST
1040 CLS
1050 PRINT@34,"CURRENT RECORD IS";RN;"RIGHT"
1060 PRINT@72,"IN";RT;"SECONDS"
1070 FORI=1T01000:NEXT
1080 PRINT@232,"ANOTHER GAME ?"
1090 PRINT@268,"(Y/N)"
1100 YN$=INKEY$
1110 IFYN$="Y"THEN300
1120 IFYN$="N"THEN1270
1130 GOTO1100
1140 REM*** PLAYER SUCCESSFUL
1150 R=R+1
1160 SOUND 160,3
1170 PRINT@294,"NUMBER RIGHT =" ;R
1180 GOSUB1340
1190 RETURN
1200 REM*** PLAYER UNSUCCESSFUL
1210 W=W+1
1220 SOUND55,3
1230 PRINT@358,"NUMBER WRONG =" ;W
1240 GOSUB1340
1250 RETURN
1260 REM*** END OF PROGRAM
1270 CLS

```

```

1280 PRINT@166,"THANKS FOR PLAYING"
1290 PLAY"T1004ACEGFEDCBA"
1300 FORI=1TO2000:NEXT
1310 CLS
1320 END
1330 REM*** DISPLAY TIME
1340 T=(TIMER-START)/50
1350 PRINT@17,T;
1360 PRINTTAB(24);"SECONDS"
1370 RETURN

```

20. Number Guess

General Description

There are several versions of this kind of program available. This program displays yours and the computer's responses nicely. This program is the only one in the book that actually asks your name, nice touch I think. It made me see that, given a data base of names, the computer could generate a 'name guess' as part of its interaction with the user. The computer generates some rather rude messages for 'stupid' guesses, and keeps track of the number of wasted goes.

Detailed Description

Lines 100–180 Display title page.

190–230 Player enters name.

240–350 Computer responds and explains game.

410 Computer generates the number to be guessed and sets low and high values and counters.

440–490 Player enters guess.

540–670 Routine for non-numeric guess.

700–770 Congratulate successful player.

780–870 Show number of guesses (including any wasted).

880–960 Invite player to play again.

970–1080 End of game.

1090–1240 Inform player of a wasted guess.

1250–1330 Show player result of a sensible guess.

Educational Notes

At one level this program is merely a game, but at the other it is useful for teaching a binary search. As such it is quite a valuable aid for teaching fifth and sixth year computer studies youngsters.

A binary search presupposes data that is kept in some form of order, and obviously numbers 0 to 100 are kept in numerical order. If you are not familiar with the binary search, consider the digits 0–9 below. (The computer has thought of 4)

0 1 2 3 4 5 6 7 8 9

Ask 'is the sought number in the top half of the digits'?

When you know the answer to that you repeat the question for the digits within which range you know the number lies. So the number '4' lies in the bottom half of the numbers . . . ask the question again and you learn that '4' must lie between 3 and 4, and you are virtually there. For the more able you can discuss retrieval times based on the number of elements of initial data.

Program Listing

```
30 REM*****
40 REM*      NUMBER GUESS      *
50 REM*      *
60 REM* WRITTEN FOR THE DRAGON *
70 REM*      *
80 REM*BY ALUN HUTCHINSON 1983*
90 REM*****
100 REM
110 REM*** DISPLAY TITLE PAGE
120 CLS
130 PRINT@38,"*****"
140 PRINT@70,"*  NUMBER GUESS  *"
150 PRINT@102,"*****"
160 PLAY"T2004CCDEDB"
170 FORI=1TO1000:NEXT
180 PLAY"T2005F"
190 REM*** ENTER PLAYER'S NAME
200 PRINT@289,"WHAT IS YOUR NAME";:INPUTN$
210 CLS
220 FORI=1TO1000:NEXT
230 PLAY"T1504ABCDABCAB"
240 REM*** EXPLANATION OF GAME
250 PRINT@137,"HELLO ";N$
260 FORI=1TO2000:NEXT
270 PLAY"T2001C"
280 PRINT@228,"I AM THINKING OF A WHOLE"
290 PRINT@260,"NUMBER BETWEEN 0 AND 100."
300 PRINT@292,"YOU HAVE TO ARRIVE AT THE"
310 PRINT@324,"NUMBER IN AS FEW GUESSES"
320 PRINT@356,"AS POSSIBLE."
330 FOR I=1TO2000:NEXT
340 PLAY"T2002C"
```



```

350 PRINT@453,"PRESS ANY KEY TO BEGIN"
360 A$=INKEY$
370 IFA$="" THEN360
380 REM*** GENERATE NUMBER AND
390 REM*** INITIALISE COUNTERS
400 REM*** & INTERMEDIATE VALUES
410 N=RND(101)-1:A=0:B=100:T=0:W=0
420 CLS
430 PLAY"T20056"
440 REM*** PLAYER ENTERS GUESS
450 PRINT@260,"PLEASE ENTER YOUR GUESS"
460 PRINT@333,"";:INPUTG$
470 PLAY"T2005A"
480 T=T+1
490 G=VAL(G$)
500 CLS
510 FORI=1TO1000:NEXT
520 REM*** CHECK THAT A NUMBER
530 REM*** HAS BEEN ENTERED
540 IF G>0 AND G<100 THEN680
550 PLAY"T501CD"
560 W=W+1
570 PRINT@166,G$;" IS NOT A NUMBER"
580 PRINT@198,"BETWEEN 0 AND 100"
590 FORI=1TO1000:NEXT
600 PLAY"T1503GG"
610 PRINT@352,"YOU'D BETTER TRY AGAIN, ";N$
620 FORI=1TO1000:NEXT
630 PRINT@451,"PRESS ANY KEY TO CONTINUE"
640 PLAY"T20056"
650 A$=INKEY$
660 IF A$="" THEN 650
670 GOTO420
680 CLS
690 IF G<>N THEN1080
700 REM*** GUESS IS CORRECT
710 FOR X=1TO3
720 PRINT@167,"CONGRATULATIONS"
730 FORY=1TO200:NEXTY
740 CLS
750 PLAY"T2004ACEGECA"
760 FORY=1TO100:NEXTY
770 NEXTX
780 REM*** DISPLAY NUMBER OF
790 REM*** GUESSES
800 PRINT@200,"YOU TOOK";T;"TRIES"
810 FORI=1TO600:NEXT
820 REM*** DISPLAY NUMBER OF
830 REM*** WASTED GUESSES
840 REM*** (IF APPLICABLE)
850 IFW>0THENPRINT@264,"BUT YOU WASTED";W:PLAY"T50
1F"

```

```

860 FORI=1TO1000:NEXT
870 PLAY"T2005F"
880 REM*** INVITE PLAYER TO
890 REM*** PLAY AGAIN
900 PRINT@418,"ANOTHER GAME, ";N$;" "":INPUTYN$
910 IF LEFT$(YN$,1)<>"Y" THEN960
920 CLS
930 PLAY"T2004GFEDCCC"
940 FORI=1TO1000:NEXT
950 GOTO410
960 CLS
970 REM*** END OF GAME
980 REM*** TEARFUL FAREWELLS
990 FORI=1TO500:NEXT
1000 PRINT@136,"GOODBYE, ";N$
1010 PLAY"T1504ABCDEFGFGFEDCBA"
1020 FORI=1TO1000:NEXT
1030 PRINT@293,"THANK YOU FOR PLAYING"
1040 FORI=1TO3000:NEXT
1050 PLAY"T1002BC"
1060 CLS
1070 END
1080 CLS
1090 IF G>A THEN1170
1100 REM*** WASTED GUESS
1110 W=W+1
1120 PLAY"T301FC"
1130 PRINT@140,"wrong"
1140 FORI=1TO1000:NEXT
1150 PRINT@258,"I TOLD YOU IT WAS ABOVE";A
1160 GOTO620
1170 IF G<B THEN1270
1180 REM*** WASTED GUESS
1190 W=W+1
1200 PLAY"T201GD"
1210 PRINT@140,"wrong"
1220 FORI=1TO1000:NEXT
1230 PRINT@258,"I TOLD YOU IT WAS BELOW";B
1240 GOTO620
1250 REM*** TELL PLAYER RESULT
1260 REM*** OF HIS/HER GUESS
1270 IF G<N THEN1310
1280 PRINT@132,"THE NUMBER IS BELOW";G
1290 B=G
1300 GOTO620
1310 PRINT@132,"THE NUMBER IS ABOVE";G
1320 A=G
1330 GOTO620

```

21. Queue

General Description

This is the only true computer science program in the book. It simulates how a 'queue' operates in the computer's memory. To remind you: there are two commonly used short-term storage methods you use in computer memories. By far the most common is the 'stack', which acts like a pile of books. The last book placed on the pile is the first book taken off the pile. It is fairly easy to keep track of what constitutes an empty pile. The 'queue', which is like a bus queue in some respects, works on the basis that the first into the queue is the first out of the queue. Consider the queue below:

(Bus stop) 1 2 3 4 5 6 7

When the bus comes number 1 joins the bus (only one seat free!). Numbers 8 & 9 join the queue, which if the queue does NOT shuffle forward, looks like this.

(Bus stop) 2 3 4 5 6 7 8 9

Another bus comes and this time numbers 2 & 3 join the bus (not the rush hour). If the queue does not shuffle forward it will now look like this.

(Bus stop) 4 5 6 7 8 9

Effectively, as people join the queue and catch the bus, the queue moves backwards down the street. This can happen in computer memory and can be avoided by 'shuffling forward' which in a long queue is very time consuming, or by keeping a 'notice board' of where the next free piece of pavement in the queue is and who is next on the bus, even if they do not 'seem' to be nearest the bus stop. Anyway, run the program and watch. To make the simulation more realistic the queue has been limited. Obviously it is a little inconvenient if a queue can grow so large in memory that it would over-write the program servicing it.

Detailed Description

Lines 10–290 Sets up display and selects user's choice.

300–440 Updates the queue and the two pointers.

450–690 Input a 2 digit hex number to add to the queue.

700–850 Delete a number from the queue.

860–940 Erase the underflow or overflow messages after approximately seven seconds.

950–1060 Display an end of simulation message and end the program.

Educational Notes

For the good O-level candidates and A-level candidates this is a very clear graphical demonstration. It assumes a knowledge of hexadecimal, as the values to be entered into the queue should be in 'hex'. The 'output register' always contains the value of the last item taken out of the queue, even if for several goes you then add to the queue. The current position of pointers is given numerically. Ten minutes experimenting with the simulation has drummed into some of the A-level candidates what a lesson of teaching failed to do.

Program Listing

