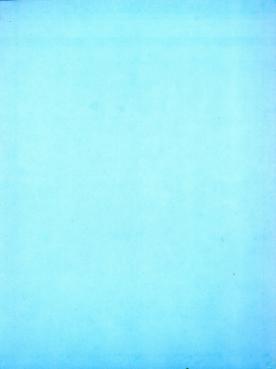
THE RAINBOW BOOK OF BASIC PROGRAMS

GAMES, PROBLEMS AND PUZZLES FOR **DRAGON AND VIC20 COMPUTERS** 無

This edition produced exclusively for





THE RAINBOW BOOK OF BASIC PROGRAMS

GAMES, PROBLEMS AND PUZZLES FOR SPECTRUM, BBC, ELECTRON, DRAGON AND VIC20 COMPUTERS



EDITOR

DESIGN Maynard Ben White

ILLUSTRATIONS

The Hayward Art Group

COVER

Pinpoint

PROGRAMS

Pat Grady Paul Greet Doug Gregory Gordon Lee Paul McGee Malcolm Neave

PHOTOGRAPHY

Paul Wilson

Special thanks to Devron Computer Centre for making available computers for photography.

First published in 1984 by Grisewood & Dempsey Limited, Elsley Court, 20-22 Great Titchfield Street, London W1P 7AD

@ Grisewood & Dempsev Ltd 1984

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the Copyright owners.

ISBN 0 906279 26 7

CONTENTS

CHAPTER ONE	Midsummer's Eve
low to Use This Book9-10	CHAPTER SIX: NUMBER PUZZLES
CHAPTER TWO: FIVE MICROS	High-Low62
pectrum 11 BC 12 Ectron 13 IC20 14 ragon 13	Magic Squares 66 Mindread 66 Air Rescue 68 Swap 7 Anvday 77
CHAPTER THREE	Star Course
ibrary of Subroutines16-21	CHAPTER SEVEN: SCIENCE PROBLEMS
CHAPTER FOUR: GAMES OF FUN	Conversion
omber 21 upid Fish 2- wiss Navy 22 it or Miss it or Miss 25 igzag Golf 33 oadrace 34	Tumbledice 99 Fast Brakes 92 Pay As You Go 99 Moon 99 Pinball 98
quash	CHAPTER EIGHT: WORD PROBLEMS
CHAPTER FIVE: THINKING GAMES	Canute 100 Lastword 102 Code 104
yramid Race 44 ungson 44 ungson 44 ungron 44 urines 44 dadam I'm Adam 55 (eps and Leaps 55)	Wordgrid.
rossnumber (2)	



HOW TO USE THIS BOOK

Read this section and the Subroutine Library first before you use the main programs.

INTRODUCTION

The Rainbow Book of Programs contains more than 50 master programs and program variations. They can be used with Spectrum, BBC, Electron, VIC20 and Dragon micro-computers. The programs in-

computers. The programs include games, problems, logic puzzles and tests of skill. They can be used on their own or to form the core of programs written by yourself. None take more than 16K of memory.

The master programs have been designed to work with five different machines. Therefore they only contain lines which are common to all. In order to make a complete working program you merge the master programs with the right subroutines. Each computer has its own set

of subroutines found in the Subroutine Library chapter. They are quite short. Once you have typed and saved the right ones they will adapt all the programs to your own computer.

You will need a tape recorder when using this book as it will save you a lot of time typing and retyping the subroutines.

TYPING PROGRAMS

Type in all the subroutines and programs exactly as they are listed. Make sure to copy all the word or symbol.

line numbers, words, symbols, punctuation and spaces exactly as they appear in the listings found in this book.

2) For the Spectrum do not type in computer instructions letter by letter but use the keys which enter a complete word at a time. With the Electron you have a choice of either entering complete words or else typing one letter at a time.

3) Be sure to press ENTER or RETURN or NEWLINE at the end of each complete line so that it is stored in the machine's memory before you start the next line of the program.

4) Use capital letters for all the programs in the same way as they appear in the book. The manual for your machine will explain how to obtain caps shift to do this.

5) The printout of the programs clearly distinguishes between the number zero and the letter O. Since the letter O is never used as a variable you can assume that whenever you see it on its own it is a zero.

 While typing in a program keep a ruler under the line on which you are working so that you do not enter material from lines below by accident.

7) To help you enter the correct number of spaces for the sections of the listings inside quotation marks, spacing symbols have been used. They look like this: 110 PRINT "BOM-

BER". This means that six spaces (_____ must be entered before typing the word BOMBER. Do not try to type in anything when you come to these symbols. They are there simply to help you count the correct number of spaces before you begin to type a word or swmbol.

RUNNING PROGRAMS

Once you have entered a complete subroutine and master program type RUN to start the program. At the end of each trial of a program you will be left with a stop message on the screen. Type RUN to make the program start again.

2) When you run a program for the first time you will also be testing it for mistakes. The computer will stop automatically if there are any bugs in the program. However, you cannot test the subroutines unless they are in a master program. Your very first program, therefore, will also be a test of the subroutines.

FINISHING PROGRAMS When you are finished with a

program, and have saved it for future use, you can clear it from memory. Refer to the Five Micros chapter for instructions how to clear your particular machine.

SAVING PROGRAMS

The best way to work with the programs in this book is to type and save the appropriate subroutines first of all. Once you have stored these on cassette you can use them with each new program you try out.

 When starting a new program (but after you have first saved the right subroutines for your machine) use the following procedure:

(a) Load the subroutine
(b) Type in the new program

(b) Type in the new program(c) When finished, save the program and subroutine

together on cassette.

3) It is always a good idea to save a program for future use. Use the

HOW TO USE THIS BOOK

program names found in the REM statements – these names are suited to all the machines. For detailed instructions how to save a program go to the Five Micros chapter that follows.

Micros chapter that follows.

4) You can build up a library of programs on cassette. Be sure to label each program and the tape position at which it is stored.

LOADING PROGRAMS

 A program or subroutine has to be loaded by the name it was saved; check the master programs in the book if you are not sure of a name.

sure of a name.

2) Remember that all names are in capital letters. You must load

them in the same way.

3) If you have made corrections to a subroutine or program you should also change the saved version so that future runs are error free.

VARIATIONS

Several master programs also contain suggestions for variations. These may make the program run faster or add a new scoring system. You can also make up your own variations. The easiest place to start are the lines that print messages. You can add anything you wish here as long as it makes sense

when the program runs.
2) The programs in this book are
all short and simple. They are
easy to crash if you do not use
them the way they are designed.
As an exercise you might try
making them more crash proof,
although this will greatly lengthen the listings.

PROGRAM DOCTOR

1) If a program stops when you try to run it, return to the listing place for mistakes to occur.

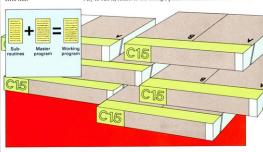
by typing LIST. As BBC and Electron machines run in Mode 5 for all the programs it is best to return to a mode that is easier to read. See the Five Micros chapter for instructions about returning to normal text mode.

 Check first of all that you have not missed out any lines in the program and that all lines have the right number.

Check that no two lines have been joined into one.

4) Starting at the beginning, check the program line by line making sure that punctuation is correct – do not confuse colonic (c) with semi-colons (c), that zero and capital O have not been confused, that all variables are correctly entered and the spacing is right.

5) As you cannot check the subroutines until you run your first program, this is a likely



SPECTRUM Having connected your Spec-

trum, TV or monitor and tape recorder, the first step in using the programs in this book is to type in and load the library of subroutines. They are essential. Whenever you enter programs

Whenever you enter programs or subroutines on your Spectrum (or when you save or load programs from this book) you must be sure to use only capital letters – exactly as shown in the listings. By pressing the CAPS SHIFT and CAPS LOCK keys you will put the machine into the right mode.

Once you have typed in all the

SUBROUTINES

subroutines double check the screen listing with that in the book. If there are no errors then you should save the subroutines onto the cassette as follows:

- 1) Type SAVE "
 ROUTINE".
- Start the tape recorder so that it records from the beginning of a blank cassette. Make sure that only the MIC sockets of the computer and recorder are connected.
- 3) Press ENTER.
- 4) When the report 9 OKappears on the screen stop the tape recorder. The subroutines will now have been saved on cassette. To double check run the tape back to the beginning. Connect the EAR leads and type in VERIFY "SUBROU-TINE". Start the tape and press ENTER. If the words "Program Subroutine", appear and then the report 90 OK: the subroutines will



have been correctly saved onto the cassette. The procedure for saving a

complete program (including subroutines) is exactly the same except that you must type in the name of the program (found in the REM statement in the listing) together with the SAVE instructions.

MAIN PROGRAMS

When you begin to enter one of the master programs the sequence becomes slightly different as you must first load the subroutines into the machine before any other instructions.

Rewind the subroutines cassette to the beginning. Make sure that only the EAR sockets are connected. Set the volume control to around 3/4 maximum and then:

1) Type LOAD "SUB-ROUTINE".

- Start the recorder playing.
 Press ENTER.
 - When the report '0 OK,0:1' appears the computer will have been loaded. Now you are ready to type-in the rest of the program.

If you have trouble with saving and loading subroutines and programs refer to the machine's manual for detailed advice about correcting it.

After you are through working with a program and have saved a copy of it for future reference, the simplest way to clear the machine is by entering NEW (to erase the program from memory). You are now ready to try a new program.

BRC

Once you have connected your BBC to a tape recorder and TV or monitor the next step before starting with any programs is to type in and load the appropriate library of subroutines for this machine. These subroutines work equally well for 1.2 and 0.1 operating systems.

Whenever you enter programs or subroutines from this book

or subroutines from this book you must be careful to use only capital letters – exactly as in the listings – by pressing CAPS LOCK. Since the BBC treats a capital 'A' as something different than a lower case 'a' this procedure is an important on

SUBROUTINES

After typing in all the subroutines double check the listing on the screen with that in the book. If you can find no errors then save the subroutines onto a cassette in the following way:

 Type in SAVE "SUB-ROUTINE". The message 'Record then return' will appear on screen.

 Start the tape recorder so that it records from the beginning of a cassette.

3) Press RETURN.
4) While it is saving, the name
of the program and various
numbers appear on screen.
When the cursor prompt
reappears stop the tape. The
subroutines have now been
saved. To double check
there are no mistakes rewind the casette and type
in *CAT. Play the tape
through so the BBC can
verify the information that
was recorded. If there are



no errors it will print 'Subroutine' and the same numbers as before.

The process for saving a com-

plete program, including all subroutines, is exactly the same except that you enter the name of the program (taken from the REM statement) together with the SAVE instructions.

MAIN PROGRAMS

When you move on to entering one of the master programs the sequence is slightly different. First of all, load the subroutines before any other instructions go into the program.

Rewind the subroutines cassette. Set the volume control to about 3/4 maximum and then:

- Type LOAD "SUB-ROUTINE".
 Press RETURN.
- The message 'Searching' appears. Start the recorder playing. When the machine

finds the subroutines it prints 'Loading' on screen.

the program.

 Once the subroutines are loaded the cursor prompt reappears. Now you can proceed to enter the rest of

If you are having a problem saving and loading refer to the machine's manual for detailed advice on how to correct it.

Once you have finished with a program save a copy for future reference. The BBC can now be cleared by typing NEW to erase the program from memory.

The programs in this book all make the BBC run in Mode 5 giving large size text that is easy to read on a TV screen. However, if you type in LIST to look at the program lines Mode 5 text becomes hard to read. To return to normal size text enter MODE 7, then type LIST and the program will reappear.

ELECTRON

After you have connected up your Electron, TV or monitor and tape recorder the first step before loading any programs is to type in the appropriate library of subroutines for the computer. Without them it is impossible to make the programs work.

Whenever you enter listings from this book you must make sure to use only capital letters – exactly as they are printed. By pressing the key marked CAPS LK you ensure that every letter you type is a capital. The Electron, like the BBC, treats capital APS as different character than lower case 'a' – therefore the procedure is an important one.



Once you have entered all the subroutines double check the screen listing with that in the book. If you can find no errors save the subroutines onto a fresh cassette as follows:

Type in SAVE "SUB-ROUTINE". The screen message 'Record then re-

Start the tape recorder so that it is recording from the beginning of the cassette.
 Press RETURN.

turn' appears.

4) While the computer is saving, the name 'Subroutine' and various numbers will be seen on the screen. When the cursor prompt reapears stop the tape. Double check that the subroutines have been correctly saved by rewinding the cassette and typing in "CAT. Play the tape all the way through again. If everything is



alright you will get 'Subroutine' on screen and the same numbers as before. The procedure for saving a

finished program, including the master listing and the subroutines, is exactly the same. The only change is that you must enter the name of the program (taken from the REM statement in the first line) between the quotes.

MAIN PROGRAMS The procedure changes once you

move on to entering one of the main programs. Here you must start by loading the subroutines before any other program lines. Rewind the cassette Set the

Rewind the cassette. Set the volume controls to about 3/4 maximum and then:

1) Type LOAD "SUB-ROUTINE". 2) Press RETURN.

3)The message 'Searching'

appears on screen. Start the recorder playing. When the computer finds the right place it prints the message 'Loading'.

 Once the subroutines have been loaded the cursor prompt reappears. Now proceed to enter the rest of the program.

After you have finished running a program save a copy for future reference on a new cassette. The machine can then be cleared by entering NEW to crase the old program from memory. You can now start a new program.

make the Electron run in Mode 5

— a large size text that is easy to
make out on a screen. However, if
you LIST a program this Mode is
hard to read. You can return to
normal size text by entering
MODE 6, then typing LIST.

VIC20

Once the VIC20 is connected up with your TV or monitor and tape recorder the next stage before loading any programs is to type in the library of subroutines suited to this computer. Without them it is impossible to get any of the programs to work.

SUBROUTINES

After you have entered all the subroutines double check the listing on screen with that in the book. If you can find no errors save the subroutines onto a blank cassette as follows:

1) Type in SAVE "SUB-

- ROUTINE".

 2) A screen message 'Press Record and Play on Tape'
- appears.

 3) Switch on the tape recorder
- so that it is recording from the beginning of the cassette.
- The computer will print the message 'OK Saving' as it begins.
 The message 'OK Ready'
- will come up on screen when the computer has finished saving. Stop the tape. To double check that the subroutines have becorrectly stored rewind the cassette to the beginning.
- 6) Type in VERIFY "SUB-ROUTINE". This is followed by the message 'Play on Tape'. Start the tape running, If errors are found they show up as a '2Verify Error' message. Otherwise 'OK Ready' reports that the checking has been completed and no mistakes have been found.



completed program (including all the subroutines) is very similar. The only difference is that you must remember to enter the name of the program (from the REM statement in the first line of the listing) between the quotes.

The procedure for saving a

MAIN PROGRAMS

When you move on to typing in the master programs a different procedure has to be used. The saved subroutines must be loaded first before any of the other lines of the program. Rewind the subroutine tape

- and then proceed as follows:

 1) Type LOAD "SUB-ROUTINE" and the message 'Press Play on Tape'
- Start the recorder. The computer prints the message 'OK Searching', then 'Found' and 'Loading'.

will appear.

 Once the subroutines have finished loading the machine displays a 'Ready' message.

 If you have any difficulty sav-

If you have any difficulty saving and loading programs refer to the VIC20 manual for more detailed advice on how to correct the problem.

After you have finished workwith a program save a copy
for future use on a new cassette.
The computer can then be
cleared by entering NEW to
erase the program from memory.
You are now able to start on a
new program.

DRAGON

When the computer is connected up to the tape recorder and TV or monitor, the first thing to do before loading any programs is to type in the appropriate library of subroutines for the Dragon computer. Without these it is impossible to make the programs run at all.

Whenever you enter listings you must be sure to use only papital letters – exactly as they are printed in the book. However, as the Dragon normally works in capitals you do not have to press the SHIFT key to obtain them.



Once you have a ped in all the subroutines double check the listing on the screen with the one in the book. If there are no errors save the subroutines onto a new cassette as follows:

 Press dow i the play and record buttons of the tape recorder.

2) Type and enter CSAVE "SUBROUTE" (the name of the program cannot be more that eight characters long). Once it is enter. ³ Le tape will start to run and record the program.

 When the program ¹ as been saved the OK prompt will

return to the screen.

The procedure for saving a finished program (including the master listing and the sub-routines) is very similar. The only difference is the name of the program. Use names taken from the REM statement in the first line of the master listing and the same CSAVE instructions.



MAIN PROGRAMS The procedure for entering the

main programs is somewhat different. Here you must start by loading the subroutines before ...y of the other program lines.

...y of the other program lines. Rewind the cassette containing the subroutines. Set the volume controls to the correct level and then continue as follows:

Press down the play button of the tape recorder.
 Type and enter CLOAD "SUBROUTE"

 The tape recorder will start and the letter 'S' will appear in the upper left corner of the screen while the computer searches for the program.

4) The 'S' changes to 'FSUB-ROUTE' when the program is found and is being loaded. When the OK prompt reappears the loading is finished. Now proceed to enter the rest of the program.

If you have any problems saving and loading refer to the Dragot's manual for detailed advice how to deal with them.

After you have finished running a program save a copy for future reference onto a new cassette. The computer can then be cleared by entering NEW to crase the memory. You can now start a new program.

INTRODUCTION

The first thing to do when trying the programs in this book is to turn to the library of subroutines that are right for your computer. These subroutines are the only way to make the master programs work. But be very careful when you are typing them in as you will have to use the subroutines often and any mistakes that are made will not show up until you try to run your first programs. Double check the screen listing with the one in the book before you save the subroutines onto cassette.

instruction RETURN is a complete subroutine in its own right. An entire library of 20 or so subroutines for each machine is needed to make the master programs in this book work.

It certainly is helpful (and less confusing) to keep the subroutines on a separate cassette from the completed programs. You will be loading them frequently as you work through the book and it is not a good idea to confuse them with other pro-

Each line or set of lines begin- For all six computers an identical ning with a number in the procedure can be used to merge

thousands and ending with the the subroutines you have saved onto cassette with the main programs that you will be typing in later

- 1) Load the subroutines from cassette to computer.
 - 2) Type in the master program line by line. As the line numbers of the subroutines are greater than 1000 they will not interfere with the main programs, which are all numbered in the hundreds.
 - 3) Run the complete program to test that it works correctly.
 - 4) Save the completed working program onto a new cassette for future use.

VARIATIONS

You may find that some of the programs run too slowly or too quickly for your taste. By changing the value of the delay loop in subroutine 1400 you can slow things down or speed them up.

In the rest of the book the variable T is always used to govern the speed at which things happen. Try experimenting with the values it takes to see how these affect a program.



SPECTRUM

Line 1000: Names the subroutine library which follows. Line 1100: Sets up the area of the screen used for text and puts the cursor down at row 8.

Line 1200: Clears the screen and sets the cursor down at row 8. Line 1300: Generates a random number.

Line 1400: Sets up a time delay while the computer runs through a loop 200 times.

Lines 1500-1520: Wait for a character entered by the user. Line 1600: Checks to see if the keyboard is being pressed.

Line 1700: Prints at position X and Y.
Line 1800: Prints at position X.

Line 1900: Moves the cursor up one line.

forward one column.

Line 2100: Inputs a number as a

variable called K. Lines 2200-2240: Input information as a string variable

called K\$.
Line 2300: Clears screen and
sets cursor at starting position.
Line 2400: Sounds a note.