```
10 REM+++++
20 REM+      QUEUE      +
30 REM+      +
40 REM+ WRITTEN FOR THE DRAGON +
50 REM+      +
60 REM+ BY GAVIN J.CRADLE 1983 +
70 REM+++++
80 DIM ML$(8)
90 I=0
100 CLS
110 PRINT "CONTENTS OF THE OUTPUT REGISTER"
120 PRINT STRING$(32, "-");
130 PRINT @195, STRING$(25, "*");
140 PRINT @ 227, "*01*02*03*04*05*06*07*08*";
150 PRINT @ 259, STRING$(25, "*");
160 FOR X=1 TO 8:ML$(X)="00":NEXT X
170 P1=RND(8)
180 P2=P1
190 PRINT @120, "HEAD="P1
```



```

200 PRINT @152,"TAIL="P2
210 PRINT @384,"TO ADD TO THE QUEUE TYPE 'A'"
220 PRINT @416,"TO DELETE TYPE 'D'.'Q' TO QUIT."
230 GOSUB 300
240 FOR X=1 TO 200
250 NEXT X
260 AD$=INKEY$
270 IF AD$="Q" THEN GOTO 950
280 IF AD$="A" THEN GOSUB 450 ELSE IF AD$="D" THEN
  GOSUB 700
290 GOTO 230
300 REM+++++
310 REM DISPLAY UPDATED POINTERS
320 REM OR UPDATED POINTERS AND
330 REM NEW QUEUE.ALSO UPDATE
340 REM HEAD AND TAIL OF QUEUE.
350 REM+++++
360 QUEUE$="*"
370 FOR C=1 TO 8
380 QUEUE$=QUEUE$+ML$(C)+"*"
390 NEXT C
400 PRINT @291,QUEUE$;
410 PRINT @ 323,STRING$(25,"*");
420 PRINT @125,P1;
430 PRINT @157,P2;
440 RETURN
450 REM+++++
460 REM USER ENTERS THE HEX
470 REM NUMBER OF HIS/HER CHOICE
480 REM THAT WILL BE DISPLAYED.
490 REM+++++
500 IF I=0 OR I=1 OR P1<>P2 THEN GOTO 560
510 PRINT @363,"overflow";
520 FOR X=1 TO 5
530 PLAY "V31T10003CDEFGABCDEF6"
540 NEXT X
550 GOTO 870
560 I=2
570 PRINT @448,"ENTER A 2 DIGIT HEX NUMBER ";
580 ML$(P1)=""
590 FOR Y=1 TO 2
600 A$=INKEY$
610 IF A$<"0" THEN GOTO 600
620 IF A$>"9" AND A$<"A" THEN GOTO 600
630 IF A$>"F" THEN GOTO 600
640 ML$(P1)=ML$(P1)+A$
650 NEXT Y
660 PRINT @448,STRING$(30,32)
670 P1=P1+1
680 IF P1=9 THEN P1=1
690 RETURN

```

```

700 REM+++++++
710 REM USER WANTS TO DELETE
720 REM FROM QUEUE.
730 REM+++++++
740 IF I=2 OR P1<>P2 THEN GOTO 800
750 PRINT @363,"underflow";
760 FOR X=1 TO 5
770 PLAY"V31T10003CDEFGABCDEFBAB"
780 NEXT X
790 GOTO 870
800 I=1
810 PRINT @78,ML$(P2)
820 ML$(P2)="00"
830 P2=P2+1
840 IF P2=9 THEN P2=1
850 RETURN
860 REM+++++++
870 REM DELETE UNDER/OVERFLOW
880 REM MESSAGES AFTER APPROX.
890 REM SEVEN SECONDS.
900 REM+++++++
910 FOR X=1 TO 7000
920 NEXT X
930 PRINT @363,STRING$(14,32)
940 RETURN
950 REM+++++++
960 REM END OF SIMULATION.
970 REM+++++++
980 FOR X=1 TO 100
990 CLS0
1000 PRINT @231,"end"CHR$(128)"of"CHR$(128)"simula
tion";
1010 PLAY "V31T3004C"
1020 NEXT X
1030 FOR X=1 TO 1000
1040 NEXT X
1050 CLS
1060 END

```

22. Back to Front

General Description

This program is an old mathematical favourite. Once you get the 'knack' it is easy but for the first years through to the fifth years it has provided great entertainment in trying to get the knack. It is a test of logical thinking.

You are given a line of numbers, in this case digits 1–9. They can be in any order. The object is to re-order the line of numbers in as few steps as possible using the following rule:

You specify the number of digits at the left of the lines that you wish to reverse from their currently displayed order.

e.g. line of digits >>>3 2 1 5 6 8 7 9 4

If I specify three digits reversed I will end up with a line like this:

new line of digits >>>1 2 3 5 6 8 7 9 4

Then if I specify eight digits reversed I will end up with a line like this:

new line of digits >>>9 7 8 6 5 3 2 1 4

If you don't know the puzzle, can't solve it but need to solve it to save losing face in the classroom, the solution is in the educational hints.

Detailed Description

Lines 80–180 Displays title and initialises variables.

190–270 Performs subroutine to explain the puzzle, if required.

280–450 Main structure of the program. Line 430 checks the user's answer.

460–520 Subroutine to check keyboard input.

530–730 Reverses the required number of digits.

740–880 Creates the line of numbers the user has to solve. Line 810 checks that no digit is repeated.

890-1210 Instructions, giving example of play.
1220-1440 Correct answer and invitation for another game.

Educational Note

This program captivates. I have watched and listened to groups of children in heated discussion about which numbers to reverse, and it gives the lie to the theory that the computer stifles conversation. It can actively encourage discussion amongst young people. The solution by the way is, repeating from the highest digit down to the lowest digit, bring it to the left-hand end of the line before sending it into its final position.

I would suggest that you can safely use this program for 15 minutes with a group of children before they show signs of restlessness. The only problem is how to lead up in the classroom to using it, and I have tended to offer it as a lesson 'break'.

Program Listing

```
10 REM*****
20 REM*      BACK TO FRONT      *
30 REM*      *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*      *
60 REM* BY LAURENCE OWEN 1983 *
70 REM*****
80 REM *****TITLE DISPLAY*****
90 CLS2
100 FOR I=33 TO 449 STEP32
110 PRINT@I,"BACKTOFRONTTOBACKTOFRONTTOBACK";
120 NEXT I
130 FORD=1T05000:NEXT
140 A=0:G=0
150 DIM M$(2),H(9)
160 S$=""
170 A$=" 1 2 3 4 5 6 7 8 9 "
180 G$=" 4 2 6 5 1 8 9 7 3 "
190 REM *****INSTRUCTIONS ?*****
200 CLS
210 PRINT@40,"# BACK TO FRONT #";
220 SOUND 120,4:SOUND80,4
230 PRINT@98,"DO YOU KNOW THIS PUZZLE? ";
240 T1$="N":T2$="Y"
250 GOSUB490 ' GET REPLY
260 PRINT@123,KB$;
```