Line 2500: Converts ASCII code into a character. Line 2600: Finds a character in

the middle of a string. Line 2700: Converts a character into ASCII code.

Line 2800: Clears the screen. Line 2900: Sets up an array with 26 spaces.

Line 3000: Sets up an array with 60 spaces.

Line 3100: Converts a single letter to ASCII code.

LISTING

1000 REM SPECTRUM SUBROUTINES 1100 LET W=32:LET D=22:LET F=8:

GOSUB 1200: RETURN

1200 CLS:PRINT AT 8,0;:RETURN 1300 LET R=INT(N*RND+1):RETURN

1400 FOR Z=1 TO 200*T:NEXT Z:RETURN 1500 IF INKEY\$<>"" THEN GOTO 1500

1510 IF INKEYS="" THEN GOTO 1510 1520 LET GS=INKEYS:RETURN

1600 LET K\$=INKEY\$:RETURN 1700 PRINT AT Y,X;:RETURN

1800 PRINT TAB(X);:RETURN 1900 PRINT CHR\$(11);:RETURN 2000 PRINT CHR\$(9);:RETURN

2000 PRINT CHR\$ (9); RETURN 2100 GOSUB 2200:LET K=VAL("0"+K\$); RETURN

2200 LET K#=""

2210 GOSUB 1500: IF CODE (G\$)=13 THEN
PRINT: RETURN

2220 IF CODE(G\$)<>12 THEN LET K\$=K\$+G\$:
PRINT G\$;:GOTO 2210
2230 IF LEN(K\$)=0 THEN GOTO 2210

2240 PRINT CHR\$(8);".";CHR\$(8);: LET K\$=K\$(TO LEN(K\$)-1);GOTO 2210

LET K\$=K\$ (TO LEN(K\$)-1):GOTO 22 2300 CLS:RETURN 2400 BEEP 1 7/10-4:RETURN

2400 BEEP .1,Z/10-6:RETURN 2500 LET R\$=CHR\$(R+64):RETURN 2600 LET P\$=K\$(P):RETURN

2500 LET P\$=K\$(P):RETURN 2700 GDSUB 2500:LET A=CODE(P\$):RETURN 2800 CLS:RETURN

2900 DIM V\$(26):RETURN 3000 DIM W\$(60,26):RETURN

3100 LET GG=CODE (G\$): RETURN

BBC

Line 1000: Names the subroutine library which follows. Line 1100: Puts the computer into Mode 5; sets up a blue background and sets the cursor

at row 8.
Line 1200: Clears the screen and sets the cursor at row 8.

Line 1300: Generates a random number.

Line 1400: Sets up a time delay while the computer runs through a loop 1000 times.

Line 1500: Waits to get a character from the keyboard. Line 1600: Checks keyboard to see if a key is being pressed. Line 1700: Prints at position X

and Y.
Line 1800: Prints at position X.

Line 1800: Prints at position X. Line 1900: Moves the cursor up one line.

Line 2000: Moves the cursor forward one column. Line 2100: Inputs a number as a

variable called K. Line 2200: Inputs text as a string variable called K\$.

Line 2300: Resets the screen to black and normal text. Line 2400: Sounds a note.

Line 2500: Converts ASCII code to a character.

Line 2600: Finds a character in the middle of a string. Line 2700: Converts a character

into ASCII code. Line 2800: Clears the screen. Line 2900: Sets up an array with

26 spaces. Line 3000: Sets up an array with

60 spaces.
Line 3100: Converts a single letter to ASCII code.

LISTING

1000 REM BBC SUBROUTINES

1100 MODE 5:VDU 19,0,4;0;:W=20:D=32: F=8:GOSUB 1200:RETURN

1200 CLS:PRINT TAB(0,8);:RETURN 1300 R=RND(N):RETURN

1400 FOR Z=1 TO 1000*T: NEXTZ: RETURN

1500 G\$=GET\$:RETURN

1600 K\$=INKEY\$(0):RETURN 1700 PRINT TAB(X,Y)::RETURN

1800 PRINT TAB(X);:RETURN 1900 PRINT CHR\$(11)::RETURN

2000 PRINT CHR\$(9);:RETURN 2100 INPUT K:RETURN

2200 INPUT K\$:RETURN 2300 MODE 7:RETURN

2400 SOUND 1,-15, Z,2:RETURN 2500 R\$=CHR\$(R+64):RETURN

2600 P\$=MID\$(K\$,P,1):RETURN

2700 GOSUB 2600: A=ASC (P\$): RETURN 2800 CLS: RETURN

2900 DIM V\$ (26) : RETURN

3000 DIM W\$ (60): RETURN 3100 GG=ASC (G\$): RETURN

ELECTRON

Line 1000: Names the subroutine library which follows. Line 1100: Puts the computer into Mode 5; sets up a blue background and sets the cursor

down at row 8 Line 1200: Clears the screen and sets the cursor down at row 8.

Line 1300: Generates a random number. Line 1400: Sets up a time delay

while the computer runs through a loop 1000 times. Line 1500:

Waits to get a character entered by the user. Line 1600: Checks to see if the keyboard is being pressed. Line 1700: Prints at position X

Line 1800: Prints at position X.

Line 1900: Moves the cursor up one line.

Line 2000: Moves the cursor forward one column. Line 2100: Inputs a number as a

variable called K Line 2200: Inputs text as a string variable called KS.

Line 2300: Resets the screen to black and normal text size. Line 2400: Sounds a note.

Line 2500: Converts ASCII code to a character Line 2600: Finds a character in

the middle of a string. Line 2700: Converts a character into ASCII code.

Line 2800: Clears the screen. Line 2900: Sets up an array with

26 spaces. Line 3000: Sets up an array with 60 spaces.

Line 3100: Converts a single letter to ASCII code.

LISTING

1000 REM ELECTRON SUBROUTINES

1100 MODE 5: VDU 19, 0, 4:0; : W=20: D=32: F=8:GOSUB 1200:RETURN

1200 CLS: PRINT TAB(0.8):: RETURN 1300 R=RND(N): RETURN

1400 FOR 7=1 TO 1000*T: NEXTZ: RETURN

1500 G\$=GET\$: RETURN

1600 K\$=INKEY\$(0):RETURN 1700 PRINT TAR(X, Y)::RETURN

1800 PRINT TAB(X)::RETURN 1900 PRINT CHR\$(11);:RETURN

2000 PRINT CHR\$ (9) :: RETURN

2100 INPUT K: RETURN 2200 INPUT K\$: RETURN

2300 MODE 6: RETURN 2400 SOUND 1.-15.Z.2: RETURN

2500 R\$=CHR\$ (R+64) : RETURN 2600 P\$=MID\$(K\$,P,1):RETURN

2700 GOSUB 2600: A=ASC (P\$): RETURN 2800 CLS: RETURN

2900 DIM V\$ (26) : RETURN 3000 DIM W\$ (60) : RETURN

3100 GG=ASC (G\$) : RETURN

VIC20

Line 1000-Names the subroutine library which follows. Lines 1100-1110: Set up the area of the screen used for text and put the cursor at row 8. Lines 1200-1230: Clear the screen and set cursor at row 8 Line 1300: Generates a random number.

Line 1400: Sets up a time delay, Lines 1500-1510: Wait for a character entered by the user. Line 1600: Checks to see if the

keyboard is being pressed. Lines 1700-1730: Starting with the cursor at the top left move down Y rows and across X

columns to print at position X and Y. Lines 1800-1810: Print at posi-

tion X. Line 1900: Moves the cursor up

one line. Line 2000: Moves the cursor forward one column.

Line 2100: Inputs a number as a variable called K. Line 2200: Inputs text as a

string variable called K\$. Line 2300: Clears screen and sets the cursor back at starting position.

Lines 2400-2420: Sound a Line 2500: Converts ASCII code

to a character. Line 2600: Finds a character in the middle of a string.

Line 2700: Converts a character into ASCII code

Line 2800: Clears the screen. Line 2900: Sets up an array with 26 spaces.

Line 3000: Sets up an array with 60 spaces. Line 3100: Converts a single

letter to ASCII code

LISTING

1000 REM VIC20 SUBROUTINES

1100 W=20: D=23

1110 F=8: GOSUB 1200: RETURN 1200 PRINT CHR\$ (147)

1210 FOR LL=1 TO 8:PRINT: NEXT LL 1220 PRINT:

1230 RETURN

1300 R=INT (RND(1) #N+1) : RETURN

1400 FOR Z=1 TO 1000*T: NEXT Z: RETURN 1500 GET G\$: IF G\$="" THEN 1500

1510 RETURN

1600 GET K\$: RETURN

1700 PRINT CHR\$ (19) 1710 IF Y>O THEN FOR LL=1 TO Y:

PRINT CHR\$(17);:NEXT LL

1720 IF X>0 THEN FOR LL=1 TO X: PRINT CHR\$(29)::NEXT LL

1730 RETURN 1800 IF X>O THEN FOR LL=1 TO X:

PRINT CHR\$(29)::NEXT LL 1810 RETURN

1900 PRINT CHR\$ (145) :: RETURN 2000 PRINT CHR\$ (29)::RETURN

2100 INPUT K: RETURN 2200 INPUT K\$: RETURN

2300 PRINT CHR\$ (147) : RETURN

2400 POKE 36878, 15: POKE 36876, Z+128 2410 FOR M=1 TO 50: NEXT M 2420 POKE 36876, 0: RETURN

2500 R\$=CHR\$ (R+64) : RETURN 2600 P\$=MID\$(K\$,P.1):RETURN

2700 GOSUB 2600: A=ASC (P\$): RETURN

2800 PRINT CHR\$ (147) : RETURN 2900 DIM V\$ (26) : RETURN

3000 DIM W\$ (60) : RETURN 3100 GG=ASC (G\$): RETURN

DRAGON

Line 1000: Names the subroutine library which follows. Line 1100: Sets up the area of the screen to be used for text and sets the cursor at row 8

Line 1200: Clears the screen and sets the cursor starting position. Line 1300: Generates a random number.

Line 1400: Sets up a time delay while the computer runs through a loop 650 times.

Lines 1500-1510: Wait to get a character entered by the user. Line 1600: Checks to see if the keyboard is being pressed.

Line 1700: Prints at position X and Y. Line 1800: Prints at position X.

Line 1800: Prints at position X. Line 1900: Moves the cursor up one line.

Line 2000: Moves the cursor forward one column.

Line 2100: Inputs a number as a variable called K.

string variable called K\$. Line 2300: Clears the screen and sets the cursor back at the starting position.

Line 2400: Sounds a note. Line 2500: Converts ASCII code to a character.

Line 2600: Finds a character in the middle of a string. Line 2700: Converts a character

into ASCII code. Line 2800: Clears the screen.

Line 2900: Sets up an array with 26 spaces. Line 3000: Sets up an array with

60 spaces.

Line 3100: Converts a single letter to ASCII code.

LISTING

1000 REM DRAGON SUBROUTINES

1100 W=32:D=16:F=8:GOSUB 1200:RETURN

1200 CLS:PRINT @ 256,"";:RETURN

1400 FOR Z=1 TO 650*T: NEXT Z: RETURN

1500 G\$=INKEY\$: IF G\$="" THEN GOTO 1500

1600 K#=INKEY#: RETURN

1700 PRINT @ 32*Y+X,"";:RETURN

1800 PRINT TAB(X);:RETURN
1900 PRINT CHR\$(94)::RETURN

2000 PRINT CHR\$(94);:RETURN 2100 INPUT K:RETURN

2200 INPUT K\$:RETURN

2400 SOUND Z,1:RETURN

2500 R\$=CHR\$(R+64):RETURN 2600 P\$=MID\$(K\$,P,1):RETURN

2700 GOSUB 2600: A=ASC (P\$): RETURN 2800 CLS: RETURN

2900 DIM V\$ (26): RETURN 3000 DIM W\$ (60): RETURN 3100 GG=ASC (6\$): RETURN

21

GAMES OF FUN: BOMBER

(III)

HINTS

To improve your chances of landing safely you must make sure to take out the tallest towers first when you attack. Every building must be destroyed before you can land.

VARIATIONS

The speed with which the spaceship crosses the screen is governed by the value of T in line 260. The speed can be doubled by changing T=.2 to T=.1 instead.

INTRODUCTION

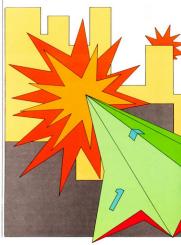
As you are an extraterestial pilot, you have very odd ideas about what are, and what are not, good flying habits. For example, whenever you visit Earth you think it perfectly alright to blast a new landing strip for yourself – no matter whether you come down in the country or in built-up areas.

The problem, as you see it, is simply that you need a smooth, wide place to touch down. Rough terrain will only foul your landing gear.

On this trip to Earth you have decided to pay a visit to the Mayor of New York. As you approach you look out of the porthole and realize that there are too many skyscrapers in the way. You will have to blast out a new landing site in order to park close to City Hall.

Of course, you cannot begin to imagine how upset the Mayor will be after you bomb his city flat. However, your biggest worry at the moment is your landing gear.

With every pass over the city release your bombs by pressing either the space bar or the number zero on your computer. Remember that you can only drop a single bomb with each run. If you press twice your next bomb will fall wide.



GAMES OF FUN: BOMBER

LINE NOTES

Lines 100-120: Set the area of the text on screen and print the title.

Lines 130-180: Create the towers and the layout of the city over which you fly. The positions of the towers change from game to game.

Lines 190-340: Print the spaceship at the top of the screen, control the speed of the ship's travel and the dropping of the bombs.

Lines 350-370: Print out messages which report how well or badly you did during the game.



LISTING

- 100 REM BOMBER
- 110 GOSUB 1100: PRINT ".....BOMBER"
- 120 LET T=2:60SUB 1400:60SUB 2800
- 130 DIM H(16) 140 LET N=6:FOR I=1 TO 14:GOSUB 1300:
 - 140 LET N=6:FOR I=1 TO 14:GOSUB 1300: LET H(I+1)=R+2:NEXT I:LET H(1)=0: LET H(16)=0
- 150 FOR Y=15 TO 6 STEP -1:LET X=3: GOSUB 1700
- 160 FOR I=1 TO 16: IF H(I)>=16-Y THEN PRINT "+";: GOTO 180
- 170 PRINT ".";
- 180 NEXT I:PRINT:NEXT Y
 190 LET A=0:LET B=1:LET D=1:LET F=0
- 200 LET X=A:LET Y=B:GOSUB 1700:
- PRINT "....":LET A=A+D
 210 IF A<1 OR A>16 THEN GOTO 230
 220 LET X=A:LET Y=B:GOSUB 1700:
- PRINT".<=>.";:60T0 260
 230 LET F=0:IF D=1 THEN LET D=-1:
- GOTO 200 240 LET X=A:LET Y=B:GOSUB 1700: PRINT "....";:LET B=B+1:LET D=1:
- IF B<14 THEN GOTO 200 250 LET T=4:GOSUB 1400:GOTO 360
- 260 LET T=.2:GOSUB 1400: IF H(A)+B=16 THEN GOTO 320
- 270 IF F=1 THEN GOTO 200 280 GOSUB 1600: IF K\$<>"." THEN
- GOTO 200 290 LET F=1:LET X=A+2:FOR Y=B+1 TO 15: GOSUB 1700:PRINT "*";:LET T=0.1: GOSUB 1400
- 300 LET Z=100:GOSUB 2400:GOSUB 1700: PRINT ".";:LET T=0.1:GOSUB 1400: NEXT Y:LET H(A)=0
- 310 GOTO 200 320 LET T=.1:FOR I=1 TO 8:GOSUB 1700: PRINT ".....":LET Z=20:GOSUB 2400
- 330 GOSUB 1400:GOSUB 1700:PRINT 340 GOSUB 1700:PRINT ".XXX.":LET T=4: GOSUB 1400
- 350 GOSUB 2800:PRINT:PRINT: PRINT "YOU.DIDN'T.MAKE.IT!": GOTO 380
- 360 GOSUB 2800:PRINT:PRINT:
 PRINT "A.SAFE.LANDING.!":PRINT
 370 PRINT "HAVE.A.NICE.DAY.!"
- 380 PRINT: PRINT: PRINT: STOP

GAMES OF FUN: STUPID FISH



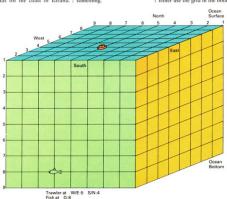
INTRODUCTION

You are the master of a North Atlantic trawler. You and the crew are out fishing for thickheads off the coast of Iceland. Thickheads are such stupid fish that they will only take food that is right in front of their nose. The only way to catch one is by dropping a baited hook right on top of it.

As master of the ship you must find the right surface location and then the right depth at which the thickheads are swimming. After each attempt your ship's sonar tells you how close you came to hooking a fish. Because it is a highly advanced machine the sonar will also tell you when you have managed to catch somethine.

Shown on the grid below is the area in which you are to fish. The distance from west to east (W/E) is numbered from 9 to 1. The range from south to north (S/N) is also marked off from 1 to 9. The depth of the seabed (D) ranges from 1 at the surface to 9 at the bottom.

With each turn that you take you enter your choice of a surface position and also the choice of depth. So that you do not end up making blind guesses it is important to keep track of each of your previous moves. You can either use the grid in the book to



GAMES OF FUN: STUPID FISH

keep track of them or else draw up your own on a separate piece of paper and use a pencil. Once you get the knack of fishing in a three-dimensional area you should be able to catch a fish in no more than four casts at the most.

With each new game the computer hides the thickheads in a new position somewhere on the

HINTS

located.

If your first cast is at position 5;5;5 you will be able to rule out three-quarters of the fishing ground with a single move.

LINE NOTES

Lines 100-130: Define the text area and put the title on the screen for a short time. Then

clear screen.

Line 140: Creates the grid system in which the fish are

Lines 150-200: Set up the questions for you to guess the location of the fish.

Line 210: Prints message reporting when you have hooked a fish.

Lines 220-310: Compare your guesses with the actual location of the fish and report the direction in which to go if you have not hooked anything.

Line 320: Loops back for another go at guessing the location.

Lines 330-340: Summary of the number of tries it took for you to catch a fish.