```

270 IFKB$="N" THEN GOSUB920 'EXPLAIN
280 GOSUB770 'CREATE NUMBER
290 CLS6
300 PRINT@229,G$;" ";
310 PRINT@261,S$;" ";
320 PRINT@426,"TRIES =" ;G;
330 PRINT@36,"HOW MANY NUMBERS DO YOU";
340 PRINT@68,"WANT TO REVERSE ? ";
350 T1$="2":T2$="9"
360 GOSUB490 'GET REPLY
370 T=ASC(KB$)-48
380 PRINT@88,KB$;
390 G=G+1
400 PRINT@433,G;
410 GOSUB560 'REVERSE
420 PRINT@88," ";
430 IF G=A$ THEN 1230
440 PLAY"T0ADA"
450 GOTO360
460 REM *****
470 REM ** REPLY SUB **
480 REM *****
490 KB$=INKEY$:IFKB$="" THEN490
500 IFKB$<T1$ OR KB$>T2$ THEN490
510 IF T1$="N" AND KB$>T1$ AND KB$<T2$ THEN490
520 RETURN
530 REM *****
540 REM **REVERSE NUMBERS SUB**
550 REM *****
560 PRINT@261,LEFT$(G$,T*2+1);
570 PRINT@229,RIGHT$(S$,T*2+1);
580 H1$=RIGHT$(G$,18-(T*2))
590 H$=" "
600 W=2
610 SOUND120,4
620 FOR X=T TO 1 STEP-1
630 H$=H$+MID$(G$,X*2+1,1)+" "
640 PRINT@261+X*2,CHR$(32);
650 FOR Y=T*2 TO W STEP-1
660 PRINT@230+Y,CHR$(32);
670 PRINT@229+Y,MID$(G$,X*2+1,1);
680 FORD=1T050:NEXT
690 NEXTY
700 W=W+2
710 NEXTX
720 G$=H$+H1$
730 RETURN
740 REM *****
750 REM ** CREATE NUMBER SUB **
760 REM *****
770 G$=" "
780 FOR X=1 TO 9

```

```

790 HH=RND(9)
800 FOR Y=1 TO 9
810 IF H(Y)=HH THEN 790
820 IF Y=X THEN 840
830 NEXT Y
840 H(X)=HH
850 G$=G$+STR$(HH)
860 NEXT X
870 G$=G$+" "
880 RETURN
890 REM *****
900 REM **      EXPLAIN SUB      **
910 REM *****
920 FORD=1TO1000:NEXT
930 CLS2
940 PRINT@34," YOU WILL SEE THE NUMBERS ";
950 PRINT@66," 1 TO 9 IN THE WRONG ORDER ";
960 PRINT@229,G$;" ";
970 PRINT@261,S$;" ";
980 FORD=1TO5000:NEXT
990 T=3+RND(5)
1000 PRINT@130,"IF YOU TYPE";T;"I'LL TURN THE ";
1010 PRINT@162,"FIRST";T;"NUMBERS - LIKE THIS ";
1020 FORD=1TO4000:NEXT
1030 GOSUB560      ' REVERSE
1040 FORD=1TO5000:NEXT
1050 PRINT@130,STRING$(28," ");
1060 PRINT@162,STRING$(28," ");
1070 T=11-T
1080 PRINT@130,"IF YOU TYPE";T;"I'LL TURN THE ";
1090 PRINT@162,"FIRST";T;"NUMBERS ROUND ..... ";
1100 FORD=1TO3000:NEXT
1110 GOSUB560      ' REVERSE
1120 FORD=1TO1000:NEXT
1130 PRINT@362," AND SO ON! ";
1140 FORD=1TO3000:NEXT
1150 CLS8
1160 PRINT@34,"YOU HAVE TO FINISH WITH ... ";
1170 PRINT@261,A$;" ";
1180 FORD=1TO3000:NEXT
1190 PRINT@429," OK ? ";
1200 FORD=1TO3000:NEXT
1210 RETURN
1220 REM *****FINISHED*****
1230 FORD=1TO3000:NEXT
1240 PLAY"BABBAGE"
1250 CLS
1260 PRINT@37,"YOU GOT THE ANSWER IN";:PRINT@75,G;
"MOVES";
1270 TU$="DEFGABAGFEDC"
1280 PLAY"T20"+TU$+TU$+TU$
1290 PRINT@330,"PLAY AGAIN ?";

```

```
1300 T1$="N":T2$="Y"  
1310 GOSUB490 ' GET REPLY  
1320 PRINTKB$;  
1330 FORD=1T0100:NEXT  
1340 IFKB$="N" THEN 1370  
1350 G=0  
1360 GOTO280  
1370 CLS  
1380 PRINT@77,"BYE !!!";  
1390 PLAY"T4ADP2AD"  
1400 END
```

23. Maths Stars

General Description

This program produces various mathematical stars within a circle, starting with three points on the circle and increasing the number to up to twenty points. It is possible to stop the demonstration by pressing SHIFT and @. The program can be restarted by pressing any key.

Detailed Description

Lines 80–160 Display title page.

190–200 Clear string space and initialise arrays.

230–430 Set up numerals as constants.

440–650 Main program loop:–

480 Display numeral for current number of points.

510–540 Calculate co-ordinates of the points on the circle.

570–610 Draw lines between points.

620 Pause.

640 Clear screen.

660–690 End of program.

Educational Note

Drawing these ‘roses’ is a favourite end of term pastime, and as such it is now possible to add a little spice to this. Many children can cope with drawing the simple roses but find the ones with several points very difficult to master. Usually their patience runs out before the pencil lead does. This will at least show them quickly what it was intended that they achieve.

Program Listing

```
10 REM*****
20 REM*      MATHS STARS      *
30 REM*                               *
40 REM* WRITTEN FOR THE DRAGON *
50 REM*                               *
60 REM*BY ALUN HUTCHINSON 1983*
70 REM*****
80 CLS
90 PRINT@102,"*****"
100 PRINT@134,"*      MATHS STARS      *"
110 PRINT@166,"*****"
120 PRINT@258,"THE DISPLAY SHOWS THE FIGURES"
130 PRINT@289,"GENERATED BY LINKING REGULARLY"
140 PRINT@325,"SPACED POINTS ON THE"
150 PRINT@355,"CIRCUMFERENCE OF A CIRCLE"
160 PRINT@452,"PRESS ANY KEY TO BEGIN"
170 A$=INKEY$
180 IFA$="" THEN 170
190 PCLEAR4: CLEAR 400
200 DIMX(20),Y(20),N$(20)
210 PMODE4,1:PCLS: SCREEN1,1
220 REM*** SET UP NUMERALS
230 N$(0)="BR7L4U6R4D6BR1"
240 N$(1)="BR5U6BM+3,+6"
250 N$(2)="BM+3,-6R4D4L4D2R4BR1"
260 N$(3)="BR3R4U6L4BM+1,+3;R3BM+1,+3"
270 N$(4)="BM+7,-2;L5U4BR4D6BR2"
280 N$(5)="BR3R4U4L4U2R4BM+1,+6"
290 N$(6)="BM+3,-6D6R4U3L4BM+5,+3"
300 N$(7)="BM+3,-5;U1R4D6BR1"
310 N$(8)="BR7L4U6R4D3L4R4D3BR1"
320 N$(9)="BR7U6L4D3R4BM+1,+3"
330 N$(10)=N$(1)+N$(0)
340 N$(11)=N$(1)+N$(1)
350 N$(12)=N$(1)+N$(2)
360 N$(13)=N$(1)+N$(3)
370 N$(14)=N$(1)+N$(4)
380 N$(15)=N$(1)+N$(5)
390 N$(16)=N$(1)+N$(6)
400 N$(17)=N$(1)+N$(7)
410 N$(18)=N$(1)+N$(8)
420 N$(19)=N$(1)+N$(9)
430 N$(20)=N$(2)+N$(0)
440 REM*** START OF LOOP
450 FORS=3 TO 20
460 REM*** DRAW NUMERAL SHOWING
470 REM*** NUMBER OF POINTS
480 DRAW"C1BM10,150"+N$(S)
```

```

490 REM*** CALCULATE POSITIONS
500 REM*** OF POINTS
510 FOR I=1TOS
520 X(I)=90*COS(ATN(1)*8*I/S)
530 Y(I)=90*SIN(ATN(1)*8*I/S)
540 NEXT I
550 REM*** DRAW LINES TO
560 REM*** CONNECT POINTS
570 FOR I=1TOS-1
580 FOR J=I+1TOS
590 LINE(X(I)+120,Y(I)+96)-(X(J)+120,Y(J)+96),PSET
600 NEXT J
610 NEXT I
620 FORK=1TO1000:NEXTK
630 REM*** CLEAR SCREEN
640 PCLS
650 NEXTS
660 REM*** END OF PROGRAM
670 FORK=1TO1000:NEXTK
680 CLS
690 END

```

24. Periodic Table

General Description

The program displays the periodic table of the elements, in various groupings, with full categorisation of the element types, which both distinguishes it from simpler and shorter programs merely listing the elements, and accounts for the length of the program generally, with the multiple data lists such a categorisation requires. The program first presents a menu to the user, to allow a choice of the main display groupings, and the two test routines included in the program. Subsidiary menus are presented in the procedures accessed from the main menu to allow further choice of groupings and periods. The presentation of the elements in tabular form makes them clear and easy to understand and remember.

Detailed Description

Lines 10–250 Main program section calling routines selected from menu display.

260–1950 These procedures initialise the program, display titles and the instructions (when requested from the menu), and provide the data for the program. Great care must be exercised to ensure the data is typed in correctly. The 999 which appears in the data statements after line 1240 acts as a data terminator, but the trailing zeroes are needed to prevent the READ statements in lines 480 to 680 failing.

1960–2510 These lines contain the menu procedure, and the text display for the routines to display elements by periodic grouping, with a subsidiary menu for choice of period. Note the keyboard validations in lines 2210–2240 and 2460–2490.

2520–3330 These lines have the procedure for display of chosen periods of elements.

3340–4400 Procedures to display particular groups of elements as chosen in lines 3860–3900.

4410–5350 This section contains the test sections of the program. The correct answers are displayed when you have finished the test. Both tests are 12 questions chosen at random.

Educational Note

These types of programs are good exercises in information retrieval for upper school youngsters. There is nothing in the program which cannot be found in a book, but the presentation in groupings and periods and the very interaction with the computer has tutorial value. The level of knowledge assumed is good O-level or A-level, and below this standard youngsters will not find it very meaningful. It provides a good quick reference for teachers, but must of course be loaded before it is needed to fulfil this function.

Program Listing

```
10  '*****
20  '*      PERIODIC TABLE      *
25  '*                                          *
30  '* WRITTEN FOR THE DRAGON    *
35  '*                                          *
40  '* BY      RAY NEW      1983  *
50  '*****
60  CLEAR 6000
70  DIM NAME$(103),SYMBOL$(103)
80  DIM GROUP(8,8),PERIOD(7,9)
90  DIM TRANS(3,11),RARE(15)
100 DIM URAN(15),TEST(20)
110 DIM ANS$(20)
120 GOSUB 290 'HEADER
130 GOSUB 450 'SETUP
140 GOSUB 1600 'INSTRUCTIONS
150 GOSUB 1990 'MENU
160 ON SELECT GOTO 170,180,190,200,210,220,230
170 GOSUB 1600:GOTO230 'INSTR
180 GOSUB 2290:GOTO230 'DISP 1
190 GOSUB 3680:GOTO230 'DISP 2
200 GOSUB 3600:GOTO230 'DISP 3
210 GOSUB 4440:GOTO230 'TEST 1
220 GOSUB 4920:GOTO230 'TEST 2
```



```

230 IF SELECT<>7 THEN 150
240 CLS
250 END
260 REM #
270 REM HEADER PAGE
280 REM #
290 CLS3
300 PRINT@172,"tHe";
310 PRINT@236,"PeRiOdIc";
320 PRINT@303,"TaBlE";
330 FOR I=22T040
340 SET(I,8,5):SET(I,21,5)
350 SOUND100+RND(50),1
360 NEXTI
370 FOR I=8T021
380 SET(22,I,5):SET(40,I,5)
390 SOUND200+RND(50),1
400 NEXTI
410 RETURN
420 REM #
430 REM SETUP
440 REM #
450 FOR I=1 TO 103
460 READ NAME$(I),SYMBOL$(I)
470 NEXT I
480 FOR I=1 TO 8
490 FOR J=1 TO 8
500 READ GROUP(I,J)
510 NEXT J
520 NEXT I
530 FOR I=1 TO 7
540 FOR J=1 TO 9
550 READ PERIOD(I,J)
560 NEXT J
570 NEXT I
580 FOR I=1 TO 3
590 FOR J=1 TO 11
600 READ TRANS(I,J)
610 NEXT J
620 NEXT I
630 FOR I=1 TO 15
640 READ RARE(I)
650 NEXT I
660 FOR I=1 TO 15
670 READ URAN(I)
680 NEXT I
690 DATA HYDROGEN,H,HELIUM,HE
700 DATA LITHIUM,LI,BERYLLIUM,BE
710 DATA BORON,B,CARBON,C
720 DATA NITROGEN,N,OXYGEN,O
730 DATA FLUORINE,F,NEON,NE
740 DATA SODIUM,NA,MAGNESIUM,MG

```

750 DATA ALUMINIUM,AL,SILICON,SI
 760 DATA PHOSPHORUS,P,SULPHUR,S
 770 DATA CHLORINE,CL,ARGON,A
 780 DATA POTASSIUM,K,CALCIUM,CA
 790 DATA SCANDIUM,SC,TITANIUM,TI
 800 DATA VANADIUM,V,CHROMIUM,CR
 810 DATA MANGANESE,MN,IRON,FE
 820 DATA COBALT,CO,NICKEL,NI
 830 DATA COPPER,CU,ZINC,ZN
 840 DATA GALLIUM,GA,GERMANIUM,GE
 850 DATA ARSENIC,AS,SELENIUM,SE
 860 DATA BROMINE,BR,KRYPTON,KR
 870 DATA RUBIDIUM,RB,STRONTIUM,SR
 880 DATA YTTRIUM,Y,ZIRCONIUM,ZR
 890 DATA NIOBIUM,NB,MOLYBDENUM,MO
 900 DATA TECHNETIUM,TC,RUTHENIUM,RU
 910 DATA RHODIUM,RH,PALLADIUM,PD
 920 DATA SILVER,AG,CADMIUM,CD
 930 DATA INDIUM,IN,TIN,SN
 940 DATA ANTIMONY,SB,TELLURIUM,TE
 950 DATA IODINE,I,XENON,XE
 960 DATA CAESIUM,CS,BARIUM,BA
 970 DATA LANTHANUM,LA,CERIUM,CE
 980 DATA PRASEODYMIUM,PR
 990 DATA NEODYMIUM,ND,PROMETHIUM,PM
 1000 DATA SAMARIUM,SM,EUROPIUM,EU
 1010 DATA GADOLINIUM,GD,TERBIUM,TB
 1020 DATA DYSPROSIUM,DY,HOLMIUM,HO
 1030 DATA ERBIUM,ER,THULIUM,TM
 1040 DATA YTTERBIUM,YB,LUTETIUM,LU
 1050 DATA HAFNIUM,HF,TANTALUM,TA
 1060 DATA TUNGSTEN,W,RHENIUM,RE
 1070 DATA OSMIUM,OS,IRIDIUM,IR
 1080 DATA PLATINUM,PT,GOLD,AU
 1090 DATA MERCURY,HG,THALLIUM,TL
 1100 DATA LEAD,PB,BISMUTH,BI
 1110 DATA POLONIUM,PO,ASTATINE,AT
 1120 DATA RADON,RN,FRANCIUM,FR,RADIUM,RA
 1130 DATA ACTINIUM,AC,THORIUM,TH
 1140 DATA PROTACTINIUM,PA,URANIUM,U
 1150 DATA NEPTUNIUM,NP,PLUTONIUM,PU
 1160 DATA AMERICIUM,AM,CURIUM,CM
 1170 DATA BERKELIUM,BK,CALIFORNIUM,CF
 1180 DATA EINSTEINIUM,ES,FERMIUM,FM
 1190 DATA MENDELEVIUM,MD,NOBELIUM,NB
 1200 DATA LAWRENCIUM,LW
 1210 REM #
 1220 REM ELEMENTS IN GROUPS
 1230 REM #
 1240 DATA 1,3,11,19,37,55,87,999
 1250 DATA 4,12,20,38,56,88,999,0
 1260 DATA 5,13,31,49,81,999,0,0

```

1270 DATA 6,14,32,50,82,999,0,0
1280 DATA 7,15,33,51,83,999,0,0
1290 DATA 8,16,34,52,84,999,0,0
1300 DATA 9,17,35,53,85,999,0,0
1310 DATA 2,10,18,36,54,86,999,0
1320 REM #
1330 REM ELEMENTS IN PERIODS
1340 REM #
1350 DATA 1,0,0,0,0,0,0,2,999
1360 DATA 3,4,5,6,7,8,9,10,999
1370 DATA 11,12,13,14,15,16,17,18,999
1380 DATA 19,20,31,32,33,34,35,36,999
1390 DATA 37,38,49,50,51,52,53,54,999
1400 DATA 55,56,81,82,83,84,85,86,999
1410 DATA 87,88,999,0,0,0,0,0,0
1420 REM #
1430 REM TRANSITIONAL ELEMENTS
1440 REM #
1450 DATA 21,22,23,24,25,26,27,28,29,30,999
1460 DATA 39,40,41,42,43,44,45,46,47,48,999
1470 DATA 72,73,74,75,76,77,78,79,80,999,0
1480 REM #
1490 REM RARE EARTH
1500 REM #
1510 DATA 57,58,59,60,61,62,63,64,65,66,67,68,69,70,71
1520 REM #
1530 REM TRANS-URANIC
1540 REM #
1550 DATA 89,90,91,92,93,94,95,96,97,98,99,100,101,102,103
1560 RETURN
1570 REM #
1580 REM INSTRUCTIONS
1590 REM #
1600 CLS
1610 GOSUB 1930 'TITLE
1620 PRINT:PRINT
1630 PRINT" THIS PROGRAM DISPLAYS THE"
1640 PRINT"PERIODIC TABLE OF ELEMENTS BY"
1650 PRINT"PERIODS AND GROUPS.IT ALSO LISTS";
1660 PRINT"THE ELEMENT NAME AND SYMBOL FOR"
1670 PRINT"EACH ELEMENT.SIMPLE TESTS ARE"
1680 PRINT"INCLUDED TO TEST YOUR KNOWLEDGE"
1690 PRINT"OF THE ELEMENTS,SYMBOLS AND THE"
1700 PRINT"PERIODIC TABLE.ALL ANSWERS ARE"
1710 PRINT"INCLUDED IN THE PROGRAM."
1720 PRINT@480," PRESS ANY KEY TO CONTINUE";
1730 DU$=INKEY$
1740 IF INKEY$=""THEN1740
1750 CLS
1760 GOSUB 1930 'TITLE

```