LISTING

- 100 REM FISH
- 110 GOSUB 1100 120 PRINT "...STUPID.FISH"
- 130 LET T=2:60SUB 1400:60SUB 2800
- 140 LET N=9:GOSUB 1300:LET A=R: GOSUB 1300:LET B=R:GOSUB 1300:
- LET C=R:LET S=0 150 LET S=S+1:PRINT:
- PRINT STR*(S);"..W/E:"; 160 GDSUB 1500:GDSUB 3100:
- LET D=GG-48: IF D<1 OR D>9 THEN GOTO 160
- 170 PRINT 64;".S/N:";
- 180 GOSUB 1500: GOSUB 3100: LET E=GG-48:
- IF E<1 OR E>9 THEN GOTO 180
- 200 GOSUB 1500: GOSUB 3100:
 - LET F=GG-48: IF F<1 OR F>9 THEN GOTO 200
- 210 PRINT G*:PRINT: IF A=D AND B=E AND C=F THEN PRINT "YOU.HOOKED.IT!":
- GOTO 330 220 PRINT "GO.":
- 230 IF A=D AND B=E THEN GOTO 300
- 240 IF E<B THEN PRINT "NORTH";
- 250 IF E>B THEN PRINT "SOUTH"; 260 IF D>A THEN PRINT "EAST";
- 270 IF DOA THEN PRINT "WEST";
- 280 IF F=C THEN GOTO 320 290 PRINT ".&.";
- 300 IF F<C THEN PRINT "DOWN";
- 310 IF FXC THEN PRINT "UP"; 320 PRINT:PRINT:GOTO 150 330 LET T=4:GOSUB 1400:GOSUB 2800:
- PRINT:PRINT: PRINT "YOU.TOOK.";STR\$(S);".GOES" 340 PRINT:PRINT "TO.HOOK.THE.FISH"
- 350 PRINT: PRINT: PRINT: STOP

GAMES OF FUN: SWISS NAVY



INTRODUCTION

Due to a clerical error in the Ministry of Transport, landlocked Switzerland discovers one day that it has bought itself a firstclass modern navy. The rest of the world is outraged. In a bid to bring the land of cuckoo clocks and dark chocolate to its senses an international squadron of bombers is sent to sink the Swiss fleet which, for some strange reason, has been hidden in the

countryside under camouflage. The Swiss Navy consists of the following ships:

- 1 Battleship (BBBB) 1 Cruiser (CCC)
- 1 Cruiser (CCC) 1 Destroyer (DD)
- 1 Destroyer (DD) 2 Patrol Boats (P) (P)

Each of the five ships is hidden on a grid for which the various squares all have row numbers followed by column letters that identify them. To bomb a position you key in the row and then the column.

Every bomb that you drop is noted on the screen. A hit is indicated by the letter of the ship that has been struck. To sink the Swiss Navy you must find and destroy all five ships using as few bombs as possible. To make things more difficult, if you bomb a mountain (M) by mistake your planes are blown up and you have to start again.



LISTING

- 100 REM NAVY
- 110 GOSUB 1100
- 120 DIM B(6,8)
- 130 DIM S\$ (6)
- 140 PRINT "....SWISS.NAVY"
- 150 FOR I=1 TO 6:FOR J=1 TO 8:
- READ B(I,J):NEXT J:NEXT I
- 160 DATA 0,1,1,1,1,0,0,0
- 170 DATA 0,0,0,0,0,0,2,0 180 DATA 3,3,0,0,0,0,2,0
- 190 DATA 0,0,0,4,0,0,2,0
- 200 DATA 5,0,0,0,0,0,0,0 210 DATA 0,0,5,0,0,0,0,4
- 220 FOR I=1 TO 6:READ S\$(I):NEXT I
- 230 DATA "*","B","C","D","P","M" 240 LET T=2:60SUB 1400:60SUB 2800:
- PRINT
 250 PRINT "...A.B.C.D.E.F.G.H":PRINT
 260 PRINT ".1.+.+.+.+.+.+":PRINT
- 260 PRINT ".1.+.+.+.+.+.+":PRINT 270 PRINT ".2.+.+.+.+.+":PRINT 280 PRINT ".3.+.+.+.+.+":PRINT
- 290 PRINT ".4.+.+.+.+.+":PRINT 300 PRINT ".5.+.+.+.+.+":PRINT 310 PRINT ".6.+.+.+.+.+.+
- 320 LET B=0:LET H=0 330 LET B=B+1
- 340 LET X=1:LET Y=14:GOSUB 1700: PRINT "...."
- 350 GOSUB 1700:PRINT "ROW."; 360 GOSUB 1500:GOSUB 3100:

GAMES OF FUN: SWISS NAVY



LET R=66-48:

IF R<1 OR R>6 THEN GOTO 360

370 PRINT G\$: ".COL.":

380 GOSUB 1500: GOSUB 3100: LET C=GG-64.

IF C<1 OR C>8 THEN GOTO 380 390 PRINT 64: "...";

400 LET P=B(R.C): IF P=0 THEN PRINT "MISS";:

BOTO 430 410 IF P>4 THEN GOTO 430

420 PRINT "HIT!":

LET B(R,C)=0:LET H=H+1

430 LET X=2*C+1:LET Y=2*R+1: GOSUB 1700: PRINT S\$ (P+1):

440 IF P<>5 THEN GOTO 470 450 LET X=1:LET Y=15:GOSUB 1700:

PRINT "HIT. MOUNTAIN. ! "::LET T=2: GOSUR 1400 460 GOTO 520

470 LET TH3: GOSLIR 1400: IF H<11 THEN **60TO 330**

480 GOSUR 2800: PRINT: PRINT: PRINT "YOU. TOOK. "; STR\$ (B); ". BOMBS"

490 PRINT: PRINT "TO, SINK, THE, NAVY" 500 PRINT: IF B<30 THEN

PRINT "..GOOD.FLYING.!": **GOTO 520**

510 PRINT "YOU.NEED.PRACTICE.!"

520 PRINT: PRINT: PRINT: STOP

LINE NOTES

Lines 100-140: Set screen area of the game and print out title.

Lines 150-230: Place the ships of the Swiss Navy on the grid.

Lines 240-340: Print the empty grid area on the screen.

Lines 350-390: Ask you to input your guesses as you search the grid for ships.

Lines 400-430: Report whether your bombs have hit or missed the targets.

Lines 440-460: Report if you have hit a mountain by mistake and have brought the game to a premature end. If so you must go back to the beginning and try again.

Line 470: Takes you back for another chance to bomb a square on the grid.

Lines 480-510: Summary of the number of bombs you used to sink the entire navy.

VARIATIONS

In order to hide the ships at the start of a new game you have to change the data in lines 160-210. As long as these numbers form a straight line, they can be placed across, up and down, or diagonally in any direction. The code for each variety of ship is as follows.

- 1 = Battleship
- 2 = Cruiser
- 3 = Destroyer 4 = Patrol Boats
- 5 = Mountains

GAMES OF FUN: HIT OR MISS



INTRODUCTION

The trajectory of shells fired from guns can be very precisely determined. As captain of a field gun it is your task to learn to fire with 100% accuracy.

In this contest a spotter in an observation balloon radios the range of the most promising targets to you. You must destroy as many as possible with the ten rounds of ammunition that you have been given.

have been given. Your gun has a range from 500m to 15,000m. The greatest distance can be obtained with an angle of 45° although the barrel can be elevated to 90° and crank-el down as low as 1,00° at 1,00° for by raising and lowering the barrel to the nearest degree. However, if you shoot at an angle of 90° the shell will go straight up—and straight back down again. You and the gun crew will all be wiped out!

Targets are scattered at random 500m intervals throughout the range. You score a hit anywhere within 250m of a bullseye. Reports of a miss, whether too far or too short, are accurate to the nearest 250m.

Due to the very high cost of shells you only have one chance to hit each target. But as long as your angle of fire is correct you will never miss.

LISTING

100 REM HITMISS

110 GOSUB 1100:LET H=0

120 PRINT "HIT.OR.MISS":PRINT:PRINT

130 PRINT "YOU. HAVE. 10. SHELLS"

140 LET H=0:FOR G=1 TO 10 150 LET N=29:GOSUB 1300:

50 LET N=29:60SUB 1300: LET R=500#R+500

160 LET T=3:GOSUB 1400 170 GOSUB 2800:PRINT:

PRINT "TARGET.AT.";R;".MTRS"

180 PRINT:PRINT "ANGLE.OF.FIRE?.";;

GOSUB 2100:IF K>90 OR K<1 THEN

BOTO 170

190 PRINT: PRINT: PRINT

200 IF K=90 THEN PRINT: PRINT "*BOOM*.YOU.ARE.DEAD!":

GOTO 270 210 LET D=15000#SIN(6.28#K/180)

220 LET E=250*INT((D-R)/250+.5): IF ABS(E)>250 THEN GOTO 250

230 PRINT:PRINT:PRINT "**BOOM**"
240 LET H=H+1:GGT0 270

250 LET M\$="SHORT": IF E>O THEN LET M\$="FAR"

260 PRINT:PRINT "THAT.WAS.";ABS(E);
".M":PRINT:PRINT "TOO.";M\$
270 NEXT G

270 NEXT G 280 PRINT:PRINT "OUT.OF.SHELLS.!" 290 LET T=3:GOSUB 1400:GOSUB 2800

300 IF H>5 THEN LET C\$=".GREAT.SHOOTING.!": GOTO 330 310 IF H>2 THEN LET

C\$="....NOT.BAD.!": GOTO 330 320 LET C\$="YOU.NEED.PRACTICE.!"

330 PRINT:PRINT:PRINT C\$
340 PRINT:PRINT:PRINT "YOU.HIT.":H:

".TARGETS"
350 PRINT:PRINT "OUT.OF.A.POSSIBLE.10"

360 PRINT:PRINT:PRINT:STOP

GAMES OF FUN: HIT OR MISS

HINTS

If you keep a list of the targets, the angles at which you set the gun barrel, and the reports of where the shells landed you should soon see the pattern of your shots. Your accuracy should improve quickly making a perfect score easy to get.

VARIATIONS

Up to 45° your shells will have a fairly flat trajectory. Try hitting targets using only barrel settings between 45° and 89° to lob your shots high. Which high angles give the same range as low angles?

LINE NOTES

Lines 100-130: Set the text area of the screen and print the title and first message.

Lines 140-170: Select the location of the target and print it on the screen.

Line 180: Asks you to guess the correct angle of fire.

Line 200: Reports the result of a shot fired at 90°.

Lines 210-250: Calculate where each shot lands and if it is close enough to the target to be a hit.

Line 260: Reports shots that missed and by how much.

Line 270: Moves on to the next target.

Lines 280–350: Summary of your success rate after taking ten shots.



GAMES OF FUN: FRUIT MACHINE



INTRODUCTION

After a long night in a Nevada casino you find that you have nothing more than ten dollars to your name. That and the shirt on your back! Desperate to win back your money you decide to stake everything playing a luckychance fruit machine.

Bets range in size from one dollar to 30 dollars. If you win 101 dollars you clean out the machine. You are broke, of course, when it takes the last of your money.

The combinations of fruit are generated randomly; there is no pattern to detect. The machine pays out on the following fruit combinations:

1×bet with	Cherry	Any Fruit	Any Fruit
3×bet with		Same Fruit	Any Fruit
4×bet with	Same Fruit	Same Fruit	Same Fruit

With any other fruit combinations you lose your bet to the machine.

HINTS

Keep your bets small when your purse is small. As it grows raise them in size so that you stand a chance of beating the fruit machine.

LISTING

- 100 REM FRUIT
- 110 DIM F\$(4)
 120 FOR I=1 TO 4:READ F\$(I):NEXT I
- 130 DATA "LEMON", "ORANGE", "CHERRY",
- 140 LET M=10 150 GOSUB 1100
- 160 PRINT "...FRUIT.MACHINE"
- 170 LET T=2:GOSUB 1400 180 GOSUB 2800:PRINT:
- PRINT "..HOW.MUCH.DO.YOU"
- 190 PRINT:PRINT "..WANT.TO.BET?.";: GOSUB 2100
- 200 IF K<1 OR K>30 OR K<>INT(K) OR K>M THEN GOTO 180
- 210 LET T=2:GOSUB 1400:LET N=4: GOSUB 1300:LET R1=R:GOSUB 1300:
- LET R2=R:LET R3=R
 220 PRINT:PRINT:PRINT F\$(R1);".";
- F\$(R2);".";F\$(R3)
- 230 IF R1=R2 THEN GOTO 250 240 LET Ms="LOSE":LET M=M-K:GOTO 270
- 250 IF R2<>R3 THEN LET M\$="WIN.\$"+STR\$(3\$K):LET M=M+3\$K:
 - GOTO 270 260 LET Ms="WIN.\$"+STR\$(4*K):
 - LET M=M+4*K 270 IF M\$="LOSE" AND R1=3 THEN
 - LET M=="LUSE" AND R1=3 THEN
 LET M=="WIN.\$"+STR\$(K):
 LET M=M+2*K
 - 280 PRINT:PRINT "..YOU.";M\$;"!":
 - 290 IF M=0 THEN PRINT "SORRY.YOU'RE.BROKE!":
- GOTO 320 300 IF M>100 THEN
- PRINT "YOU.BROKE.THE.BANK!"
 310 PRINT "YOU.NOW.HAVE.\$"; M:LET T=3:
 GOSUB 1400:IF M<=100 THEN
 SOTO 170
 - 320 PRINT: PRINT: STOP

*For Spectrum change line 110 to read: DIM F\$(4.6)

GAMES OF FUN: FRUIT MACHINE

VARIATIONS

You can vary the payout made by the machine if you wish. At the moment the odds give a zero-profit game; in the long run you will lose a much as you wiin. If you change \$TR\$63 'K', to \$TR\$64 'K', and \$M\$84 'A' ST.

TR\$64 'K', and MASSA' ST.

TO TR\$65 'K', and T

LINE NOTES

Lines 100-130: Read in the data; in this case the four fruit used by the fruit machine to create different betting combinations.

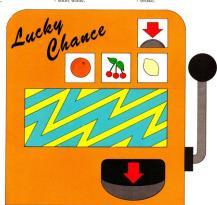
Line 140: Sets the size of your purse when you start to play the fruit machine.

Lines 150–170: Set the text area of the screen and print the title of the game. This is held on screen a short while Lines 180-200: Ask you to choose and enter your bet.

Lines 210-220: Create a random combination of the four fruit.

Lines 230-270: Work out the payoff if you win, or your loss if you don't.

Lines 280-310: Report if you have won or lost, how much money you are left with and, finally, if you or the machine are broke.



GAMES OF FUN: ZIGZAG GOLF



INTRODUCTION

This is a game of computer golf in which only one kind of shot may be used - the ZigZag Shot! You play the ball by selecting

the height of its flight (zig)from 0 to 9. The distance it travels (zag) is also a number chosen from 0

As you watch the flight of the ball you will notice a peculiar thing - it must stop directly at the hole in order to sink in. Hard shots will skip right past if you misjudge the distance. You must play nine holes to

complete a round, taking as few strokes as possible. The lowest score at the end of a round wins the game. With practice you should be able to finish a round of nine holes in much less than 27 shots.

HINTS

When playing a hole that lies especially near and toward the top of the screen, it can be easier to score by zigging to it directly while entering nil for the zag part of the shot. If you do the reverse the ball will travel toward the bottom of the screen.

Remember too that the size of the green is defined so that the hole is never more than ten units away in a straight line from your starting position.

Lines 100-120: Set up text area and print title.

the green, display the hole number being played and show the position from which you start.

Line 160: toward which you are playing.

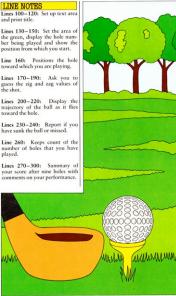
guess the zig and zag values of the shot.

trajectory of the ball as it flies toward the hole

have sunk the ball or missed.

number of holes that you have played.

your score after nine holes with comments on your performance.



GAMES OF FUN: ZIGZAG GOLF



LISTING

100 REM GOLF

110 GOSUB 1100 120 PRINT "...ZIGZAG.GOLF"

130 LET H=1:LET S=0

140 LET N=10: GOSUR 1300: LET A=R-1: GOSUB 1300: LET B=R-1: IE ARS(R-A) >5 OR A+RmO THEN **GOTO 140**

150 LET T=2: GOSUB 1400: GOSUB 2800: LET X=7:LET Y=1:GOSUB 1700: PRINT "HOLE."; H: LET X=1: LET Y=8:

GOSUB 1700: PRINT "#" 160 LET X=1+A+B:LET Y=B-A+B:

GOSUB 1700: PRINT "O": LET S=S+1 170 LET X=1:LET Y=13:GOSUB 1700:

PRINT "716. " . . GOSUB 2100:LET C=K 180 LET X=11:LET Y=13:GOSUB 1700:

PRINT "ZAG. ":: GOSUB 2100: LET D=K 190 LET X=1:LET Y=8:LET T=. 2:

IF C=0 THEN GOTO 210

200 FOR N=1 TO C:GOSUB 1700:PRINT ".": LET X=X+1:LET Y=Y-1:GOSUB 1700: PRINT "#":: 60SUB 1400: NEXT N 210 IF D=0 THEN GOTO 230

220 FOR N=1 TO D: GOSUB 1700: PRINT ".": LET X=X+1:LET Y=Y+1:GOSUB 1700:

PRINT "#":: GOSUB 1400: NEXT N 230 LET X=3:LET Y=14:GOSUB 1700: IF A<>C OR B<>D THEN PRINT "MISSED. !"::

60TO 150 240 PRINT "SUNK. IT. !"

250 LET T=2:60SUB 1400 260 LET H=H+1: IF H<10 THEN GOTO 140

270 GOSUB 2800: PRINT: PRINT: PRINT "YOU, TOOK, ":S: ", STROKES"

280 PRINT: PRINT: IF S<18 THEN PRINT "A.REAL.PRO.!!": GOTO 310 290 IF S<27 THEN PRINT"SHOWS. PROMISE. ! ":

60TO 310 300 PRINT "KEEP, PRACTISING, !"

310 PRINT: PRINT: STOP

GAMES OF FUN: ROADRACE



INTRODUCTION

While driving in an international rally you realize that racing really boils down to two things – good reflexes and tremendous stamina. You know you can win any race as long as you stay on course. Other drivers, with less concentration than you, will invariably drop out as fatigue wears them down. Your secret is that you know how to stay alert longer than anyone else.

In this rally the object is to stay in the race for as long as possible. The winner is the driver who travels furthest. You must keep the car between the crash barriers by using the X and Z keys to steer either to the right or to the left.

The race comes to an end after you have steered off the track for a total of ten stages in all. The distance that you have managed to travel is then flashed on the screen as a record to beat in the next race.

HINTS

Just touch the X and Z keys lightly to steer the car. If you hold them down you will oversteer and skid out of control.

VARIATIONS

Your first taste of racing was not bad – for a beginner. But if you want to try something more impressive try changing T=.4 in line 260 to T=.2 to make the car travel twice as fast.

LINE NOTES

Lines 100-120: Set up text area and print the title on the screen.



GAMES OF FUN: ROADRACE

up on the screen.

Lines 190-200: Move racing car to the left and to the right of the screen in response to your commands.

Line 210: Draws the car on the screen.

Line 220: Draws the road on the screen.

Lines 230-270: Calculate the position of the car and whether it is on or off the road. Measure when you have hit the barriers ten times and, if so, then bring the race to an end.

Lines 280-330: Report how far you have travelled before the race ended and comment on your driving ability.