```

1770 PRINT:PRINT:PRINT
1780 PRINT"  EVERY ELEMENT HAS AN ATOMIC"
1790 PRINT"NUMBER STARTING WITH HYDROGEN"
1800 PRINT"WITH AN ATOMIC NUMBER OF 1."
1810 PRINT"THE HORIZONTAL ROWS ARE CALLED"
1820 PRINT"PERIODS AND THE VERTICAL COLUMNS";
1830 PRINT"ARE CALLED GROUPS."
1840 PRINT"  THERE ARE 7 PERIODS AND 8"
1850 PRINT"GROUPS."
1860 PRINT@480,"  PRESS ANY KEY TO CONTINUE";
1870 DU$=INKEY$
1880 IF INKEY$=""THEN1880
1890 RETURN
1900 REM #
1910 REM DISPLAY TITLE
1920 REM #
1930 PRINTSTRING$(9,CHR$(159))+ "periodic"+" "+"tab
le"+STRING$(9,CHR$(159));
1940 POKE1041,32
1950 RETURN
1960 REM #
1970 REM MENU
1980 REM #
1990 CLS
2000 GOSUB 1930 'TITLE
2010 Z=0
2020 PRINT@65,"no.      title";
2030 POKE1091,46
2040 PRINT@130,"1. INSTRUCTIONS";
2050 POKE1154,49:POKE1155,46
2060 PRINT@162,"2. ELEMENTS BY PERIOD";
2070 POKE1186,50:POKE1187,46
2080 PRINT@194,"3. ELEMENTS BY GROUP";
2090 POKE1218,51:POKE1219,46
2100 PRINT@226,"4. ELEMENT NAMES AND SYMBOLS";
2110 POKE1250,52:POKE1251,46
2120 PRINT@258,"5. TEST1-SYMBOLS FOR ELEMENTS";
2130 POKE1282,53:POKE1283,46
2140 PRINT@290,"6. TEST2-NAMES FOR SYMBOLS";
2150 POKE1314,54:POKE1315,46
2160 PRINT@322,"7. END";
2170 POKE1346,55:POKE1347,46
2180 PRINT@386,"  CHOOSE YOUR SELECTION FROM";
2190 PRINT@418,"THE MENU,AND TYPE THE NUMBER";
2200 PRINT@450,"OF YOUR CHOICE(1-7).";
2210 DU$=INKEY$
2220 SELECT$=INKEY$
2230 IF SELECT$<"1"OR SELECT$>"7"THEN2220
2240 SELECT=VAL(SELECT$)
2250 RETURN

```



```

2260 REM #
2270 REM DISPLAY BY PERIOD
2280 REM #
2290 CLS
2300 GOSUB 1930 'TITLE
2310 PRINT
2320 PRINT" THERE ARE SEVEN PERIODS (1-7).";
2330 PRINT"THE HIGHER THE PERIOD,THE HIGHER";
2340 PRINT"THE ATOMIC NUMBER OF ITS"
2350 PRINT"ELEMENTS.THE CHEMICAL PROPERTIES";
2360 PRINT"OF THE ELEMENTS VARY GRADUALLY"
2370 PRINT"ACROSS THE PERIOD.THE ELEMENTS"
2380 PRINT"WITH LOWER NUMBERS IN A PERIOD"
2390 PRINT"ARE METALS,AS THE ATOMIC NUMBER"
2400 PRINT"INCREASES THE ELEMENTS BECOME"
2410 PRINT"NON-METALLIC AND END WITH AN"
2420 PRINT"INERT GAS."
2430 PRINT"CHOOSE THE PERIOD YOU WANT TO"
2440 PRINT"LOOK AT.";
2450 PRINT"TYPE IN THE NUMBER (1-7)";
2460 DU$=INKEY$
2470 Z$=INKEY$
2480 IF Z$<"1"OR Z$>"7"THEN2470
2490 Z=VAL(Z$)
2500 GOSUB 2550 'DISPLAY A PERIOD
2510 RETURN
2520 REM #
2530 REM DISPLAY A PERIOD
2540 REM #
2550 CLS
2560 GOSUB 1930 'TITLE
2570 PRINT@44,"PERIOD";Z
2580 PRINT
2590 PRINT"group";
2600 POKE1126,32:POKE1125,32
2610 FOR I=0TO7:POKE1130+I*3,49+I:NEXTI
2620 PRINT@128,"at no ";
2630 POKE1155,32:POKE1154,46:POKE1158,46
2640 I=1
2650 PRINTTAB(8+(I-1)*3);
2660 IF PERIOD(Z,I)=0THEN 2680
2670 PRINT USING "####";PERIOD(Z,I);
2680 I=I+1
2690 IF I<9 AND PERIOD(Z,I)<>999 THEN 2650
2700 PRINT@160,"element";
2710 I=1
2720 PRINTTAB(8+(I-1)*3);
2730 LS=LEN(SYMBOL$(PERIOD(Z,I)))
2740 PRINTSTRING$(3-LS," ");
2750 PRINT SYMBOL$(PERIOD(Z,I));
2760 I=I+1
2770 IF I<9 AND PERIOD(Z,I)<>999 THEN 2720

```

```

2780 IF Z<4 THEN 3240
2790 IF Z=7 THEN 2940
2800 PRINTTAB(5);"transitional elements"
2810 POKE1233,32
2820 I=1
2830 PRINTTAB((I-1)*3);
2840 PRINTUSING"###";TRANS(Z-3,I);
2850 I=I+1
2860 IF I<11 AND TRANS(Z-3,I)<>999 THEN2830
2870 I=1:PRINT
2880 LS=LEN(SYMBOL$(TRANS(Z-3,I)))
2890 PRINTSTRING$(3-LS," ");
2900 PRINTSYMBOL$(TRANS(Z-3,I));
2910 I=I+1
2920 IF I<11 AND TRANS(Z-3,I)<>999 THEN 2880
2930 IF Z<6 THEN 3240
2940 IF Z=6 THEN PRINT:PRINTTAB(10);"rare earths":
POKE1326,32 ELSE PRINT:PRINTTAB(9);"trans-uranics"
:POKE1230,45
2950 I=1
2960 PRINTTAB((I-1)*3);
2970 IF Z=6THEN PRINTUSING"###";RARE(I);
2980 IF Z=7THEN PRINTUSING"###";URAN(I);
2990 I=I+1
3000 IF I<11THEN2960
3010 I=1:PRINT
3020 L1=LEN(SYMBOL$(RARE(I)))
3030 L2=LEN(SYMBOL$(URAN(I)))
3040 IF Z=6THENPRINTSTRING$(3-L1," ");SYMBOL$(RARE
(I));
3050 IF Z=7THENPRINTSTRING$(3-L2," ");SYMBOL$(URAN
(I));
3060 I=I+1
3070 IF I<11 THEN 3020
3080 I=11:PRINT
3090 PRINT
3100 PRINT TAB((I-11)*3);
3110 IF Z=6 THENPRINTUSING"###"; RARE(I);:GOTO3140
3120 IF URAN(I)>99 THENPRINT" ";
3130 IF Z=7 THENPRINTUSING"###"; URAN(I);
3140 I=I+1
3150 IF I<16 THEN 3100
3160 I=11:PRINT
3170 L1=LEN(SYMBOL$(RARE(I)))
3180 L2=LEN(SYMBOL$(URAN(I)))
3190 IF Z=6 THEN PRINTSTRING$(3-L1," ");SYMBOL$(RA
RE(I));:GOTO3220
3200 IF URAN(I)>99 THENPRINT" ";
3210 IF Z=7 THEN PRINTSTRING$(3-L2," ");SYMBOL$(UR
AN(I));
3220 I=I+1
3230 IF I<16 THEN 3170

```