LISTING

- 100 REM ROADRACE
- 110 GOSUB 1100
- 120 LET X=4:LET Y=D/2:GOSUB 1700:
- PRINT "ROADRACE"
- 130 LET T=2:GOSUB 1400:GOSUB 2800 140 LET A=8:LET B=10:LET C=0:LET G=0:
- 150 LET X1=8:LET X2=8:LET X3=8:
- LET X4=8 160 LET N=10:60SUB 1300:
- 160 LET N=10:GOSUB 1300: IF R<4 AND A>1 THEN LET A=A-1
- 170 IF R>6 AND A<13 THEN LET A=A+1
 180 LET X=B:LET Y=D-5:GOSUB 1700:
 PRINT".":
- 190 GOSUB 1600: IF K#="Z" AND B>2
- THEN LET B=B-1 200 IF K*="X" AND B<18 THEN LET B=B+1
- 210 LET X=B:LET Y=D-4:GOSUB 1700: PRINT "O";
- 220 LET X=A:LET Y=D-1:GOSUB 1700: PRINT "*....*";
- 230 PRINT 240 LET X1=X2:LET X2=X3:LET X3=X4: LET X4=A:
- IF B>A AND B<A+5 THEN GOTO 260
- 250 LET Z=50:GOSUB 2400: LET C=C+1
- 260 LET G=G+1:LET T=.4:GOSUB 1400: IF C<10 THEN GOTO 160
- 270 LET T=2:60SUB 1400:60SUB 2800 280 PRINT:PRINT:
- PRINT ".YOU.COMPLETED."; STR*(G)
 290 PRINT: PRINT ".STAGES.OF. THE.RACE"
- 290 PRINT:PRINT ".STAGES.OF.THE.RACE" 300 PRINT:PRINT 310 IF G<40 THEN
- PRINT ".YOU.NEED.LESSONS.!":
- 320 IF G<100 THEN PRINT "..NOT.BAD.FOR":PRINT:
- PRINT "..AN.AMATEUR!":GOTO 340
 330 PRINT "LOOK.OUT.FITTIPALDI!"
- 340 PRINT: PRINT: PRINT: STOP

GAMES OF FUN: SQUASH



INTRODUCTION

This is a game of squash with a difference. You play without a partner and you don't have to run around and exhaust yourself. However, you have only one chance to score. If you make a single mistake the game is over and you are out.

As you play you score one point each time you hit a side wall and two points whenever you hit the back wall. Your score is displayed at the end of the game. If you organise a tournament with friends, then the high score wins.

You move the racket back and fourth with the X and Z keys of the keyboard. Tap the keys lightly rather than hold them down or else you will loose control of the racket.

VARIATIONS

You can double the speed of the ball by changing the value of T=.2 in line 350 to T=.1 instead.

LINE NOTES

Lines 100-130: Define area of the text on the screen, print title and keep it in view for a short time.

Lines 140-150: Establish area

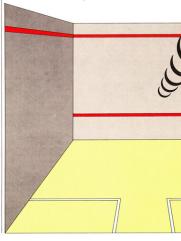
of the squash court and print it on the screen.

Lines 160-170: Print the racket and ball on screen.

Lines 180-290: Move ball around the court in response to being hit and compile the score that is attained. Line 300: Reports when a ball is missed.

Lines 310-320: Report the score that was reached, and end the game.

Lines 330–350: Move racket in response to commands. Control speed of ball.



GAMES OF FUN: SQUASH



- 100 REM SQUASH
- 110 GOSUB 1100:LET P=0
- 120 PRINT "....SQUASH"
- 130 LET T=2:60SUB 1400:60SUB 2800
 - 140 LET P=0:LET X=1:LET Y=1:
 - 150 FOR Y=2 TO 15:LET X=1:GOSUB 1700:
- PRINT "#";:LET X=18:GOSUB 1700: PRINT "#";:NEXT Y 160 LET A=4:LET B=14:LET X=A:LET Y=B:
- 160 LET A=4:LET B=14:LET X=A:LET Y=B: GOSUB 1700:PRINT "O";:LET C=1: LET D=-1:LET E=6
- 170 LET F=14:LET X=E:LET Y=F:
- GOSUB 1700:PRINT ".XXX."
 180 LET X=A:LET Y=B:GOSUB 1700:
- PRINT ".";:LET A=A+C:LET B=B+D: LET X=A:LET Y=B
- 190 GOSUB 1700
- 200 PRINT "D";
- 210 IF A<>17 THEN GOTO 230 220 LET C=-1:LET P=P+1:LET Z=40:
 - GOSUB 2400
 - 230 IF A<>2 THEN GOTO 250 240 LET C=1:LET P=P+1:LET Z=40:
- GOSUB 2400 250 IF B<>2 THEN GOTO 270
- 260 LET D=1:LET P=P+2:LET Z=60: 60SUB 2400
- 270 IF B<14 THEN GOTO 330
- 280 IF ABS (A-E-2)>1 THEN GOTO 300 290 LET D=-1:LET P=P+2:LET Z=60:
- GOSUB 2400:GOTO 330 300 LET X=6:LET Y=4:GOSUB 1700: PRINT "MISSED.!":LET Z=20:
- GOSUB 2400 310 LET X=4:LET Y=6:GOSUB 1700:
- PRINT "POINTS.=.."; STR\$(P)
 320 LET T=2:60SUB 1400:60SUB 2800:STOP
 330 GOSUB 1600:IF K\$="7" AND E>1
- THEN LET E=E-1
 340 IF K\$="X" AND E<14 THEN LET E=E+1
 350 LET X=E:LET Y=14:GOSUB 1700:
- 350 LET X=E:LET Y=14:GOSUB 1700: PRINT ".XXX.":LET T=.2:GOSUB 1400: GOTO 180

GAMES OF FUN: XMAS



INTRODUCTION

A lightning strike at Father Christmas' workshop has completely fouled up delivery schedules for presents this year. At the last meeting of the board of directors, it was obvious to all that it would be impossible for him to visit every home and find the time to climb down the

chimney to deliver the presents in person.

Instead a new method of delivery was suggested. This year Father Christmas will fly fast and low dropping gifts directly into chimney pots as he sails by without stopping. He hopes

there will be few breakages. In order to improve his aim he is making a few practice runs. Your task is to help him perfect his delivery technique. The object of this exercise is to get all of the presents into the chimneys. Pressing the space born or the zero on some machines releases one gift as he flies over a chimney. Remember that there are ten

presents in each sack - and only

one chance to get each of them to

its target.

HINTS

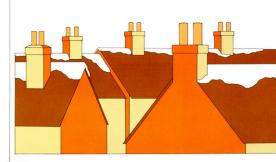
Because Father Christmas is flying so fast, it is important for each present to be dropped with a lot of lead distance. Otherwise you will probably overshoot the targets.

VARIATIONS

To make this exercise more difficult you can speed up the sled by changing T=.2 to T=.1 in line 190. This will double the speed of its travel.

INE NOTES

Lines 100-130: Set up text area, then print and hold title on screen for a few moments.



GAMES OF FUN: XMAS

Line 140: Sets up counter for the number of passes made by the sled and stops at ten.

Lines 150-170: Print the various positions of the chimney on the screen

Lines 180-240: Move the sled back and forth across the screen at a steady speed and fixed height.

Lines 250-290: Print the dropped presents on the screen and show the paths of their fall.

Lines 300-330: Report whether a dropped parcel hits or misses its target chimney and move program on to the next pass at the chimneys.

Lines 340-380: Print messages on the screen informing players how many presents went into the chimneys and comment on the



- 100 REM YMAS
- 110 GOSUB 1100:LET H=0
- 120 PRINT ".....XMAS"
- 130 LET T=2:60SUB 1400
- 140 LET H=0:FOR S=1 TO 10:GOSUB 2800 150 LET X=2:LET Y=14:GOSUB 1700:
- PRINT "#############":
- 160 LET N=10:60SUB1300:LET R=R+4 170 LET X=R:FOR Y=12 TO 14:60SUB 1700:
- PRINT " * . . * " : : NEXT Y 180 LET A=0:LET B=1:LET P=0
- 190 LET T=. 2: GOSUB 1400: LET A=A+1: LET X=A:LET Y=B:GOSUB 1700:
- PRINT ".<*>"
- 200 IF X=17 THEN GOTO 310 210 IF P=1 THEN GOTO 250
- 220 GOSUB 1600: IF K\$<>"." THEN GOTO 240
- 230 LET P=1:LET C=A:LET D=B:LET N=1
- 240 GOTO 190 250 LET C=C+1:LET D=D+N:LET N=N+1: LET X=C:LET Y=D-1:60SUB 1700:
- PRINT "+": 260 IF (X=R OR X=R+3) AND Y>11 THEN
- **60TO 310** 270 IF Y<14 THEN GOTO 190
- 280 IF X=R+1 OR X=R+2 THEN GOTO 300 290 GOTO 310
- 300 LET X=8:LET Y=5:GOSUB 1700: PRINT "BULLSEYE!": LET H=H+1:
- **60TO 320** 310 LET X=6:LET Y=5:GOSUB 1700:
- PRINT "MISSED!" 320 LET T=2:GOSUB 1400:NEXT S
- 330 LET T=2:GOSUB 1400:GOSUB 2800 340 PRINT: PRINT: PRINT "YOU. GOT. ":
- STR# (H); ". PRESENTS" 350 PRINT: PRINT:
- PRINT "OUT. OF. A. POSSIBLE. 10"; PRINT: PRINT
- 360 IF H>4 THEN GOTO 380 370 PRINT "NOT. MUCH. FOR. XMAS": PRINT:
- PRINT "...THIS.YEAR.!": 60TO 390
- 380 PRINT "APPLY.FOR.A.JOB.AT": PRINT: PRINT "THE.NORTH.POLE.!"
- 390 PRINT: PRINT: PRINT: STOP

GAMES OF FUN: ZOMBIES



INTRODUCTION

You have just been abandoned at the lost Amazonian ranch of Count Morue, the mad big game hunter. As luck would have it, the only game he really enjoys hunting is human. He sets you free on the ranch and, with a half day's head start, sends four slave zombies out to track you down. Your goal is to stay alive.

The ranch is surrounded on all sides by impenetrable swamp so there is no way of escaping from here. Pools of quicksand abound as well. Your only chance to survive is to lure the zombies (Z) into the quicksand (while avoiding it yourself) before they can catch you.

Your position on the map of the ranch is shown by an [41]. With each turn you can move by pressing any one of the keys numbered from 1 to 9. The arrows on the grid show the direction in which each number will take you. When you press 5 you stay in the same spot. It takes very careful planning to out-manoeuver the zombies.

$$\begin{array}{c|c}
 & 1 & 2 & 3 \\
\hline
 & 4 & 5 & 6 \\
\hline
 & 7 & 8 & 9
\end{array}$$

LISTING

100 REM ZOMBIES 110 GOSUB 1100

120 PRINT "....ZOMBIES"

130 LET T=2:GOSUB 1400:GOSUB 2800

150 FOR I=1 TO 13:

NEXT I

170 DIM Q(8,2) 180 DIM Z(4,2)

190 FOR I=1 TO 4:LET N=16:GOSUB 1300: LET Z(I.1)=R+1:LET N=13:

LET Z(I,1)=R+1:LET N=13: GOSUB 1300:LET Z(I,2)=R+1 200 LET X=Z(I,1):LET Y=Z(I,2):

GOSUB 1700:PRINT "Z"; 210 NEXT I

220 FOR I=1 TO 8:LET N=16:GOSUB 1300: LET Q(I,1)=R+1:LET N=13:

GOSUB 1300:LET Q(I,2)=R+1
230 LET X=Q(I,1):LET Y=Q(I,2):
GOSUB 1700:PRINT "#";

HINTS

The zombies always come toward you by the most direct route possible. They are not aware, for example, that quicksand is dangerous.

LINE NOTES

Lines 100–130: Set area of text

Lines 140-160: Print area of ranch within which you and zombies can move.

Lines 170-210: The starting positions of the zombies, which are randomly chosen, are shown.

Lines 220-240: The randomly chosen quicksand traps are placed on screen.

Line 250: Your own position is placed on screen.

Lines 260–280: Your choice of move is registered.

Lines 290–370: Check your new position and report if you hit quicksand, strayed into the swamp or were caught by zombies.

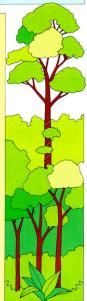
Lines 380–460: Move zombies in response to your own move so that they steadily close in around you.

Lines 470-490: Announce your win and end the game.



GAMES OF FUN: ZOMBIES

- 240 NEXT I 250 LET N=16: GOSUB 1300: LET A=R+1: LET N=13:GOSUB 1300:LET B=R+1: LET X=A:LET Y=B
- 260 GOSUB 1700: PRINT "H"::LET N=4 270 GOSUB 1500: GOSUB 3100: LET G=GG-48:
- IF 6<1 OR 6>9 THEN 60TO 270 280 LET X=A:LET Y=B:60SUB 1700: PRINT ".":
- 290 LET D=INT((G-4)/3):LET C=G-3*D-5: LET A=A+C:LET B=B+D
- 300 IF A=1 OR A=18 OR B=1 OR B=15 THEN GOTO 320
- 310 GOTO 330 320 LET MS="YOU'RE. IN. THE. SWAMP":
- **GOTO 480** 330 LET F=0:FOR I=1 TO 8: IF A=Q(I.1) AND B=Q(I.2) THEN
- LET F=1 340 NEXT I: IF F=1 THEN LET MS="YOU'RE. IN. QUICKSAND":
- IF F=1 THEN GOTO 480 350 FOR I=1 TO 4: IF A=Z(I,1) AND B=Z(I,2) THEN LET F=1
- 360 NEXT I: IF F=1 THEN LET MS="YOU'RE.CAUGHT!": IF F=1 THEN GOTO 480
- 370 LET X=A:LET Y=B:GOSUB 1700: PRINT "H";:LET Z=50:GOSUB 2400: LET T=0.5:60SUB 1400
- 380 FOR I=1 TO 4: IF Z(I,1)=0 THEN GOTO 460
- 390 LET T=0.5:GOSUB 1400:LET X=Z(I.1): LET Y=Z(I,2):GOSUB 1700: PRINT ". "::LET F=0
- 400 LET X=X+SGN(A-X):LET Y=Y+SGN(B-Y): LET F=0:FOR J=1 TO 8: IF X=Q(J,1) AND Y=Q(J,2) THEN LET F=1
- 410 NEXT J: IF F<>1 THEN GOTO 430 420 LET N=N-1:LET Z(1,1)=0:LET Z=35: GOSUB 2400: GOTO 460
- 430 GOSUB 1700:PRINT "Z":LET Z(I,1)=X: LET Z(1,2)=Y:LET Z=20:GOSUB 2400
- 440 IF YEA AND YER THEN LET MS="CAUGHT!": GOTO 480
- 450 GOSUB 1700:PRINT "Z":LET Z(I,1)=X: LET Z(1,2)=Y
- 460 NEXT I: IF N>O THEN GOTO 270 470 LET M\$="YOU'VE. MADE. IT. !" 480 LET X=1:LET Y=16:60SUB 1700:
- PRINT MS: 490 LET T=4:GOSUB 1400:GOSUB 2800:
 - PRINT: PRINT: STOP



THINKING GAMES: PYRAMID RACE



INTRODUCTION

Here is a chance to impress your computer buff friends by beating a computer in a race of mental agility. The winner (who is you) is the first person to determine the final digit that will appear at the top of a number pyramid.

Ten single numbers are entered into the bottom row of the pyramid. When the race begins, each pair of digits is added

together and entered into the boxes on the row immediately above. For example, on the left of the diagram below the numbers 1 plus 8 would be added together and 9 would appear in the row above.

If the sum is a number with more than one digit, then these two digits are also added and the result entered into the box. 50, for instance, 6 plus 7 would equal 13. We would therefore add the 1 to the 3 and enter 4 in

the empty box.
To demonstrate your 'superhuman' powers of calculation, enter the program and show your friends a trial run. Use any choice of single digit numbers in the bottom line. Then tell them

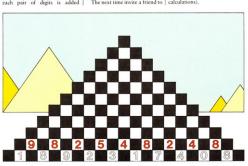
that you will now race the computer in a test of mental agility. The next time invite a friend to

enter any ten digits, one by one. When you give the word 'go' you will be able to amaze everyone by beating the computer hands down. Here is how it is done.

down. Free is now it is done.

As your friend enters the bottom line of figures, note the fourth and seventh from the left (in the diagram the 2 and the 7), Add them together (giving 9), then multiply by three (giving 27), how take the two numbers on either end of the bottom line of the pyramid (in this case 1 and 8) and add them to your last to the 6 to get 9. This is your final prediction.

Remember not to make it look too easy. Ask your friends, for instance, to double check the numbers they entered (this gives you time to finish your own calculations).



THINKING GAMES: PYRAMID RACE

LINE NOTES

the pyramid.

Line 110: Sets up the dimensions of an array to hold ten variables.

Lines 120-200: Print out the shape of the pyramid.

Lines 210-270: Ask that ten single digits be input and then print them on the bottom line of

Lines 280-290: Start the race.

Lines 300-370: Add each pair of digits and print the total on the line above the last full line.

Lines 380-400: Print a message to the screen and stop the program.



THE SHORTCUT

- (1) Add the digits in box 4 and box 7 = (2) Multiply the answer by 3 = 2
- (2) Multiply the answer by 3 = 27 (3) Add the digits in box 1 and box 10 to result in (2) = 36
- (4) Take the answer in (3) and add the two digits – you now have the final result = 9

- 100 REM PYRAMID
- 110 DIM X(10)
- 120 GOSUB 1100 130 LET N=0:LET C=9
- 140 FOR L=1 TO 10
- 150 FOR A=C TO C+N STEP 2
- 160 LET X=A:LET Y=L:GOSUB 1700: PRINT "#."
- 170 NEXT A
- 180 LET C=C-1
- 190 LET N=N+2
- 200 NEXT L 210 LET X=0:LET Y=14:GOSUB 1700:
- PRINT"ENTER. 10. DIGITS"
- 220 FOR N=1 TO 10
- 230 GOSUB 1500:LET K\$=G\$:LET P=1:
- 240 IF Y<0 OR Y>9 THEN GOTO 230
- 250 LET X(N)=Y 260 LET X=2*N-2:LET Y=10:GOSUB 1700: PRINT G4:
- 270 NEXT N
- 280 LET X=0:LET Y=14:GOSUB 1700: PRINT "PRESS.S.TO.START"
- 290 GOSUB 1500: IF G\$<>"S" THEN GOTO 290
- 300 FOR L=9 TO 1 STEP -1
- 310 LET T=2:60SUB 1400 320 FOR N=1 TO L
- 330 LET X(N)=X(N)+X(N+1) 340 IF X(N)>9 THEN
- LET X(N)=X(N)-9*INT(X(N)/10)
- 350 LET X=8-L+2*N:LET Y=L:GDSUB 1700: PRINT :X(N):
- 360 NEXT N
- 380 LET X=0:LET Y=14:GOSUB1700
- 390 PRINT "THE.END.RESULT.IS."; X(1)
- 400 STOP

THINKING GAMES: DUNGEON



INTRODUCTION

Deep underground, below the castle of an evil magician, a princess is kept imprisoned. The dungeon, as can be seen, consists of 81 cells, each communicating with the others as shown in the plan. There are also four outer doors A. B. C and D.

The princess is at present outside door D. To escape she has to zigzag through every cell and exit by door C to freedom. The magician has promised to release her if she can do this safely.

However, while she is doing this a hideous monster is also visiting every cell, except that he will enter the dungeon by door A and will exit by door B. The monster will also zig-zag through the cells, but his route will be at right-angles to that of the prin-

cess. In order to give her a sporting chance the magician has agreed that both the princess and the monster will each move an agreed number of cells in turn. These numbers, which can be amything from a single cell to a maximum of nine, are to be chosen by the princess. The princess will move first, then the monster, and so on alternately until she escapes or is captured. She is can be the search of the princes will be searched to the princes will be searched be searched to the princes will be searched to

 The monster, on its turn, passes through the cell occupied by the princess.

2) The princess passes through the cell occupied

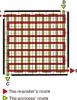
by the monster.

3) They both end up in the same cell.

For example, if the princess were to choose a move of one cell

were to choose a move of one cell for herself and nine cells for the monster, she would be captured at once. She would enter into the top right cell, and he would reach that same cell at the end of his first move. So, clearly, this is not a good choice.

If you could advise her, which numbers would you recommend her to choose?



The pri

LISTING

100 REM DUNGEON

110 DIM M(100)

120 DIM Q(100) 130 GOSUB 1100

130 GOSUB 1100 140 FOR I=0 TO 4

150 FOR J=1 TO 9

160 LET Q(18*I+J)=90+J-20*I

170 LET Q(18#I+J+9)=90-J-20#I

180 LET M(18*I+J)=2*I+10*J+1 190 LET M(18*I+J+9)=2*I-10*J+102

200 NEXT J:NEXT I

210 LET X=0:LET Y=2:GOSUB 1700 220 PRINT "..*********

230 FOR N=1 TO 9
240 PRINT "...*+++++++**

250 NEXT N

280 LET X=0:LET MT=0
280 LET X=0:LET Y=14:GOSUB 1700

290 PRINT "ENTER.PRINCESS'.MOVE" 300 GOSUB 1500:LET K#=G#:LET P=1: GOSUB 2700:LET Q=A-48

310 IF Q<1 OR Q>9 THEN GOTO 300 320 GOSUB 1700:

PRINT"ENTER.MONSTER'S.MOVE"

330 GOSUB 1500:LET K#=G#:LET P=1:
GOSUB 2700:LET M=A-48

340 IF M(1 DR M)9 THEN GOTO 330

THINKING GAMES: DUNGEON

TINTS

Of the 81 possible moves that can be made only 21 will allow the princess to escape the clutchess of the monster.

Note that she should never allow the monster to make a move of three spaces. It will result in inevitable capture no matter what number she picks as a move for herself.

LINE NOTES

Lines 110-120: Set up the arrays which will contain the princess' and monster's paths through the dungeon.

Lines 140-200: Store the screen locations of each cell in the dungeon.

Lines 210-260: Print the layout of the dungeon on the screen.

Lines 270-280: Set the initial values of the starting position of the monster and the princess.

Lines 290-340: Enter the moves that the princess and monster make.

Lines 350-470: Control the

princess' moves.

Lines 480-580: Control the monster's moves.

Lines 630-690: The routine which puts a 'win' message on the screen.

Lines 700-750: The routine which puts a 'lose' message on the screen.

Answers on page 118

350 GOSUB 1700:

PRINT "NOW. THE. MOVES. BEGIN."

360 LET T=1:60SUB 1400 370 GOSUB 1700:

PRINT "....."
380 LET K=0:LET B=0

390 LET QY=9:LET QX=1

410 FOR G=1 TO Q 420 LET PT=PT+1:LET T=0.1:GOSUB 1400

430 IF PT>81 THEN GOTO 630

LET L=Q(PT)-10*C 450 LET X=QX+2:LET Y=QY+2:GOSUB 1700:

PRINT "+"; 460 LET X=C+2:LET Y=L+2:GOSUB 1700

470 PRINT "P"; 480 IF MY=L AND MX=C THEN GOTO 700 490 LET GY=L

500 LET QX=C 510 NEXT G

520 FOR G=1 TO M 530 LET MT=MT+1:LET T=0.1:GOSUB 1400

540 IF MT>81 THEN GOTO 630 550 LET B=INT(M(MT)/10):

LET K=M(MT)-10*B 560 LET X=MX+2:LET Y=MY+2:GOSUB 1700: PRINT "+";

570 LET X=B+2:LET Y=K+2:GOSUB 1700 580 PRINT "M"; 590 IF K=QY AND B=QX THEN GOTO 700

600 LET MY=K:LET MX=B 610 NEXT 6

620 GOTO 410 630 LET T=3:GOSUB 1400:GOSUB 1200 640 LET X=0:LET Y=6:GOSUB 1700

650 PRINT"CONGRATULATIONS!.THE"
660 PRINT:PRINT"CHOICE.OF.";Q;
".FOR"
670 PRINT:PRINT"PRINCESS.AND.":M:

".FOR"

680 PRINT:PRINT"MONSTER.HAS.WORKED"

690 PRINT:STOP

700 LET T=3:GOSUB 1400:GOSUB 1200 710 LET X=0:LET Y=6:GOSUB 1700 720 PRINT"TOO.BAD.-.YOU.MET"

730 PRINT:PRINT"IN.CELL.";L;C;".AND" 740 PRINT:PRINT"THE.PRINCESS.WAS"

750 PRINT:PRINT"CAUGHT.!" 760 PRINT:STOP

THINKING GAMES: SUM TROUBLE



INTRODUCTION

The four members of SUMS (Society of Unusual Mathematics) met the other evening. Each put forward a problem for the other three to solve. How many can you answer?

Basil Byte's Problem
"I have a piece of paper," said

Basil, "on which is written a four-digit square number. If I tear the paper in two each of the pieces now also has written on it a two-digit perfect square.

For example, if I had written 3481 (the square of 59) I would find that although I could tear it to produce 81 (the perfect square of 9), the remaining 34 is not a perfect square.

If none of the digits can be zero, what number could I have written on the paper?"

Cynthia Cube's Problem

"I too have a four-digit number," said Cynthia. "If I multiply this number by itself, the resulting eight digit number still ends in the same original four digits. What is the number?"

Cynthia's problem is best tackled by testing all of the four-digit numbers from 3163 to 9999 (all of which give an eight-digit number when squared). The computer takes several minutes to run this program while it is testing the various numbers. Darrell Digit's Problem

"I happened to notice the other day that my wife's age is the same as my age – but with the digits reversed. Also, the sum of our ages is eleven times greater than the difference. I am the eldest, but what are our ages?" Ernest Euclid's Problem

"My eldest son has just taken his

last exams at school," explained Ernest. "He had to answer three papers each with 100 questions. I agreed to give him 11 pence for each correct answer, but fine him 14 pence for every question he answered wrongly.

After the exams were over we found that the amount I owed him was the same as the amount that he owed me. So no money changed hands.

How many answers did he get right in the exam and how many wrong?"

LINE NOTES

Lines 120-140: Generate a four-digit number in which the first and second, and the third and fourth digits form perfect squares.

Lines 150-160: This four-digit number is tested to see if it is itself a perfect square.

Lines 170–180: The remaining values of B and A are tested for other possible answers.

Cynthia

Lines 120 & 140: Four-digit numbers in the range 3163 to 9999 are squared.

Line 130: Positions message on

screen that the program is testing all possible values (this takes quite a long time).

Lines 150-160: When a value is found with the last four digits the same as the original number, then the answer is printed on screen.

Line 170: The next four-digit number is checked.

Line 180: The program ends.

Darrell

Lines 120-150: Values for the two digits of Darrell's age and his wife's age are introduced.

Lines 160–180: The sum and difference of the two ages is worked out and tested to see if the sum is 11 times greater than the difference. Values that agree are printed out by line 160. The remaining possible values of their ages are then tested.

Ernest

Lines 120–140: For each value of the number of right answers (R) the corresponding number of wrong answers (W) is found.

Line 130: A message showing which number is being tested is shown on screen.

Lines150-160: If the amount payable for right answers equals that payable for wrong answers then the result is printed out.

Line 170: The next value of the number of right answers is tested.

Answers on page 118

THINKING GAMES: SUM TROUBLE

- 100 REM BASIL 110 GOSUB 1100
- 120 FOR A=4 TO 9
- 130 FOR B=4 TO 9 140 LET C=100*A*A+B*B
- 150 LET D=SQR (C)
- 160 IF ABS(D-INT(D))<0.001 THEN
 PRINT C
- 170 NEXT B
- 180 NEXT A
- ... ----
- 100 REM CYNTHIA
- 110 GOSUB 1100 120 FOR I=3163 TO 9999
- 130 LET X=2:LET Y=4:GOSUB 1700:
- PRINT "TESTING.": I
- 140 LET J=I*I
- 150 LET K=J-10000*INT(J/10000)
- 160 IF K=I THEN PRINT I,J
- 180 PRINT:STOP
- 100 REM DARRELL
 - 110 GOSUB 1100
 - 120 FOR A=0 TO 9
 - 130 FOR B=0 TO 9
- 150 LET D=10*B+A
- 160 IF C+D=11*(C-D) THEN
 PRINT "AGES.ARE.";C;".AND.";D
- 170 NEXT B 180 NEXT A 190 PRINT:STOP
- 100 REM ERNEST
- 110 GOSUB 1100 120 FOR R=0 TO 300
- 130 LET X=0:LET Y=6:GOSUB 1700: PRINT "TESTING.";R
- 140 LET W=300-R 150 IF 11*R<>14*W THEN GDTD 170
 - 160 PRINT:PRINT "HE.GOT.";R;".RIGHT": PRINT:PRINT"AND.";W;".WRONG": PRINT:STOP
 - 170 NEXT R

THINKING GAMES: PRIMES



INTRODUCTION

Prime numbers are one of the oddities of our number system whose unpredictable nature has intrigued mathematicians for thousands of years! Can you spot the difference between the primes and non-primes below?

to imagine that you have 14 coins to share out. You could divide them in half with each person receiving seven coins, or you could give seven people two coins each. However, if you had only 13 coins (a prime number) there is no way that you could divide them up exactly. Remember, there must always be more

than one person sharing, and each must receive more than one coin.

So how can we tell if a number is prime? There is no easy way except by repeated trial division

unfortunately.

We can eliminate certain numbers right away however.

Obviously an even number can4) Is the year 1983 a prime?

few thousand.

Using the program, try and solve as many of the following problems as you can:

1) How many primes are there between 1 and 100?

2) How many primes are there between 100 and 200? Between 300 and 400? Do the number of primes increase, decrease or stay the same for each of these ranges?

3) The five primes 5, 11, 17, 23,

29 all have a difference of six in common. Can you find another ten primes with a common difference? (The smallest is over 100 and the difference between them is more than 200).

5 7 49 9 21 25 41 427 33 14 27 33 51

The numbers on the left are primes. They cannot be divided exactly (leaving no remainder), by any other whole number, except of course by themselves and by one. On the other hand, the composites (which is the name given to numbers that are not prime) can be divided by smaller whole numbers. For example, 14 can be divided exactly by 7 and 2.

An easy way to think of this is

not be prime as it can be divided by 2. There is one exception to this and that is 2 itself which has the distinction of being the only even prime to exist. Similarly, all numbers ending in 5 are divisible by 5. They too can be ruled out.

Here is a program that will test any number up to a million to see if it is a prime. However, the higher you go, the slower the process becomes. But it works oute well for numbers up to a

When was the last year that was a prime? When is the next?

5] Until this year, what is the longest span of consecutive nonprime years you can find? For example, there was a period of 18 consecutive years from 1913 (prime) until 1931 (prime) when every year was non-prime, although longer periods than this can be found.

s 6) How many primes can you a find that are palindromes (read

THINKING GAMES: PRIMES

the same backwards as forwards)?

7) What is the biggest prime under one million?

LINE NOTES

Line 110: Sets up an array with 200 spaces that holds primes up to 1000.

Line 120: Sets up screen display.

Lines 130–150: Print message on screen while program tests all numbers to 999 to find the primes.

Lines 160–230: Store all primes up to 999 for future reference in the program.

Line 210: Prints number being tested.

Lines 220-230: Sort through all

odd numbers up to 999.

Lines 240–250: Request that you enter a number for testing.

Lines 260-290: Ensure that only numbers between one and a million are valid for testing.

million are valid for testing.

Lines 300-330: Report if a number is prime or not.

Lines 340-410: Subroutine that tests a number to see if it is prime.

LISTING

100 REM PRIMES

110 DIM P(200)

120 GOSUB 1100 130 LET X=0:LET Y=10:GOSUB 1700:

PRINT "**WORKING**"

140 LET NP=2:LET P(1)=2:LET P(2)=3

140 LET NP=2:

160 GOSUB 340 170 IF F>0 THEN GOTO 220

180 LET T=0.2:GOSUB 1400

190 LET NP=NP+1

200 LET P(NP)=N

210 LET X=14:LET Y=10:GOSUB 1700: PRINT :N

220 LET N=N+2

230 IF N<1000 THEN GOTO 160

240 LET X=0:LET Y=10:GOSUB1700 250 PRINT "ENTER.YOUR.NUMBER."

260 GDSUB 2100:LET N=K

270 IF N=0 THEN PRINT:STOP
280 IF N<1 OR N>999999 OR N<>INT(N)
THEN PRINT "INVALID.NUMBER":

PRINT: GOTO 250 290 GOSUB 340

300 IF F=0 THEN PRINT: PRINT "IT.IS.PRIME":GOTO 320

310 PRINT:PRINT "=.";F;".X.";N/F:
PRINT:PRINT"AND.IS.NOT.PRIME"

320 PRINT 330 PRINT: GOTO 250

340 LET F=0 350 LET S=SQR(N)

360 FOR I=1 TO NP 370 IF P(I)>S THEN LET I=NP:00TO 400

380 LET R=N/P(I) 390 IF R=INT(R) THEN LET F=R:

S90 IF R=INT(R) THEN LET LET I=NP

400 NEXT I 410 RETURN

Answers on page 118

THINKING GAMES: MADAM I'M ADAM



INTRODUCTION

Bob, Hannah, Eve, Otto and Anna are members of a very special club. Bill, Debbie, Emma, Brian and Vicky are not. Can you see why?

Perhaps if you learned that the club was called the *Palindrome Society* you'd see why. All the members' names are palindromes. They read the same spelled backwards as they do spelled forwards.

spelled forwards.

Indeed, you would probably
gather this from the sort of
bizarre conversations you would
hear at a club meeting.

"Madam I'm Adam"

"A man, a plan, a canal – Panama!"

"Note: Nine posts open in Eton"

"Egad! A base tone denotes a bad age"

In fact, any kind of palindrome attracts their interest. At the club's last meeting they turned their attention to number palindromes.

"A number is a palindrome," quoted Bob, "if it reads the same backwards as forwards." He paused and then con-

tinued. "I have two puzzles here for you to solve. One is for Hannah and Eve, and the other for Otto and Anna."

"What I want you to do," said Bob to Hannah and Eve, "is to find the three smallest palindromes that are perfect squares, but which do not themselves have square roots that are palindromes."

"Well," said Eve. "I always thought that a perfect square was a person who . . ."

"A perfect square," interrupted Bob sternly, "is the result hat is obtained when you multiply a whole number by itself. For instance, 121 is the perfect square you get when multiplying 11 by 11. It is also a palindrome. However, its square root, 11, is a palindrome too so it cannot be the answer we are after." "And, for the second puzzle, Otto and Anna have to find the smallest perfect square with an even number of digits that is also a palindrome."

Fortunately they all had access to a computer and the problems were soon solved.

Both programs need a subroutine to test that a number is a palindrome. This subroutine takes a number N and if it proves to be a palindrome comes up with a value of T=1. If it is not, then T=0. With both of their problems, Hannah, Eve, Otto and Anna used the same subroutine.

LISTING

```
100 REM PALINI
110 GOSUB 1100
120 LET C=0
130 LET Z=11
140 LET X=0:LET Y=6:GOSUB 1700:
    PRINT"TESTING. "; Z
150 LET N=Z#Z
160 GOSUB 260
170 IF T=0 THEN GOTO 230
180 LET N=Z
190 GOSUB 260
200 IF T=1 THEN GOTO 230
210 LET C=C+1
220 PRINT: PRINT"NUMBER. = . ": Z:
    PRINT: PRINT"SQUARE. = . "; Z*Z
230 LET Z=Z+1
240 IF C<3 THEN GOTO 140
250 PRINT: STOP
260 REM TEST PALINDROME
270 LET A=N
280 LET R=0
290 LET A1=INT (A/10)
```

300 LET B=10*B+A-10*A1

320 IF A>O THEN GOTO 290

340 IF B=N THEN LET T=1

310 LET A=A1

330 LET T=0

350 RETURN

THINKING GAMES: MADAM I'M ADAM

LINE NOTES

Palin 1

Line 110: Sets up text area on the screen.

Line 120: Sets starting value of the counter for the three smallest nalindromes

Line 130: Sets the starting value of the numbers being tested.

showing the number being tested.

Line 150: Finds the square of the for the next one.

number being tested.

Line 160: Sends program to a subroutine.

Lines 170 & 230: If number is not a palindrome these lines send the program to the next number for testing.

Lines 180-190: Check if the number being tested is also a palindrome.

Line 140: Prints a message Line 200: If one palindrome has been found the program keeps on testing other numbers looking



Lines 210-240: Program keeps testing numbers until all three palindromes are found.

Lines 260-350: Subroutine to test if a number is a palindrome.

Palin 2

Line 110: Sets up area of the text on screen.

Line 120: Sets limits to the initial values of the perfect squares.

Line 130: Sets starting value of the numbers being tested.

Line 140: Prints a message showing the numbers being tested.

Lines 150-170: Find perfect square of number being tested.

Line 180: Sends program to a subroutine

Line 190: Prints a message when correct answer is found.

Lines 200-210: Increase value of number being tested by one and repeat test procedure.

Lines 220-310: Subroutine that tests if the perfect square is a palindrome.

Answers on page 118

LISTING

100 REM PALIN2

110 GOSUB 1100

120 LET L1=10:LET L2=100 130 LET Q=4

140 LET X=0:LET Y=6:GOSUB 1700:

PRINT"TESTING. ";Q 150 LET N=Q*Q

160 IF NCL1 THEN GOTO 200

170 IF N>=12 THEN LET L1=100#L1: LET L2=100#L2:GOTO 200

180 GOSUB 220 190 IF T=1 THEN PRINT: PRINT"NUMBER. = . " : Q: PRINT:

PRINT"SQUARE. =. ": N: PRINT: STOP 200 LET Q=Q+1

210 GOTO 140

220 REM TEST PALINDROME

230 LET A=N 240 LET B=0

250 LET A1=INT(A/10) 260 LET B=10*B+A-10*A1

270 LET 4=41

280 IF A>O THEN GOTO 250 290 LET T=0

300 IF BEN THEN LET TE1 310 RETURN

THINKING GAMES: STEPS AND LEAPS



INTRODUCTION

The object of this game is to reverse the order of the counters with the fewest possible moves, and with the empty space always ending up at the extreme left of the box.

A piece can be moved in one of two ways. If an adjacent square is vacant it can 'step' into the empty space. On the other hand, a piece can 'leap' over another counter into the empty space beyond.

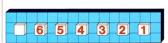
Taking the diagram below as an example, at the start of the game one would have a choice. You can either move the 6 to the vacant square or else leap the 5 over the 6 and into it.

Pieces may shuttle back and forth as many times as you like. Simply press the number of the piece you want to move and it will automatically go to the new position as long as the move is a valid one.

If you get stuck press 'S' and the computer will play the game automatically and with the least possible number of moves.

Once you have mastered the game you should be able to match the computer's score.