```

3240 PRINT@480,"DO YOU WANT ELEMENT NAMES(Y/N)?";
3250 IK$=INKEY$
3260 IF IK$<>"Y"AND IK$<>"N" THEN 3250
3270 IF IK$="N"THEN3330
3280 TITLE$=STRING$(11," ")+ "PERIOD"+STR$(Z)
3290 IF Z<7 THEN VF=PERIOD(Z+1,1)-1
3300 IF Z=7 THEN VF=103
3310 VS=PERIOD(Z,1)
3320 GOSUB 3370 'ELEMENT LIST
3330 RETURN
3340 REM #
3350 REM ELEMENT LIST
3360 REM #
3370 CLS
3380 GOSUB 1930 'TITLE
3390 PRINT@33,TITLE$
3400 LC=3
3410 PRINTTAB(1);"at. no.";TAB(14);"name";TAB(25);
"symbol"
3420 POKE1091,46:POKE1092,32:POKE1095,46
3430 PRINTTAB(2);
3440 PRINTUSING"####";VS;
3450 PRINTTAB(10);NAMES$(VS);TAB(27);SYMBOL$(VS)
3460 LC=LC+1:VS=VS+1
3470 IF LC<15 AND VS<=VF THEN3430
3480 IF VS=VF+1 THEN 3530
3490 PRINT@480,"PRESS ANY KEY TO CONTINUE";
3500 DU$=INKEY$
3510 IF INKEY$=""THEN3510
3520 GOTO3370
3530 PRINT@480,"PRESS ANY KEY TO FINISH";
3540 DU$=INKEY$
3550 IF INKEY$=""THEN3550
3560 RETURN
3570 REM #
3580 REM DISPLAY ALL ELEMENTS
3590 REM #
3600 VS=1
3610 VF=103
3620 TITLE$="ALL ELEMENTS BY ATOMIC NUMBER"
3630 GOSUB 3370 'ELEMENT LIST
3640 RETURN
3650 REM #
3660 REM ELEMENTS BY GROUP
3670 REM #
3680 CLS
3690 GOSUB 1930 'TITLE
3700 PRINT
3710 PRINT" THERE ARE 8 GROUPS OF ELEMENTS";
3720 PRINT"(1-8),THERE ARE ALSO THE"
3730 PRINT"TRANSITION ELEMENTS,RARE EARTH"
3740 PRINT"AND TRANS-URANIC ELEMENTS."

```

```

3750 PRINT"  THESE ARE ALL DISPLAYED HERE."
3760 PRINT"TO LIST THE ELEMENTS IN THESE "
3770 PRINT"GROUPS TYPE 1 TO 8 FOR THE"
3780 PRINT"GROUPS,T FOR THE TRANSITIONAL"
3790 PRINT"ELEMENTS,R FOR THE RARE EARTHS"
3800 PRINT"AND U FOR THE TRANS-URANIC"
3810 PRINT"ELEMENTS."
3820 PRINT
3830 PRINT"  TYPE YOUR CHOICE (1-8,T,R,U)"
3840 Z$=INKEY$
3850 IF (Z$<"1"ORZ$>"8")ANDZ$<>"R"AND Z$<>"T"AND Z$
<>"U"THEN3840
3860 IF Z$="R"THEN GOSUB3950:GOTO3910
3870 IF Z$="T"THEN GOSUB4030:GOTO3910
3880 IF Z$="U"THEN GOSUB4190:GOTO3910
3890 Z=VAL(Z$)
3900 GOSUB 4270 'GROUP
3910 RETURN
3920 REM #
3930 REM RARE EARTH
3940 REM #
3950 TITLE$="          RARE EARTHS"
3960 VS=57
3970 VF=71
3980 GOSUB 3370 'ELEMENT LIST
3990 RETURN
4000 REM #
4010 REM DISPLAY TRANSITIONAL
4020 REM #
4030 TITLE$="TRANSITION ELEMENTS - PERIOD 4"
4040 VS=21
4050 VF=30
4060 GOSUB 3370 'ELEMENT LIST
4070 TITLE$="TRANSITION ELEMENTS - PERIOD 5"
4080 VS=39
4090 VF=48
4100 GOSUB 3370 'ELEMENT LIST
4110 TITLE$="TRANSITION ELEMENTS - PERIOD 6"
4120 VS=72
4130 VF=80
4140 GOSUB 3370 'ELEMENT LIST
4150 RETURN
4160 REM #
4170 REM DISPLAY TRANS-URANICS
4180 REM #
4190 TITLE$="          TRANS-URANIC ELEMENTS"
4200 VS=89
4210 VF=103
4220 GOSUB 3370 'ELEMENT LIST
4230 RETURN

```



```

4240 REM #
4250 REM DISPLAY ELEMENT IN GROUP
4260 REM #
4270 CLS
4280 GOSUB 1930 'TITLE
4290 PRINT@44,"GROUP "+Z$
4300 PRINTTAB(1);"at. no.";TAB(14);"name";TAB(25);
"symbol"
4310 POKE1091,46:POKE1092,32:POKE1095,46
4320 I=1
4330 PRINTTAB(2);
4340 PRINTUSING"###";GROUP(Z,I);
4350 PRINTTAB(10);NAMES$(GROUP(Z,I));TAB(27);SYMBOL$(GROUP(Z,I))
4360 I=I+1
4370 IF GROUP(Z,I)<999THEN4330
4380 PRINT@485,"PRESS ANY KEY TO FINISH";
4390 IF INKEY$=""THEN4390
4400 RETURN
4410 REM #
4420 REM TEST SYMBOLS FOR ELEMENTS
4430 REM #
4440 CLS
4450 GOSUB 1930 'TITLE
4460 PRINT@65,"TEST 1 - SYMBOLS FOR ELEMENTS"
4470 PRINT
4480 PRINT"  IN THIS TEST THE NAMES OF"
4490 PRINT"DIFFERENT ELEMENTS ARE SHOWN ON"
4500 PRINT"THE SCREEN.YOU HAVE TO TYPE IN"
4510 PRINT"THE CORRECT CHEMICAL SYMBOL FOR"
4520 PRINT"EACH ELEMENT."
4530 PRINT"  FOR EXAMPLE IF THE ELEMENT "
4540 PRINT"DISPLAYED IS 'MAGNESIUM' YOUR"
4550 PRINT"REPLY SHOULD BE 'MG'."
4560 PRINT"  THERE ARE 12 QUESTIONS IN THE"
4570 PRINT"TEST."
4580 PRINT@483,"PRESS ANY KEY TO CONTINUE";
4590 IF INKEY$=""THEN4590
4600 FOR I=1 TO 12
4610 TEST(I)=RND(103)
4620 FOR J=0 TO I-1
4630 IF TEST(J)=TEST(I) THEN TEST(I)=RND(103):J=0
4640 NEXT J
4650 NEXT I
4660 SCORE=0
4670 CLS
4680 GOSUB 1930 'TITLE
4690 PRINT"      element          symbol"
4700 FOR I=1 TO 12
4710 PRINT TAB(3);NAME$(TEST(I));TAB(19);
4720 INPUTANS$(I)

```

```

4730 IF LEN(ANS$(I))>2 THEN ANS$(I)=LEFT$(ANS$(I),2
)
4740 NEXT I
4750 PRINT@483,"PRESS ANY KEY FOR ANSWERS";
4760 IF INKEY$="" THEN 4760
4770 CLS
4780 GOSUB 1930 'TITLE
4790 PRINT"element          your reply symbol"
4800 POKE1074,32
4810 FOR I=1 TO 12
4820 PRINTNAME$(TEST(I));TAB(18);ANS$(I);TAB(27);S
YMBOL$(TEST(I));TAB(31);
4830 IF ANS$(I)=SYMBOL$(TEST(I)) THEN PRINT"r";:SC
ORE=SCORE+1 ELSE PRINT"X";
4840 NEXT I
4850 PRINT"          SCORE =";SCORE;"OUT OF 12"
4860 PRINT@483,"PRESS ANY KEY TO CONTINUE";
4870 IF INKEY$="" THEN 4870
4880 RETURN
4890 REM #
4900 REM TEST2 NAMES FOR SYMBOLS
4910 REM #
4920 CLS
4930 GOSUB 1930 'TITLE
4940 PRINT@65,"TEST 2 - ELEMENTS FOR SYMBOLS"
4950 PRINT
4960 PRINT"  IN THIS TEST THE SYMBOLS OF"
4970 PRINT"DIFFERENT ELEMENTS ARE SHOWN"
4980 PRINT"ON THE SCREEN.YOU HAVE TO TYPE"
4990 PRINT"IN THE CORRECT ELEMENT NAME."
5000 PRINT"  FOR EXAMPLE IF THE SYMBOL"
5010 PRINT"DISPLAYED IS 'LI',YOUR REPLY"
5020 PRINT"SHOULD BE 'LITHIUM'."
5030 PRINT"  THERE ARE 12 QUESTIONS IN THE"
5040 PRINT"TEST."
5050 PRINT@483,"PRESS ANY KEY TO CONTINUE";
5060 IF INKEY$="" THEN 5060
5070 FOR I=1 TO 12
5080 TEST(I)=RND(103)
5090 FOR J=0 TO I-1
5100 IF TEST(J)=TEST(I) THEN TEST(I)=RND(103):J=0
5110 NEXT J
5120 NEXT I
5130 SCORE=0
5140 CLS
5150 GOSUB 1930 'TITLE
5160 PRINT"          symbol          element"
5170 FOR I=1 TO 12
5180 PRINTTAB(5);SYMBOL$(TEST(I));TAB(15);
5190 INPUTANS$(I)
5200 IF LEN(ANS$(I))>13 THEN ANS$(I)=LEFT$(ANS$(I),
13)

```