- 100 REM STEPS
- 110 DIM X (7) 120 LET C=6
- 130 FOR A=2 TO 7
- 140 LET X(A)=C+2-A
- 160 LET X(1)=0
- 170 LET 77=-1
- 180 GOSUB 1100 190 LET Y=14: LET X=0: GOSUB 1700: PRINT"PRESS. S. FOR. SOLUTION"
- 210 LET Y=12:LET X=0:GOSUB 1700:
- PRINT"PRESS.KEYS. 1. TO. 6" 220 LET X=0:LET Y=10:GDSUB 1700:
 - PRINT "##########
- 230 GOSUB 1500
- 240 IF 6\$="S" THEN GOTO 650 250 LET K\$=G\$:LET P=1:GOSUB 2700:
 - LET G=A-48: IF G<1 OR G>6 THEN BOTO 230
- 260 FOR B=1 TO C+1
- 270 IF G=X(B) THEN LET Q=B 280 IF X(B)=0 THEN LET R=B
- 290 NEXT B 300 IF ABS (Q-R) <3 THEN GOTO 330-
- 310 LET X=0:LET Y=10:GOSUB 1700: PRINT"INVALID. MOVE"
- 320 LET T=5:60SUB 1400:60T0 220
- 330 LET X(Q)=0 340 IFT Y(R)=6
- 350 GOSUB 380
- 360 GOTO 220 370 REM DRAWS BOARD
- 380 LET X=0:LET Y=4:60SUB 1700: PRINT ". ************ 390 LET Y=6: GOSUB 1700:
 - PRINT ". 400 LET Y=5:60SUB 1700: PRINT "....."



THINKING GAMES: STEPS AND LEAPS

410 FOR N=1 TO C+1+1 FT Y=2*N+ **GOSUB 1700** 420 IF X(N)>0 THEN LET R=X(N)-16: GOSUB 2500: PRINT R\$: 430 NEXT N: FOR X=1 TO 15 STEP 2: GOSUB 1700: PRINT "#"; 440 NEXT X 450 LET 77=77+1:LET F=0 460 FOR N=1 TO C+1 470 IF X(N) <> N-1 THEN LET F=1 480 NEXT N 490 IF F=1 THEN RETURN 500 IF G\$="S" THEN PRINT: PRINT"I.ONLY.TOOK."; ZZ: ".MOVES": BOTO 530 510 LET X=0:LET Y=12:60SUB 1700: PRINT"CONGRATULATIONS, !!" 520 PRINT: PRINT"YOU. TOOK. "; ZZ; ". MOVES ... " 530 LET X=0:LET Y=15:60SUB 1700 540 STOP 550 REM COMPUTER MOVES 560 FOR B=1 TO C+1 570 IF G=X(B) THEN LET Q=B 580 IF X(B)=0 THEN LET R=B 590 NEXT B 600 LET X(Q)=0 610 LET X(R)=6 620 GOSUB 380 430 LET T=2: GOSUB 1400 640 RETURN 450 REM AUTOPLAY 660 GDSUB 1100 670 FOR A=1 TO C+1 680 LET X(A)=C+2-A A90 NEXT A 700 LET X(1)=0 710 BOSUB 380 720 FOR M=1 TO 4 730 FOR G=5 TO 1 STEP -2 740 GOSUB 550 750 NEXT G 760 IF M=4 THEN PRINT: STOP 770 FOR G=2 TO 6 STEP 2 780 GDSUB 550

790 NEXT G

LINE NOTES

Lines 110-170: Initialize the starting values of the variables.

Line 180: Sets up the area of the

Lines 190-230: Print starting messages on the screen.

Line 240: If solution is required then sends the program to an automatic play subroutine.

Lines 250-290: Define the position of the numbers and the position of the spaces that separate them.

Lines 300-320: If an invalid move is attempted a message is printed.

Lines 330-340: Store numbers that have been swapped to new positions.

Lines 370–490: Subroutine that draws the shape of the board on screen with the current position of the numbers.

Links 460-480: The numbers are checked to see if they are in the correct order.

Lines 500-520: Messages reporting the number of moves that were taken to complete the puzzle.

Line 540: Ends the program.

Lines 550-640: Subroutine that sorts the computer's moves.

Lines 650-800: Subroutine which controls the computer's moves when it is playing automatically.



INTRODUCTION

This is a page taken from a copy of Huggett's Quality Nougat Newsletter. Can you complete the crossnumber puzzle at the foot of the page?

The six clues to the puzzle fall into three pairs. Numbers 6 & 2 relate to the chairman's age, 1 & 3 concern the pieces of nougat and 4 & 5 the number of emplovees.

From the text we can learn several clues which help to solve the puzzle:

1) The chairman's age and year of birth add up to 1980 the date of the newsletter

2) The number of employees has doubled on five occasions (this is not the same thing as saving that it is now five times

as big). If all possible values are tested it will be found that only correct answers will fit the puzzle.

CHAIRMAN'S AGE

This program determines the chairman's age and his year of birth. We know that the last digit of the year of birth must be the same as the first digit of his age in order to fit the puzzle. Most importantly, both numbers are primes.

In the program all the oddnumbered ages between 11 and 99 years) to see if it is prime.

LISTING: CHAIRMAN'S AGE

100 REM AGE 110 BOSUB 1100

120 FOR A=11 TO 99 STEP 2

130 LET N=A

140 LET X=0:LET Y=2:GOSUB 1700: PRINT "AGE. BEING. TESTED. "; A

150 GOSUB 250 160 IF T=1 THEN GOTO 230

170 LET N=1980-A

180 GOSUB 250 190 IF T=1 THEN GOTO 230

200 LET A1=INT (A/10)

210 LET N1=N-10#INT(N/10)

220 IF A1=N1 THEN PRINT"AGE. = . ": A: PRINT"YEAR. OF. BIRTH. =. "; N: BOSUR 1500

230 NEXT A 240 PRINT: PRINT: STOP

250 REM TEST FOR PRIME

260 LET T=0 270 FOR D=3 TO SOR(N) STEP 2

280 LET R=N/D 290 IF R=INT(R) THEN LET T=1:

LET DESOR(N) 300 NEXT Q

310 RETURN

not tested because, by definition, they are not prime. The program gives values for

his age and year of birth. These numbers should be entered into the crossnumber puzzle as clues 6 and 2 respectively.

However, several possible values in all are found. Make a note of each value when the program stops, then press any key to restart it. Only one of these values will work in the puzzle.

LINE NOTES Lines 120-150: Test the chair-

man's age (in the range of 11 to

99 are tested. Even numbers are | Line 120: Tests all ages from 11 to 99 but excludes even numbers.

> Lines 160-180: Ages which are prime are subtracted from 1980 to find the year of the chairman's birth. This number is also tested to see if it is prime.

Lines 190-220: If both numbers are prime they are placed in strings to see if the digits which are shared in the puzzle are the same. If they are the answer is printed on the screen.

Lines 250-310: Subroutine which tests numbers to see if they are prime. It returns a value of T=0 if a prime is found; otherwise T=1.

NEWSLETTER

FEBRUARY 1980

HUGGETT'S CELEBRATE 100 YEARS OF NOUGAT MAKING This year sees the centenary of toffee making at Huggett's Quality Nougat

Factory. In the hundred years since the firm was founded by Mr. Josiah Hugget, grandather of the present chairman, nougat has been made continuously on the present site, and the firm has always been renowned

Our reporter managed to have a few words with the chairman, Mr. Horatio Huggett, at a party held on the first of last month, to celebrate

Mr. Huggett is reported as saying that this centenary year should, once again, prove a success for the firm, and we were informed that since the firm was founded the number of employees has doubled exactly five times

ver, we reach no presents thomson. Mr. Huggett went on to say that the year the factory was founded was also the year of his late father's birth, and he then hazarded (continued

ACROSS

- 3 If I have as many 2lb boxes of nougar as there are pieces in a box - how many have I?
- 5 Original number of employees. 6 The age of the present chairman (hint: a prime number). DOWN
- 1 Number of pieces of nougat in
- 2 The year our present chairman was born (hint: a prime
- Present number of employees

CROSSNUMBER PUZZLE 4.786



PIECES OF NOUGAT

This program determines the number of pieces of nougat in a 2lb box and also the square of this number (the number of pieces × the same number of hoves)

It is not necessary of course to have a box of nougat to find these values. From looking at the layout of the puzzle we know that the number of pieces in a box must be between 32 and 99 in order to end up with a square that has only four digits. Also, the last digit of the number of pieces in a box is the same as the first digit of the square of this number (to fit the puzzle). Finally, the last digit of the square must equal 9 so as to match the answer we have already discovered

However, the program which follows gives us two possible answers both of which fit the bill. To get the correct solution we have to go on to the next program.



LISTING: NOUGAT

100 REM NOUGAT

110 GOSUB 1100 120 FOR N=32 TO 99

130 LET S=N#N

140 LET N2=N-10*INT(N/10)

150 LET S1=INT (S/1000)

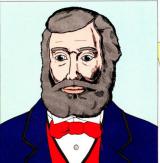
170 IF N2=S1 AND S4=9 THEN PRINT N,S

180 NEXT N

LINE NOTES

Lines 120-130: Every number from 32 to 99 is evaluated one at a time, while the value of the square of each is assigned to the variable S.

Lines 140–180: These values are entered into strings and the values of the relevant digits in each number are checked. All possible answers that fit are printed out.



EMPLOYEES

This program calculates the original number of employees at Huggett's. As the answer has two digits the original number of employees must have been between 10 and 99. To find the present number of employees. the program multiplies the original number by two, five times over.

For the present number to fit in the grid the first digit must be either a 5 or a 9 (from the two answers to the nougat program). Also, the last digit must be the same as the last digit of the original number of employees.

The set of figures returned by the program also eliminates one of the pair of numbers from the nougat program. It is now possible to complete the crossnumber puzzle.

LISTING: EMPLOYEES

100 REM EMPLOY

110 GOSUB 1100

120 FDR N=10 TD 99 130 LET M=N

140 FOR F=1 TO 5 150 LET M=2*M

160 NEXT F

170 LET M1=INT (M/100)

LET M3=M-10#INT(M/10) 180 LET N2=N-10# INT (N/10)

190 IF M1=5 AND M3=N2 OR M1=9

AND M3=N2 THEN PRINT N: ". ": M 200 IF M>999 THEN PRINT: PRINT: STOP 210 NEXT N

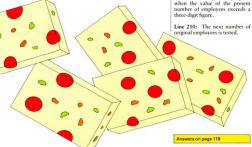
LINE NOTES

Lines 120-160: For each possible value of the number of original employees (from 10 to 99), the present number is found

by doubling it five times.

Lines 170-190: These values are put into strings and the digits are tested.

Line 200: Stops the program when the value of the present



THINKING GAMES: MIDSUMMER'S EVE



INTRODUCTION

It was midsummer's eye and Professors Addle and Glumb were dozing in the common room late that afternoon, the day's lectures behind them and the prospect of dinner with wine and cigars to follow later.

"What's the square root of twelve?" murmured Professor

Addle "Three point four six something," replied Glumb.

"No. no - a whole number! It's got to be a whole number." "Can't be done, old boy!" proclaimed Professor Glumb re-

lapsing into slumber. "Oh but it can!" exclaimed Addle with such enthusiasm that all hope of relaxation instantly vanished.

"Don't see how," said Glumb. "Not in figures," replied Addle, "but if you write it down in letters it can be done."

Seizing a piece of chalk he wrote:

$\sqrt{TWELVE} = ?$

"Simply substitute numbers for the letters and the resulting square root will also make a familiar English word."

"Could be anything," proclaimed Glumb

"Well not really - you see the word can only be made up of the letters found in 'TWELVE' otherwise it wouldn't be a fair question,"

"I see!" said Glumb, "You find out what six-digit number could represent 'TWELVE' by substituting numbers for letters - 3) The program eliminates all each different letter equalling a different digit. Next you find the square root of this number. Then by re-substituting the numbers of

the root for letters you find a common English word." "Exactly!" said Addle.

Glumb reaching for a pencil and paper, "lets see, could it be 'let', wet', 'vet'???"

What is the square root of TWELVE?

SOLUTION

1) From the position of the letters in the word 'TWELVE' we are looking for a perfect square with six digits in which the third and sixth digits are the same, and the other four are all different.

2) In order to produce a square of six digits the root must be in the range 317 to 999.

possible values in which the square root contains digits that are not present in the square. 4) The program also checks the

square to determine if the third and sixth digits are alike. "Still looks tricky," said 5) It then checks the other four digits to see if they are all different from each other, and different from the matching pair of digits already tested.

> 6) Then the program coines up with all possible values within the range 317 to 999. After a value is found press any key to start searching for the next value. 7) You must substitute these

values for letters to find the correct solution.



THINKING GAMES: MIDSUMMER'S EVE

LINE NOTES

Line 110: Sets up an array for the six digits of each perfect square.

Line *20: Sets up text area of the screen.

Line 130: Sets range within which the roots must lie. Line 140: Prints out the number

that is now being tested on the screen.

Line 150: Introduces the vari-

able for the perfect squares.

Lines 160-380: Calculate the values which fit the bill.

Lines 390-410: Print the correct answers for the roots and squares and stop the program when all numbers of the range



LISTING

100 REM TWELVE 110 DIM F(6)

120 GOSUB 1100

130 FDR N=317 TD 999 140 LET X=0:LET Y=6:GOSUB 1700:

PRINT"TESTING.";N

150 LET S=N#N 160 FOR D=6 TO 1 STEP -1

170 LET S1=INT (S/10) 180 LET F(D)=S-10#S1

190 LET S=S1

200 NEXT D 210 IF F(3)<>F(6) THEN GOTO 400

220 LET T=0

230 FOR A=1 TO 4

250 IF F(A)=F(B) THEN LET T=1: LET B=5:LET A=4

260 NEXT B

270 NEXT A 280 IF T=1 THEN GOTO 400

290 LET S=N

300 FOR D=1 TO 3 310 LFT S1=INT(S/10)

320 LET R=S-10#S1:LET S=S1 330 LET T=0

340 FDR A=1 TD 5

350 IF R=F(A) THEN LET T=1 360 NEXT A

370 IF T=0 THEN LET D=3 380 NEXT D

390 IF T=1 THEN PRINT: PRINT"TWELVE.=.";N*N:PRINT: PRINT"ROOT...=.":N:PRINT:

60SUB 1500 400 NEXT N

410 PRINT: PRINT: PRINT: STOP

*After a value is found, press any key to continue the program.

THINKING GAMES: CODEMAKER



INTRODUCTION

It is sometimes necessary to transmit a message to another person so that if it is intercepted the message cannot be understood; in other words, by using some sort of code. Consider the following message:

OETE/RSDCCES

Unlike codes in which letters

are substituted for one another, this code needs a 'key' number in order to be deciphered. In this case the key number is 7.

The sequence of letters above includes '/' which represents a space. Now count from the left seven letters. This brings you to S so copy it down and cross off S in the original coded message. Continue counting another seven letters. If you arrive at the end of the line go back to the start and keep counting until you reach the next letter (E). This should also be copied and crossed off the code. Continue in this manner until the whole message is decoded but remember, do not include letters that have already been crossed off while counting along the line.

You will eventually arrive at this message:

SECRET/CODES
The 'f' is included to show where a space occurs.

Here are some coded messages to decipher. The secret number is

- 100 REM CODES
- 110 DIM U(24) 120 GOSUB 1100
- 130 FOR I=1 TO 24:LET U(I)=0:NEXT I
- 140 LET X=0:LET Y=3:GOSUB 1700:
- 150 PRINT: PRINT "D.=. DECODING"
- 160 PRINT:PRINT "PRESS..C..OR..D"
- 170 GOSUB 1500: IF G\$<>"C" AND G\$<>"D" THEN GOTO 170
- 180 LET H*=G*:GOSUB 2800:GOSUB 1700: PRINT "TYPE.IN.YOUR.MESSAGE":PRINT
- 190 GOSUB 2200:LET L=LEN(K\$):LET L\$=K\$
 200 PRINT:PRINT "ENTER.SECRET.NUMBER":
 GOSUB 2100:LET C=0:LET G=0
 - 210 LET K\$=L\$: IF H\$="D" THEN GOTO 310
 - 220 FOR I=1 TO K 230 LET G=G+1: IF G>L THEN LET G=1
 - 240 IF U(G)>0 THEN GOTO 230
- 250 NEXT I
- 260 LET C=C+1:LET P=C:GOSUB 2700: LET U(G)=A
- 270 IF CKL THEN GOTO 220
- 280 GOSUB 2800:GOSUB 1700: PRINT "CODED.MESSAGE.IS":PRINT
- 290 FOR I=1 TO L:LET R=U(I)-64: GOSUB 2500:PRINT R*::NEXT I
- 300 GOTD 380 310 GOSUB 2800: GOSUB 1700:
- PRINT "DECODED.MESSAGE.IS": PRINT 320 FOR I=1 TO K
- 320 FOR I=1 TO K 330 LET G=G+1: IF G>L THEN LET G=1
- 340 IF U(G)>0 THEN GOTO 330 350 NEXT I
- 360 LET P=G:GOSUB 2600:PRINT P\$;:
- LET U(6)=1 370 LET C=C+1: IF C<L THEN GOTO 320
- 380 PRINT: PRINT: STOP

THINKING GAMES: CODEMAKER

different in each case. As a clue the author is Shakespeare.

the author is Shakespeare.
(1) /NIGKOD//SOYAFHM

(2) GWL/TAAHTREA/LSO/LE/

The secret numbers for these messages are: (1) 17 (2) 34

The program which follows can be used both to code and decode messages although for each trial you must enter the message afresh. Note that when coding messages the '/' must be inserted to denote spaces between words.

The maximum characters and spaces in a message is 24.

LINE NOTES

Line 110: Sets up an array with 24 places for letters and spaces.

Line 120: Sets up text area on screen.

Line 130: Sets starting values of the array.

Lines 140–170: Print question on screen asking you to select the coding or decoding option.

Lines 180-200: Ask you to enter a message and a secret number.

Lines 210-270: Code the message that was entered.

Lines 280-300: Print the coded message on screen then send the program to the end.

Line 310: Prints the decoding message on the screen.

Lines 320-370: Decode the jumbled message that has been entered.



NUMBER PUZZLES: HIGH-LOW



INTRODUCTION

In this program the computer asks you to choose the upper limit to a range of numbers. Then it randomly selects a number between one and the value you picked. Your task is to guess this number in as few moves as possible while making use of the clues given to you.

The computer finally reports how many tries it took you to guess the correct number and announces whether it considers your performance to have been any good.

Most of the time you should be able to get the right number in five to seven tries. The easiest way is by the simple strategy of always picking values halfway between the upper and lower limits of the previous guess you made

Lines 100-130: Print title on the screen and establish the position of the text.

Lines 140-150: Print a question and await your answer (which sets the upper limit to the numbers that can be chosen).

Lines 160-180: Select a random number and then call up a time delay so that the machine gives the impression of thinking about the number it is choosing.

Lines 190-270: Print a message to you that a number has been chosen within the range that was specified.

Line 280: Inputs your guess.

Line 290: If guess was correct this line sends the program to the win statements at the end

Lines 300-340: If guess was too high or too low these lines send the program back for another

Lines 350-400: Print win and lose messages that depend upon the number of tries you took to guess the right number; then end the program.