```

5210 NEXT I
5220 PRINT@483,"PRESS ANY KEY FOR ANSWERS";
5230 IF INKEY$="" THEN 5230
5240 CLS
5250 GOSUB 1930 'TITLE
5260 PRINT"sm your reply      element name"
5270 POKE1063,32:POKE1069,32:POKE1070,32:POKE1071,
32:POKE1080,32
5280 FOR I=1 TO 12
5290 PRINTSYMBOL$(TEST(I));TAB(3);ANS$(I);TAB(17);
NAME$(TEST(I));TAB(31);
5300 IF ANS$(I)=NAME$(TEST(I)) THEN PRINT"r";:SCORE
E=SCORE+1 ELSE PRINT"X";
5310 NEXT I
5320 PRINT"      SCORE =" ;SCORE;"OUT OF 12"
5330 PRINT@483,"PRESS ANY KEY TO CONTINUE";
5340 IF INKEY$="" THEN 5340
5350 RETURN

```

25. Animals

General Description

This is a delightful program which captivates the young and the not so young. It is the only program in this book in which the computer 'learns' from the user.

The computer invites the user to think of an animal, and asks a series of questions before attempting a guess. If incorrect, a few more questions are asked to establish the new animal in the computer's repertoire.

The theme of this program is well known, and it nicely illustrates the use of a binary tree structure. Since the answer to a question must either be a yes or no, the program traverses the tree either left or right. If the computer doesn't guess the animal correctly, it obtains a new question to distinguish it from other animals, and updates the tree structure.

The user can also list the animals which are already known to the computer, save the game on a tape file, and load back the file to continue the same game.

Detailed Description

Lines 10–50 Clears string space and sets maximum array size. Dimensions TRY\$ array to hold questions and animals, RPTR to point to next element if right, WPTR to point to next element if wrong.

60–70 Perform subroutines to initialise arrays with starter question and animals, and to display instructions.

80–240 Display options, get reply.

250–430 Start of main game loop. Ask user to think of animal, and try questions.

440–510 Ask user if animal is correct.

520–660 If new animal, get distinguishing question.

670–800 Update tree structure.

810–870 Guessed animal correctly; invitation to continue.
880–950 Subroutine to check keyboard input.
960–1170 Subroutine to load saved game from tape.
1180–1410 Saves game on tape.
1420–1490 Gets and checks valid filename.
1500–1680 Lists animals learned so far.
1690–2130 Fancy title and introduction.
2140–2230 Subroutine to initialise starter ‘zoo’. If you want to alter the start data, a ‘yes’ reply to the question points to the animal in line 2200, a ‘no’ points to the animal in line 2220.

Educational Notes

Mainly for younger children, this game can be used to encourage them to think about animals, their individual characteristics, and the differences between them.

The structure of the program can be applied to a variety of topics and, with a little imagination, could be developed as a learning aid for older pupils.

You may consider loading the program with a pre-planned set of questions and animals (or whatever) and give the children a ‘target’ animal to see if they can answer all questions correctly and arrive at the target.

Program Listing

```

REM*****
2 REM* ANIMAL LEARNING PROGRAM *
3 REM* *
4 REM* WRITTEN FOR THE DRAGON *
5 REM* *
6 REM* BY LAURENCE OWEN 1983 *
7 REM*****
10 CLEAR12000
20 REM ***SET MAX ARRAY SIZE***
30 MAX=400
40 DIM TRY$(MAX),RPTR(MAX),WPTR(MAX)
50 ZOO=2:LAST=3
60 GOSUB 2170 'START ZOO
70 GOSUB 1720 'INTRODUCTION
80 REM ***** DISPLAY MENU *****
90 CLS6
  
```

```

100 PRINT@64," CHOOSE ONE OF THE FOLLOWING ":"PRIN
T:PRINT
110 PRINT" 1 LOAD ZOO FROM TAPE"
120 PRINT" 2 SAVE ZOO ON TAPE"
130 PRINT" 3 LIST ANIMALS"
140 PRINT" 4 ## PLAY THE ANIMALS GAME ##"
150 PRINT" 5 QUIT":PRINT
160 PRINT:PRINT:PRINT" ENTER THE NUMBER OF YOUR CH
OICE";
170 FL$=INKEY$
180 KB$=INKEY$:IFKB$=""THEN180
190 A=INT(VAL(KB$))
200 IF A<1 OR A>5 GOTO 90
210 IFA=5 THEN CLS:PLAY"ADA":END
220 IF A=4 GOTO 260
230 ON A GOSUB 990,1210,1530
240 GOTO 90
250 REM *****START HERE*****
260 CURR=2
270 CLS2
280 PRINT@103," PLEASE THINK OF ";
290 PRINT@135," AN ANIMAL ";
300 FORD=1T02000:NEXT
310 PLAY"T16BABBAGE"
320 PRINT@231," READY ? "":PRINT@243,;
330 GOSUB 910 'YES/NO REPLY
340 IFKB$="N" THEN PRINT@236," "":GOTO300
350 CLS
360 IF RPTR(CURR)=0 THEN 450
370 HOLD=CURR
380 PRINT:PRINTSTRING$(32,CHR$(147));:PRINT TRY$(C
URR);" ?";
390 PLAY"DAB"
400 GOSUB910 'YES/NO REPLY
410 IF KB$="Y" THEN CURR=RPTR(CURR) ELSE CURR=WPTR
(CURR)
420 A$=KB$
430 GOTO 360
440 REM *****ANIMAL TO TRY*****
450 FI$=LEFT$(TRY$(CURR),1)
460 P=INSTR(1,"AEIOU",FI$)
470 IF P=0 THEN FI$=" " ELSE FI$="N "
480 PRINT:PRINTSTRING$(32,CHR$(243));:PRINT"IS THE
ANIMAL YOU'RE THINKING","OF A";FI$;TRY$(CURR);" ?
";
490 PLAY"ACE"
500 GOSUB910 'YES/NO REPLY
510 IF KB$="Y" THEN 820 'RIGHT
520 REM *****DONT KNOW-ASK*****
530 PRINT:PRINTSTRING$(32,CHR$(163));:PRINT"I GIVE
UP !! WHAT ANIMAL WERE","YOU THINKING OF ";
540 PLAY"DEAD"

```

```

550 INPUT ANIMAL$
560 FA$=LEFT$(ANIMAL$,1)
570 F=INSTR(1,"AEIOU",FA$)
580 IF P=0 THEN FA$=" " ELSE FA$="N "
590 PRINTSTRING$(32,CHR$(179));:PRINT"PLEASE TYPE
  A QUESTION THAT"
600 PRINT"WILL DISTINGUISH BETWEEN A";FI$:PRINTTRY
$(CURR);" AND A";FA$;ANIMAL$
610 PLAY"FEED"
620 INPUT QUES$
630 IF RIGHT$(QUES$,1)="?" THEN QUES$=LEFT$(QUES$,
  LEN(QUES$)-1)
640 PRINTSTRING$(32,CHR$(211));:PRINT"WHAT WOULD B
  E THE CORRECT ANSWER";"FOR A";FA$;ANIMAL$;" ?";
650 PLAY"BAG"
660 GOSUB 910      *YES/NO REPLY
670 REM ***ADD ANIMAL TO ZOO***
680 ZOO=ZOO+1:LAST=LAST+2
690 IF A$="Y" THEN RPTR(HOLD)=LAST-1 ELSE WPTR(HOL
  D)=LAST-1
700 IFKB$="N"THEN 740
710 RPTR(LAST-1)=LAST
720 WPTR(LAST-1)=CURR
730 GOTO 760
740 WPTR(LAST-1)=LAST
750 RPTR(LAST-1)=CURR
760 TRY$(LAST-1)=QUES$
770 TRY$(LAST)=ANIMAL$
780 PRINT:PRINTSTRING$(32,CHR$(227));:PRINT"    I N
  OW KNOW ";ZOO;" ANIMALS!"
790 PLAY"BAGGED"
800 GOT0840
810 REM *****GOT IT RIGHT*****
820 PRINT:PRINTSTRING$(32,CHR$(195));:PRINT"
  I GOT IT RIGHT !!"
830 PLAY"CAGE"
840 PRINT:PRINT"  DO YOU WANT TO PLAY AGAIN ?";
850 GOSUB 910      *YES/NO REPLY
860 IF KB$="N" THEN 80
870 GOTO 260
880 REM *****
890 REM **   GET REPLY SUB   **
900 REM *****
910 FL$=INKEY$
920 KB$=INKEY$:IFKB$=""THEN920
930 IFKB$<>"Y"ANDKB$<>"N"THEN910
940 PRINT" ";KB$;
950 RETURN
960 REM *****
970 REM **LOAD ZOO FROM TAPE**
980 REM *****
990 CLS:PRINT"load zoo from tape"

```



```

1000 LS$="LOAD":GOSUB1450 'NAME
1010 IF NA$="" THEN RETURN
1020 CLS:PRINT"load zoo from tape"
1030 PRINT:PRINT"POSITION TAPE AND THEN"
1040 PRINT"PRESS PLAY"
1050 PRINT:PRINT"PRESS <ENTER> WHEN READY."
1060 IF INKEY$<>CHR$(13) GOTO 1060
1070 PRINT:PRINT"SEARCHING FOR ";NA$
1080 OPEN "I",#-1,NA$
1090 IF EOF(-1) GOTO 1150
1100 INPUT #-1,ZOO,LAST
1110 FOR I=1 TO LAST
1120 IF EOF(-1) GOTO 1150
1130 INPUT #-1,TRY$(I),RFTR(I),WPTR(I)
1140 NEXT I
1150 CLOSE #-1
1160 PRINT @256,"DATA FILE ";NA$;" LOADED."
1170 GOTO 1390
1180 REM *****
1190 REM ** SAVE ZOO ON TAPE **
1200 REM *****
1210 CLS:PRINT"save zoo on tape"
1220 LS$="SAVE":GOSUB1450 'NAME
1230 IF NA$="" THEN RETURN
1240 CLS:PRINT"save zoo on tape"
1250 PRINT:PRINT"POSITION TAPE AND THEN"
1260 PRINT"PRESS PLAY AND RECORD"
1270 PRINT:PRINT"PRESS <ENTER> WHEN READY."
1280 IF INKEY$<>CHR$(13) GOTO 1280
1290 PRINT:PRINT"SAVING DATA FILE ";NA$
1300 OPEN "O",#-1,NA$
1310 PRINT #-1,ZOO,LAST
1320 FOR I=1 TO LAST
1330 PRINT #-1,TRY$(I),RFTR(I),WPTR(I)
1340 NEXT I
1350 CLOSE #-1
1360 PRINT @256,"DATA FILE SAVED"
1370 GOTO 1390
1380 REM ***WAIT FOR <ENTER>***
1390 PRINT:PRINT"PRESS <ENTER> TO CONTINUE.";
1400 IF INKEY$<>CHR$(13) GOTO 1400
1410 RETURN
1420 REM *****
1430 REM ** GET FILENAME **
1440 REM *****
1450 PRINT:PRINT"PLEASE ENTER FILENAME FOR ";LS$
1460 PRINT"(MAXIMUM OF 8 LETTERS):"
1470 PRINT:PRINT:INPUT">>>";NA$
1480 IF LEN(NA$)>8 THEN 1450
1490 RETURN

```



```

1500 REM *****
1510 REM ** LIST ANIMALS **
1520 REM *****
1530 CLS
1540 FOR I=1 TO 20
1550 N=RND(8):CLS
1560 PRINT@203," CHEAT !! ";
1570 SOUND 64+16*N,1
1580 NEXT:CLS
1590 FOR I=1 TO LAST STEP 2
1600 PRINT TRY$(I),
1610 IF I=LAST THEN1630
1620 IF INT(I/59)*59<>I THEN1670
1630 PRINT@489,"PRESS ANY KEY";
1640 FL$=INKEY$
1650 KB$=INKEY$:IFKB$=""THEN1650
1660 CLS
1670 NEXT
1680 RETURN
1690 REM *****
1700 REM ** INTRODUCTION **
1710 REM *****
1720 CLS0
1730 YE$=STRING$(4,CHR$(159))
1740 TA$=STRING$(4,CHR$(255))
1750 PRINT@106,TA$;TA$;TA$;
1760 PRINT@136,TA$;YE$;YE$;TA$;
1770 PRINT@166,TA$;YE$;" ";YE$;TA$;
1780 PRINT@196,TA$;YE$;" ANIMAL ";YE$;TA$;
1790 PRINT@226,TA$;YE$;" LEARNING ";YE$;TA$;
1800 PRINT@260,TA$;YE$;" GAME ";YE$;TA$;
1810 PRINT@294,TA$;YE$;" ";YE$;TA$;
1820 PRINT@328,TA$;YE$;YE$;TA$;
1830 PRINT@362,TA$;TA$;TA$;
1840 FORD=1TO8000:NEXT
1850 CLS6
1860 PRINT@33," ### ANIMAL LEARNING GAME ### ";
1870 PRINT@65,STRING$(30,CHR$(32));
1880 PRINT@97," YOU MUST THINK OF AN ANIMAL ";
1890 PRINT@129," AND I WILL TRY TO GUESS WHAT ";
1900 PRINT@161," ANIMAL YOU ARE THINKING OF. ";
1910 PRINT@225," I WILL ASK YOU A SERIES OF ";
1920 PRINT@257," QUESTIONS TO WHICH YOU REPLY ";
1930 PRINT@289," EITHER YES OR NO - 'Y' OR 'N' ";
1940 PRINT@321," WILL DO. IF I CAN'T GUESS ";
1950 PRINT@353," YOUR ANIMAL,I WILL ASK A FEW ";
1960 PRINT@385," QUESTIONS SO THAT YOU CAN ";
1970 PRINT@417," TEACH ME ABOUT YOUR ANIMAL. ";
1980 PRINT@481," PRESS ANY KEY TO CONTINUE ";
1990 FL$=INKEY$
2000 KB$=INKEY$:IFKB$=""THEN2000
2010 CLS6

```

```

2020 PRINT@33," WHEN YOU HAVE TAUGHT ME SOME ";
2030 PRINT@65," ANIMALS,YOU CAN SAVE MY 'ZOO'";
2040 PRINT@97," ON A TAPE FILE IF YOU WISH. ";
2050 PRINT@161," IF YOU HAVE TAUGHT ME SOME ";
2060 PRINT@193," ANIMALS BEFORE AND SAVED THE ";
2070 PRINT@225," 'ZOO' ON A FILE,YOU CAN LOAD ";
2080 PRINT@257," IT BACK IN AND TEACH ME SOME ";
2090 PRINT@289," MORE ANIMALS. ";
2100 PRINT@481," PRESS ANY KEY TO CONTINUE ";
2110 FL$=INKEY$
2120 KB$=INKEY$:IFKB$=""THEN2120
2130 RETURN
2140 REM *****
2150 REM ** LOAD STARTER ZOO **
2160 REM *****
2170 FOR I=1 TO 3
2180 READ TRY$(I),RPTR(I),WPTR(I)
2190 NEXT
2200 DATA DRAGON,0,0
2210 DATA DOES IT BREATHE FIRE,1,3
2220 DATA GOOSE,0,0
2230 RETURN

```

Software

If you are interested in cassette versions of the programs featured in this book, please write for details to:

Dept. S (D)
Century Publishing
76, Old Compton Street
London, W1V 5PA

Available in Century's Science & Technology series

DICTIONARY OF NEW INFORMATION TECHNOLOGY

A. J. Meadows, M. Gordon and A. Singleton

WHAT TO BUY FOR BUSINESS: A HANDBOOK OF NEW
OFFICE TECHNOLOGY

John Derrick and Phillip Oppenheim

THE ELECTRONIC MAIL HANDBOOK

Stephen Connell and Ian A. Galbraith

THE WAY THE NEW TECHNOLOGY WORKS

Ken Marsh

MIND

David A. Taylor

In association with 'Personal Computer World'

MICROCOMPUTING FOR BUSINESS: A USER'S GUIDE
edited by Dick Olney

THE MICROCOMPUTER HANDBOOK: A BUYER'S GUIDE
edited by Dick Olney

THE SPECTRUM HANDBOOK
Tim Langdell

35 PROGRAMS FOR THE DRAGON 32
Tim Langdell

THE INTIMATE MACHINE
Neil Frude

CENTURY COMPUTER PROGRAMMING COURSE
Peter Morse and Ian Adamson

The Dragon 32 is becoming a bestseller on the micro market yet most of the software available is restricted to games programs. Here at last is a selection of educational programs ideal for the younger user. The subject areas covered include science, history, geography, spelling, computer science and maths. All are fully documented with useful tips and are great fun to use as well as providing an insight into programming techniques.

A version of this book is available for Spectrum and BBC micro users.