- 100 REM HILD 110 GOSUB 1100
 - 120 PRINT "...HIGH.LOW"
 - 130 LET T=2:60SUB 1400:60SUB 2800 140 PRINT: PRINT "WHAT. IS. LIMIT?. ":
 - 150 GOSUB 2100
- 160 LET N=K
- 170 GOSUB 1300
- 180 LET T=2: GOSUB 1400 190 LET 6=0 200 BOSUB 2800
- 210 LET 6=6+1
- 220 PRINT
- 230 PRINT "I'VE.PICKED.A.NUMBER"
- 240 PRINT
- 250 PRINT "BETWEEN. 1. AND. ": STR\$ (N) 260 PRINT: PRINT
- 270 PRINT "WHAT. IS. IT?. ":
- 280 GOSUB 2100
- 290 IF K=R THEN GOTO 350 300 PRINT: PRINT
- 310 IF KKR THEN PRINT "TOO.LOW"
- 320 IF KOR THEN PRINT "TOO.HIGH"
- 330 LET T=1:60SUB 1400

NUMBER PUZZLES: HIGH-LOW

340 GOTO 200 350 PRINT: PRINT 360 IF G>6 THEN PRINT "DREADFUL!!" 370 IF G<7 AND G>4 THEN PRINT "NOT. BAD!" 380 IF G<5 THEN PRINT "VERY.GOOD!!" 390 PRINT: PRINT: PRINT" YOU, TOOK, ": STR\$ (G): ". TRIES"

400 STOP



NUMBER PUZZLES: MAGIC SQUARES



INTRODUCTION

Big Harry and Kevin were sitting in the Rolls one afternoon.

"Kevin my son," said Harry, blowing a cloud of cigar smoke into Kevin's face, "business is far too slow. If things don't pick up soon we'll be needing a new line

of work. Know what I mean?" "Right Boss," Kevin gagged, "I'll get onto it right away.

The next night Kevin was wearing a big grin as he slumped down in the Rolls.

"Boss. You'll love this, I've found a jeweller's shop that's just perfect. His valuable stuff is all in an old safe - as big as a room it is. And the lock's a piece of cake. My Uncle Fred, who works there, says it has a 16 number combination. You turn the dial four numbers to the left. Four to the right. Then four to the left and four right again. The only trouble is that old Fred's memory is shot - used to box in the ring as a lad. He says he can only remember seven of the numbers. Funny though, he noticed that all the numbers of the combination make my age. No matter if you add them up and down, across or sideways they always come to the same number - 34!"

Big Harry looked doubtful and blew more cigar smoke.

"This had better be good Kevin, or I'll have your hide as a duster," He flicked at an imaginary speck on the inlaid rosewood panelling.

Kevin still has his hide. The drawing to the right shows the scrap of paper Uncle Fred gave Kevin with the seven numbers he could remember. What was the safe's combination?

```
100 REM MAGIC
110 GOSUB 1100
120 PRINT "...MAGIC.SQUARE"
130 LET T=2:GOSUB 1400:GOSUB 2800
140 LET N=1:LET C=1
150 DIM A(4.4)
160 DIM C(4.4)
170 FOR J=1 TO 4
180 FOR K=1 TO 4
```

190 LET C(J.K)=0 200 NEXT K 210 NEXT J

220 FOR J=1 TO 4 230 FOR K=1 TO 4 240 LET A(J.K)=(J-1) *4+K+N-1

250 NEXT K 260 NEXT J 270 FOR J=1 TO 2

280 LET Z=A(J,J):LET A(J,J)=A(5-J,5-J) 290 LET A(5-J,5-J)=Z:LET Z=A(J,5-J) 300 LET A(J.5-J)=A(5-J.J):

LET A(5-J, J)=Z 310 NEXT J 320 LET C(1,1)=A(1,1) 330 LET C(1.3)=A(1.3)

340 LET C(1.4)=A(1.4) 350 LET C(2,2)=A(2,2) 360 LET C(3, 2)=A(3, 2) 370 LET C(3,3)=A(3,3)

420 FOR J=1 TO 4

380 LET C(4,1)=A(4,1) 390 GOSUB 2800 400 LET X=5:LET Y=0:GOSUB 1700 410 PRINT "A...B...C...D"

430 LET Y=2+3# (J-1) 440 FOR K=1 TO 4 450 LET X=4#K+1: IF C(J.K)>9 THEN LET X=X-1

460 GOSUB 1700: PRINT: C(J.K)

NUMBER PUZZLES: MAGIC SQUARES

470 NEXT K 480 NEXT J 490 LET X=0 500 FOR J=1 TO 4 510 LET Y=2+3*(J-1):60SUB 1700: PRINT STR#(J); 520 NEXT J: IF C=0 THEN GOTO 700 530 LET X=0:LET Y=13:60SUB 1700: PRINT "COLUMN..ROW..NUMBER" 540 LET X=0:LET Y=14:GOSUB 1700 550 GOSLIB 2200+LET C\$=K\$ 560 IF C\$="A" THEN LET C=1 570 IF C\$="B" THEN LET C=2 580 IF C\$="C" THEN LET C=3 590 IF C\$="D" THEN LET C=4 600 LET X=8: GDSUB 1700: GDSUB 2100: 610 LET X=15: GOSUB 1700: GOSUB 2100: LET Nek

620 IF A(R,C)=N THEN LET C(R,C)=A(R,C) 630 LET C=0 640 FOR J=1 TO 4

640 FOR J=1 TO 4 650 FOR K=1 TO 4 660 IF C(J,K)=0 THEN LET C=1 670 NEXT K

680 NEXT J 690 GOTO 390 700 STOP

LINE NOTES

Lines 100-120: Set up the text area of the screen and print the program's title.

Lines 140-210: Set up the initial values of the variables and the dimension of the array that contains the numbers of the safe's combination.

Lines 220-310: Calculate all the numbers and their locations within the array.

Lines 320-380: Store the seven values of the known numbers.

Lines 390–520: Print the magic square together with the known

values.

Lines 530-610: Ask you to enter your guess for the missing

Line 620: Stores your guess if

numbers

Lines 630-680: Check to see if the square is complete.

Line 690: Sends program to the next guess if the square is not

Line 700: Stops the program.





NUMBER PUZZLES: MINDREAD



INTRODUCTION

Not very many people know that computers can be programmed to read minds so this is a chance to surprise somebody in a very big way. Announce to someone that you have set up your machine to mind-read numbers. Say that you need human guinea pigs to test it and then start the program.

If your guinea pigs answer the computer's questions correctly and honestly they will be amazed to find that the number they originally had in mind (or wrote down) is always the one guessed.

If the computer does come up with a different number this is because someone cheated or else made a mistake with their arithmatic. Tell them to have another go and double check their replies to the computer's questions. Be sure to make a big fuss about not letting the computer peek at their calculations!

Lines 100-130: Set up the text

area and print the title on the screen.

Lines 140-200: Print a message on the screen.

Lines 210-340: Ask you to divide the number you have chosen by 3, 5 and 7 and to enter

LISTING

100 REM MINDREAD

110 GOSLIB 1100 120 PRINT"NUMBER, MINDREADING"

130 PRINT: PRINT

140 PRINT"THINK. OF. A. NUMBER"

150 PRINT 160 PRINT"1. TO. 100. BUT. DON' T"

170 PRINT

180 PRINT"TELL. ME. WHAT. IT. IS" 190 LET T=6: GOSUB 1400

200 BOSUB 2800

210 PRINT: PRINT"DIVIDED, BY, 3, THE" 220 PRINT: PRINT "REMAINDER. IS?. ":

230 GOSUB 2100

240 LET A=K 250 PRINT PRINT "DIVIDED, BY, 5, THE"

260 PRINT: PRINT"REMAINDER, IS?, ": 270 GOSUB 2100

280 LET BEK 290 PRINT: PRINT"DIVIDED, BY, 7, THE"

300 PRINT: PRINT"REMAINDER. IS?. ": 310 GOSUB 2100

320 LET C=K

330 LET T=1:60SUB 1400

340 GOSUB 2800

360 PRINT: PRINT"SHH! . I'M. THINKING"

370 PRINT 380 LET T=4:60SUB 1400

390 LET D=0 400 FOR N=1 TO 100

410 IF (N-A)/3(>INT((N-A)/3) THEN 60TO 450

420 IF (N-B)/5<>INT((N-B)/5) THEN GOTO 450 430 IF (N-C)/7<>INT((N-C)/7) THEN

GOTO 450 440 LET D=N:LET N=100

450 NEXT N

460 IF D=0 THEN PRINT "I.GIVE.UP!": GOTO 510

470 PRINT "WAS. IT. ": STR\$ (D): "7. Y. OR. N";

480 GOSUB 1500 490 PRINT: PRINT 500 IF G\$="Y" THEN

PRINT "HA!.I.READ.YOUR.MIND": BOTO 540

NUMBER PUZZLES: MINDREAD

510 PRINT:PRINT"OK!.LET'S.TRY.AGAIN" 520 LET T=8:GOSUB 1400 530 GOTO 100 540 STOP the remainders.

Lines 350–450: The computer takes all numbers between 1 and 100, finds the remainders and compares them with the 3 numbers that were entered.

Line 460: If no match is found an error message is printed.

Line 470: Prints answer.

Lines 480-540: Print closing messages and stop the program.



NUMBER PUZZLES: AIR RESCUE



INTRODUCTION

You are an Air Force general called upon to plan how to ferry emergency supplies to teams of men in trouble at various distances from your home base. However, one of the conditions of the exercise is that your planes do not have the capability to reach the destination directly, which is always outside their maximum range. Nor are they also that the distance of the

Ace pilot Rickenbacker suggests that mid-air refuelling might provide the solution.

migat provide the solution.

"Dat give an a squadron of ideal control of the cont

You have only one question to ask. How many planes do you need, say, to extend the range two times, four times or any other desired distance? All the pilots look puzzled and you are left to work out the answer with your computer.

LISTING

```
100 REM RESCUE
110 GDSUB 1100
120 LETX=4:LETY=8:GDSUB 1700
```

130 PRINT"AIR.RESCUE"

140 LET T=4:GOSUB 1400 150 GOSUB 2800

160 LET N=19:GOSUB 1300

180 LET R=0

190 FOR C=1 TO P 200 LET R=R+1/C

210 NEXT C

220 LET R=(INT(R*10))/10 230 PRINT:PRINT "YOUR.RANGE.IS.":R

240 PRINT:PRINT "TIMES.NORMAL" 250 PRINT:PRINT "HOW.MANY.PLANES"

260 PRINT: PRINT "DO. YOU. NEED?."; 270 GOSUB 2100

280 IF K=P THEN PRINT:PRINT "CORRECT"
290 IF K<>P THEN PRINT:

PRINT "WRONG. - ANSWER. "; P 300 LET T=2:GOSUB 1400:GOSUB 2800 310 PRINT:PRINT "HOW. MANY.TIMES"

310 PRINT:PRINT "HOW.MANY.TIMES"
320 PRINT:PRINT "DO.THEY.REFUEL?.";
330 GOSUB 2100

340 IF K=P-1 THEN PRINT: PRINT "CORRECT"

350 IF K<>P-1 THEN PRINT: PRINT "WRONG.-.ANSWER.";P-1 360 STOP

HINT

The number of planes needed to reach a particular destination rises dramatically as the flight range increases.

The number of refuelling operations is always one less than the number of planes that set out.

LINE NOTES

Lines 100-140: Set up the text area and print the title on the screen for a short time.

Lines 150–160: Clear screen and choose a random number for the range to the destination.

Lines 170-220: Calculate the number of planes needed to reach the destination.

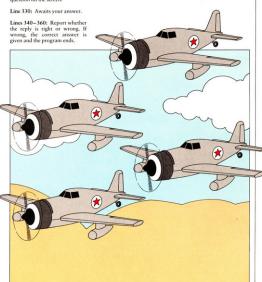
Lines 230-260: Print a message asking for your guess.

Line 270: Awaits your reply.

Lines 280-290: Report if the reply is right or wrong and also provide the correct answer.

NUMBER PUZZLES: AIR RESCUE

Lines 300-320: Print a new question on the screen.



NUMBER PUZZLES: SWAP



INTRODUCTION

In this program the computer presents you with a list of six numbers that are in jumbled order. You must rearrange them into numerical order from left to right so that they read:

10 20 30 40 50 60

With each move you must tell the computer how many numbers to reverse, counting from left to right. For example, you may find that the jumbled order

30 20 10 40 50 60

is as follows:

The computer asks how many need to be reversed. If your answer is three then the first three numbers will be swapped around and the new sequence will look like this:

The numbers are now in order.

Obviously you cannot reverse only one number since that will not change its position. The most you can reverse is six – the entire

The object of the game is to unscramble the jumbled numbers in as few moves as possible.

The best strategy to use is to bring the largest numbers to the front of the row, and then to swap them into their correct position.

TIMT

The formula 2N-3 (where there are N numbers in the row) predicts the total number of moves you would need to win the game if you yourself were a computer. Of course you should be able to do much better than a mere machine!

LINE NOTES

Lines 100-130: Establish the area of the text and print the title of the program on screen in the correct position.

Line 140: Sets up an array of six numbers.

Line 150: Introduces the numbers 10 to 60 in units of ten.

Lines 160-170: Randomly select the sequence of the six numbers and position them in jumbled order.

Line 180: Sets up the counter for the number of moves that are taken.

Lines 190-200: Check if the numbers are in the correct order.

Lines 210-230: Print the sequence of the jumbled numbers on screen.

- 100 REM SWAP
- 110 GOSUB 1100
- 120 PRINT " SWAP"
- 130 LET T=2:60SUB 1400
- 150 FOR I=1 TO 6:LET N(I)=10#I:
- NEXT I 160 LET N=6:FOR I=1 TO 6:GOSUB 1300
- 170 LET T=N(I):LET N(I)=N(R): LET N(R)=T:NEXT I:LET M=0
- 180 LET M=M+1:LET T=0.5:60SUB 1400:
- 190 LET F=0:FOR I=1 TO 6: IF N(I)<>10*I THEN LET F=1
- 200 NEXT I
- 210 PRINT: PRINT "JUMBLED.ORDER.IS": LET Y=4
- 220 FOR I=1 TO 6: LET X=3\$1-2:60SUB 1700
 - 230 PRINT STR*(N(I)):NEXT I
 - 240 LET X=0:LET Y=7:GOSUB 1700: IF F=0 THEN GOTO 300
 - IF F=C

NUMBER PUZZLES: SWAP

Lines 240–260: Ask you how many numbers you wish to reverse for a particular move.

Lines 270-290: Carry out the number swap you have chosen and prepare the program for the next move.

Lines 300-310: Print win message when the numbers have been placed in correct order and stop the program.



250 PRINT "REVERSE.HOM.MANY?.";
260 GOSUB 2100
270 FOR !=! TO !NT(K/2):
LET T-N(!):LET N(!)=N(K+1-1)
280 LET N(K+1-1)=TNEXT !
290 GENTRENT "YOU.WIN.IN.";STR*(H);
".HOVES"
30 PRINT:PRINT:STOP



NUMBER PUZZLES: ANYDAY



INTRODUCTION

Like most people you find it easy to remember the date of your birthday. But can you remember the day on which you were born? Have you any idea what your precise age is and not only the year? Can you figure out on what day the next century will begin?

These and similar questions can all be answered by this program. All you have to do is enter today's date (day, month, year). Then enter the day, month and year that you are interested in and the computer will rell you on what day of the week the date occurs and how much time will elapse, or has gone by, between then and now. In this way you can discover the day or which can discover the day or which the age of someone or somethine.

TAMA

- 1. World War II ended on May 8. 1945. What day was that?
- The United States of America declared its independence on July 4, 1776. How old is the country?
- 3. How long is it until the next century begins?

- 100 REM ANYDAY
- 110 GOSUB 1100 120 PRINT "....ANY.DAY"
- 130 LET T=2:60SUB 1400:60SUB 2800
- 130 LET T=2:GOSUB 1400:GOSUB 2800 140 PRINT:PRINT"WHAT.IS.TODAY'S.DATE"
- 150 PRINT:PRINT"WHAT. 15. TUDAY S. DATE
- 160 PRINT:PRINT"1-31..1-12...19--"
- 170 LET X=0:LET Y=8:GOSUB 1700: GOSUB 2100:LET D1=K
- 180 LET X=6:GOSUB 1700:GOSUB 2100: LET M1=K
- 190 LET X=15:GOSUB 1700:GOSUB 2100: LET Y1=K
 - 200 IF Y1<100 THEN LET Y1=1900+Y1 210 GOSUB 2800; PRINT;
- PRINT"DATE.OF.INTEREST"
 220 PRINT:PRINT"DAY...MONTH...YEAR"
- 230 LET X=0:LET Y=7:GOSUB 1700: GOSUB 2100
- 240 LET D2=K:LET X=7:60SUB 1700: G0SUB 2100:LET M2=K
- 250 LET X=15:60SUB 1700:60SUB 2100:
- 260 IF Y2<100 THEN LET Y2=1900+Y2
- 270 GOSUB 2800:PRINT:PRINT
- 280 DIM D(12) 290 FOR Z=1 TO 12:READ D(Z):NEXT Z 300 DATA 0.31.59.90.120.151.181.212.
- 243,273,304,334 310 LET Y3=Y1-1900:LET Y4=Y2-1900
- 320 LET T1=365*Y3+INT(Y3/4)+D(M1)+D1 330 LET T2=365*Y4+INT(Y4/4)+D(M2)+D2
- 340 IF Y3=4*INT(Y3/4) AND M1<3 THEN LET T1=T1-1
- 350 IF Y4=4*INT(Y4/4) AND M2<3 THEN LET T2=T2-1



NUMBER PUZZLES: ANYDAY

360 PRINT: PRINT "TIME, INTERVAL, IS." 370 PRINT: PRINT: ABS(T2-T1); ". DAYS" 380 LET T=ARS(T2-T1)

390 LET Y=INT(T/365)

400 LET T=T-365*Y 410 LET M=INT(T/30.5)

420 LET D=INT(T-30.5#M)

430 PRINT: PRINT "WHICH. IS": PRINT 440 PRINT STR\$ (Y); ". YEARS. "; M;

". MONTHS" 450 PRINT "AND. "; D; ". DAYS": PRINT 460 PRINT: PRINT STR\$ (D2): "/": STR\$ (M2):

"/":STR\$(Y2);"."; 470 IF T1<T2 THEN PRINT "WILL.BE"

480 IF T1=T2 THEN PRINT "IS" 490 IF T1>T2 THEN PRINT "WAS"

500 LET R=T2-7*INT(T2/7)+1

510 FOR I=1 TO R:READ D\$: NEXT I 520 PRINT:PRINT "A.":D\$

530 PRINT: STOP 540 DATA "SUNDAY", "MONDAY", "TUESDAY"

550 DATA "WEDNESDAY", "THURSDAY", "FRIDAY", "SATURDAY"

LINE NOTES

Line 110: Sets up the area of the text on screen.

Lines 120-160: Print program title and ask you to enter the current date.

Lines 170-200: Input current day, month and year, and adjust the year to a four-digit number if only two digits were entered.

Lines 210-220: Ask for date in which you are interested.

Lines 230-260: Input the day. month and year of interest and adjust to a four-digit number.

Lines 270-300: Convert months into days and store these together with the number of days entered.

Lines 310-330: Calculate the years and days elapsed relative to 1 January, 1900.

Lines 340-350: Adjust for leap years.

Lines 360-370: Print message showing number of elapsed days.

Lines 380-420: Convert days into months and years.

Lines 430-450: Print message showing number of days, months and years elapsed.

Lines 460-550: Calculate the day of the week on which the date of interest falls

*Because the program assumes that a month is always 30 days the time interval can be a day or two in error.

NUMBER PUZZLES



INTRODUCTION

In ancient times the King of Stam ruled five provinces. Every year the governors of each province sent the taxes they collected to the King's treasury. The taxes came in great leather sacks filled with silver bars that weighed exactly one kilo apiece. Each sack had the name of the governor and his province marked clearly on the side.

One year a beggar rushed into the court with news that enraged the King. He claimed that the bars from one of the provinces had been clipped with such skill that nobody could spot the difference. Yet each bar was exactly to grams short in weight. Before of the dishonest governor and arrow whistled in through a window and struck the poor man dead.

The King swore he would avenge the death of his loyal subject and behead the dishonest governor. But he faced a small problem with the treasury scales. As they were borrowed from his clephant ranch they were only able to measure weights greater than 10,000 grams.

But the King, who was as clever as he was rich, devised a brilliant solution. Lining up the five sacks in a row he took one bar from the first, two bars from the second, three from the third, four from the fourth and five from the fifth and last sack in line. He piled all the bars on the scale knowing that together they should weigh 15,000 grams. The number of grams they were short thus showed him which sack they came from. Which governor did the King behead?

LINE NOTES

Lines 100-210: Position the area of the text on screen and print the program's title.

Lines 220-300: Print message and diagram on screen.

Line 310: Randomly selects the governor who is dishonest.

Lines 320-370: Calculate the actual weight of the bars.

Lines 380–400: Ask for your guess and wait for an answer to be input.

Lines 410-450: Print final messages and report if your guess was right or wrong, then end program.



THE DISHONEST GOVERNOR



450 STOP

```
LISTING
 100 REM GOVERNOR
 110 BOSUR 1100
 120 PRINT
 130 PRINT "....THE.WISE.KING"
 140 PRINT
 150 PRINT ".....AND. THE ... "
 160 PRINT
 170 PRINT ".....DISHONEST.."
 180 PRINT
 190 PRINT ".....GOVERNOR..."
200 LET T=5:GOSUB 1400
210 GOSUB 2800
220 PRINT
 230 PRINT "THE GOVERNORS"
240 PRINT
250 PRINT "BARS. SHOULD. WEIGH"
260 PRINT: PRINT "1. # ........... 1000"
270 PRINT:PRINT "2.*.*.....2000"
280 PRINT: PRINT "3. *. *. *..... 3000"
290 PRINT: PRINT "4.*.*.*....4000"
 300 PRINT: PRINT "5. *. *. *. *. . . 5000"
 310 LET N=5:60SUB 1300
320 LET S=R#10
 330 LET T=15000-S
 340 PRINT: PRINT "BUT. TOTAL. WAS. ":
     STR$(T)
 350 LET T=10:GOSUB 1400
 360 GDSUB 1200
 370 PRINT: PRINT STR$ (S):
     ". GRAMS. SHORT"
 380 PRINT: PRINT "WHICH. GOVERNOR"
 390 PRINT: PRINT "WAS. BEHEADED?. ";
 400 GOSUB 2100
 410 PRINT
 420 IF K=R THEN PRINT"CORRECT":
     GOTO 450
 430 PRINT "WRONG"
 440 PRINT:
```

PRINT "IT. WAS. NUMBER. ": STR\$ (R)

NUMBER PUZZLES: MASTERMIND



INTRODUCTION

In this puzzle the computer picks a three-digit number at random using digits between one and nine (not zero). You have 20 guesses to find out what the number is. Because there are 888 numbers to choose from it takes a little cunning to work out what the right one is.

With each guess you must select a three digit number. The computer tells you how many digits are right and how many are wrong, and whether any of the right ones are in the correct position in the number.

For example, if the correct number is 135 and you guessed 259, the computer would report that you had one right number but in the wrong position. If you had guessed 925 then it would have reported one right number in the correct position.

The best strategy with this puzzle is to eliminate as many wrong numbers as possible before trying to find the correct position of the right ones.

VARIATIONS

You should be able to solve the puzzle in far less than 20 guesses once you have tried it a few times. In line 250 change I=1 to 20 into I=1 to 10 and see if you can still beat the machine.

LINE NOTES

tween 111 and 999.

Lines 100-130: Set up the text area and print the title on screen.

Lines 140-150: Set up arrays and establish the initial values in

each. Lines 160-240: Randomly select a three digit number be-

Line 250: Establishes the maximum number of guesses you can take

Lines 260-300: Ask you to enter your guess and check that it is valid

Lines 310-330: Store the number you guessed.

Lines 340-440: Check to see if some of the digits are correct but in the wrong place.

Lines 450-480: Check the number of digits that are correct and in the correct place.

Line 490. If all are correct sends

LISTING

- 100 REM MASTER 110 GOSUB 1100
 - 120 PRINT "....MASTERMIND"
- 130 LET T=4: GOSUB 1400: GOSUB 2800 140 DIM A(3)
- 150 DIM B(3)
- 160 LET Y1=0:LET T1=255
- 170 LET M=0:FOR I=1 TO 3
- 180 LET N=9:60SUB 1300
- 190 LET A(I)=R
- 200 IF I-1=0 THEN GOTO 240 210 LET F=0:FOR J=1 TO I-1
- 220 IF A(I)=A(J) THEN LET F=1
- 230 NEXT J: IF F=1 THEN GOTO 190 240 LET M=10*M+A(I):NEXT I
- 250 FOR I=1 TO 20 260 BOSUB 2800
- 270 PRINT: PRINT "GUESS, NUMBER, ": STR&(I)
- 280 PRINT: PRINT "ENTER. GUESS. ": 290 GOSUB 2100
 - 300 IF K<111 OR K>999 THEN GOTO 260 310 LET B(1)=INT(K/100) 320 LET B(2)=
 - INT ((K-100 * INT (K/100)) /10)
 - 330 LET B(3)=K-10*INT(K/10) 340 LET C=0:LET D=0
 - 350 FOR J=1 TO 2 360 IF A(J) <>B(J+1) THEN GOTO 380
 - 370 LET C=C+1 380 IF A(J+1)<>B(J) THEN GOTO 400

NUMBER PUZZLES: MASTERMIND

program to a win message.

Line 500: Checks if none are correct and in the wrong place.

Line 510: Reports on what has been checked.

Line 520: Checks if none are correct.

Lines 530-560: Print report of what has been checked. Line 570: Sends the program to

the next guess.

Lines 580-600: After 20 wrong guesses give correct answer.

Lines 610-620: Win message and end of program.

925



. 259

390 LET C=C+1
400 NEXT J
400 NEXT J
41 FF ACT (<) 28(3) THEN GOTO 430
41 FF ACT (<) 28(3) THEN GOTO 450
440 LET C=C+1
430 IF ACT (<) 28(1) THEN GOTO 450
440 LET C=C+1
450 FF R J=1 TO 3
450 IF ACT (<) 38(1) THEN GOTO 480
460 IF ACT (<) 470 IF D=3 THEN GOTO 450
500 IF C=O THEN GOTO 510
500 IF C=O THEN GOTO 510
500 IF C=O THEN GOTO 510
500 FF C=O THEN GOTO 520
510 FFRINT FERS (C);
520 FFRINT STRE (D); **CORRECT**

540 IF C+D<>O THEN GOTO 560 550 PRINT: PRINT "NOTHING. CORRECT" 560 LET T=2: GOSUB 1400 570 NEXT I

600 GOTO 620

620 STOP

580 PRINT:PRINT"THAT'S.20.GUESSES"
590 PRINT:PRINT"NUMBER.WAS.":STR*(M)

610 PRINT: PRINT "YOU, GOT, IT. !"

986

. 127

789

.675

[46

NUMBER PUZZLES: PICKUP STICKS



INTRODUCTION

Here is another chance to match wits with your computer – and perhaps win. Imagine that there are 21 brightly coloured sticks lying in three rows. With each turn, not less than one and not more than seven sticks must be 'picked up' and removed from the game.

You and the computer take turns at picking up. The object of the game is to force your opponent to take away the last stick. The computer will try to beat you in turn by making you pick

The best strategy for winning same is to work backward from the last remaining stick, or to have anything from two to eight sticks available (so you can take away from one to seven and so leave the computer with only a single stick). Therefore, you want to make sure that the computer of the single stick is a single stick to the computer with only a single stick). Therefore, you want to make sure that the computer ends up with at least intensities on its second to last

It sounds confusing at first, but if you keep working backwards like this you will find that you can win with the first move of the game if you are clever. Remember – the computer always asks you to go first so you should be able to be at it most of the time.

HINT

Draw three rows with seven sticks in each and cross off the ones which are removed as you play. It will help you to keep track of the ones which remain.

LINE NOTES

Lines 100-130: Set up the text area and print the title on screen,

Lines 140–180: Establish initial values of the variables and print

the starting number of sticks.

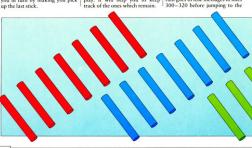
Lines 190–200: Ask that you make a move.

Lines 210-230: If you enter the wrong values these lines either send you back to go again or comment on your error (this is to stop you cheating).

Line 240: Calculates the new total after you make a move.

Line 250: If some sticks remain after your move the new total is shown by lines 330–350.

Line 260: If the last stick has been taken away then the program goes to lose messages in lines 300-320 before jumping to the



NUMBER PUZZLES: PICKUP STICKS

end of the game.

Lines 270-280: Comment if the number selected was nonsense or if you took away more than the remaining number of sticks.

Lines 330-350: The new total is printed after your move is completed.

Line 360: If more than one stick remains then the program starts the computer's move.

Lines 370-390: The computer admits defeat.

Lines 400-440: The computer selects and reports on the number it has picked.

Lines 450-480: Print the total sticks remaining. Line 490: Sends program to the

next round of moves.



LISTING

100 REM PICKUP

110 GOSUB 1100

120 PRINT"...PICKUP.STICKS"

130 LET T=4:60SUB 1400:60SUB 2800

140 LET S=21:LET L=7 150 GOSUB 2800: PRINT: PRINT

160 LET Z\$=STR\$(S)

170 PRINT "TOTAL. IS. "; Z\$

180 LET T=1:60SUB 1400 190 PRINT: PRINT"YOUR. GO. ":

200 GOSUB 2100

210 IF KOL THEN GOTO 150

220 IF K<=0 THEN GOTO 280 230 IF K)S THEN GOTO 150

240 LET S=S-K

250 IF \$>0 THEN GOTO 330 260 IF S=0 THEN GOTO 300

270 LFT S=S+K

280 PRINT: PRINT"SILLY": 60SUB 1400: **60TO 150**

290 LET T=2:60SUR 1400:60T0 410 300 PRINT: PRINT "YOU, TOOK, LAST, STICK"

310 PRINT: PRINT "YOU, LOSE, !!" 320 GOTO 500

330 LET Z\$=STR\$(S) 340 PRINT: PRINT "TOTAL. IS. ": Z\$

350 LET T=1:60SUB 1400

360 IF S<>1 THEN GOTD 400 370 PRINT: PRINT "I. HAVE. TO. TAKE."

380 PRINT: PRINT "LAST. STICK" 390 PRINT: PRINT "YOU, WIN, !!": GOTO 500 400 LET R=S-INT((S-1)/(L+1))*(L+1)-1:

IF R>O THEN GOTO 420 410 LET N=S-1: GOSUB 1300: IF R>L THEN GOTO 410

420 LET Z\$=STR\$ (R) 430 PRINT: PRINT "I. TAKE. AWAY. ": Z\$

440 LET S=S-R 450 PRINT

460 LET 74=STR4 (S) 470 PRINT "TOTAL, IS, ": 7\$

480 LET T=3:60SUB 1400:60SUB 2800

490 GOTO 150 500 STOP

NUMBER PUZZLES: STAR COURSE



INTRODUCTION

This program requires that you steer by the stars to a precise location somewhere within the Milky Way.

The computer randomly selects a region of the galaxy from a map which is divided into 100 grid sections. You must take your ship there, as quickly as possible, to await further orders from home base.

The computer's choice of location is a secret number that you must guess in the fewest possible moves (so as to conserve fuel). With each guess you make, the computer prints out a row of between one and six stars. With six you are far off target. When one appears on screen you are almost there.

After some practice you should be able to reach the right location in no less than seven guesses every time. Less than five is very good indeed.

Each star on the screen represents a power of 2:

 $1 \text{ star} = 2^1 (2)$ $2 \text{ stars} = 2^2 (4)$ $3 \text{ stars} = 2^3 (8)$

 $4 \text{ stars} = 2^4 (16)$ $5 \text{ stars} = 2^{5} (32)$

 $6 \text{ stars} = 2^6(64)$

 $7 \text{ stars} = 2^{7} \text{ (for values to } 128$

LINE NOTES

Lines 100-130: Set the text area and print the title onto the screen.

Lines 140-150: Choose a random number from 1 to 100.

Lines 160-180: Set the initial value of the counter which keeps track of the number of guesses.

Lines 190-250: Print a message asking you to start making your guesses.

Line 260: If your guess is a correct one this line sends the program to the final report messages at the end.

Lines 270-280: Calculate how close you have come to the right number.



NUMBER PUZZLES: STAR COURSE

Lines 290-310: Establish how many stars distance away you are.

Lines 320-370: Print the correct number of stars on screen.

Line 380: Sends program back for the next guess.

Lines 390-440: Print report messages and stop the program.



LISTING

100 REM STARS

110 GOSUB 1100

120 PRINT "...STAR.COURSE"

130 LET T=2:60SUB 1400:60SUB 2800

140 LET N=100 150 GOSUB 1300

150 GDSUB 1300 160 LET G=0

170 GOSUB 2800

180 LET G=G+1

190 PRINT: PRINT 200 PRINT "LOCATION. PICKED"

210 PRINT

220 PRINT "BETWEEN. 1. AND. 100"

230 PRINT "WHAT. IS. IT?. ":

250 GOSUB 2100 260 IF K=R THEN GOTO 390

260 IF K=R THEN GOT 270 PRINT:PRINT

280 LET D=ABS(R-K)

290 LET S=2:LET W=1 300 JE D<=S THEN BOTO 320

310 LET S=S#2:LET W=W+1:60TD 300

320 PRINT: PRINT "....";

330 FOR M=1 TO W 340 PRINT "#.":

340 PRINT "*. 350 NEXT M

360 PRINT 370 LET T=4:60SUB 1400

380 GOTO 170

390 PRINT: PRINT 400 IF G>6 THEN PRINT" DREADFUL!!"

410 IF G<7 AND G>4 THEN

PRINT "NOT.BAD!"
420 IF G<5 THEN PRINT "VERY.GOOD!!"

430 PRINT:PRINT:PRINT"YOU.TOOK."; STR\$(G):".TRIES"

440 PRINT: STOP

 In line 310 the symbol * stands for S to the power of two.

SCIENCE PROBLEMS: CONVERSION



INTRODUCTION

While travelling through Transylvania you arrive at a village where you wish to spend where you wish to spend the night. There is nobody around at the local inn until a helpful passerby tells you to wait for the minkeeper's wife. You can't miss her in a crowd, he adds, as she is two metres tall. You scratch your head and wonder exactly how high that is.

Later on you are shown your room. From the doorway you decide that it is not for you. The place looks bug infested. Don't worry about the fleas says the innkeeper's wife. They can only leap 20 centimetres. Nervously you wonder if any have already reached you from the bed on the other side of the from.

Later that evening you snap a lace on your boots. The innkeeper's wife says she has cord that is 12 millimetres thick. Will that do? You are unsure.

Even if you never visit Transylvania you may still find it useful to know what the metric equivalent of inches, feet and yards amounts to. The program which follows is a useful converter from imperial to metric measurement and vice very

It also allows you to feed in any combination of measurements, so that 5yd:2ft:7in gives the same answer as 211 inches.

LINE NOTES

screen.

Line 110: Sets up text area on

Lines 130-160: Print choice of conversion mode onto the

Lines 170-180: Select conver-

Lines 190-210: Input metric values.

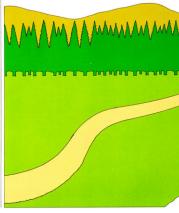
Lines 220-240: Calculate equivalent imperial values.

Lines 250-260: Print out the answer in imperial units.

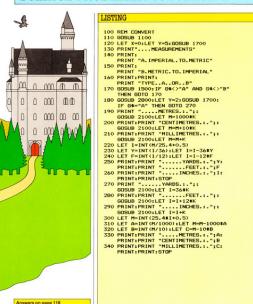
Lines 270–290: Input imperial values.

Lines 300-320: Calculate equivalent metric values.

Lines 330-340: Print out answer in metric units.



SCIENCE PROBLEMS: CONVERSION



Answers on page 116

SCIENCE PROBLEMS: AVERAGE SPEED



INTRODUCTION

This program is one that could drive commuters mad with frust tration. If they were to enter the time they left home, the time they arrived at the office and the distance travelled, most would discover that the average speed of their journey was less than a only calculate journeys up to 24 hours in length. It does this using the 24-hour clock,

LINE NOTES

Line 110: Sets up an array for keeping track of elapsed time up to 24 hours.

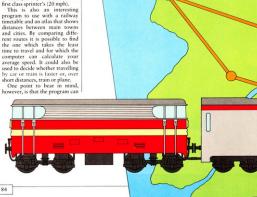
Line 120: Sets up the text area

Lines 130-150: Print title of the program and then clear the screen.

Lines 160-210: Input the starting times and the arrival times of the trip.

Lines 220-240: Calculate the time elapsed in travelling.

Line 260: Prints out length of



SCIENCE PROBLEMS: AVERAGE SPEED

time travelled.

Lines 270–280: Input distance travelled.

Lines 290-300: Calculate the average speed and print it on the screen.

Line 310: Stops the program.

LISTING

- 100 REM SPEED
- 110 DIM T(2) 120 GOSUB 1100
- 130 LET X=3:LET Y=D/2:GOSUB 1700:
- PRINT "AVERAGE SPEED"
 140 LET T=3:GOSUB 1400:GOSUB 2800
- 150 LET X=0:LET Y=3:GOSUB 1700
- 160 PRINT"STARTING.TIME" 170 PRINT:PRINT "HOURS.:.";:
- GOSUB 2100:LET T(1)=60*K
- 180 PRINT:PRINT "MINUTES .:. ";:
- GOSUB 2100:LET T(1)=T(1)+K 190 PRINT:PRINT "ARRIVAL.TIME"
- 200 PRINT:PRINT "ARRIVAL.TIME"
- 900 PRINT:PRINT "HOURS.:.";: 60SUR 2100:LET T(2)=60*K
- 210 PRINT: PRINT "MINUTES .:. ";:
- GOSUB 2100: LET T(2)=T(2)+K 220 IF T(2)<T(1) THEN
- LET T(2)=T(2)+720
- 230 LET U=T(2)-T(1)
- 240 LET H=INT (U/60):LET M=U-60#H
- 250 GOSUB 2800:LET X=0:LET Y=2: GOSUB 1700
- 260 PRINT "TIME.TRAVELLED.=":PRINT:
- PRINT ;H; ".HOURS..";M; ".MINUTES"
- 270 PRINT: PRINT: PRINT "DISTANCE. TRAVELLED"
- 280 PRINT:PRINT "MILES .: . ";:
- GOSUB 2100
- 290 LET S=INT(60*K/U+0.5)
 300 PRINT:PRINT "AVERAGE SPEED.=":
- PRINT: PRINT S: ".MPH."
- 310 PRINT: STOP



SCIENCE PROBLEMS: FLY BALL

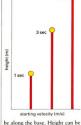


INTRODUCTION

This program tests your knowledge of the way gravity works. I calculates the effect of throwing a ball into the air at different speeds. Then it asks you to estimate the time that it will stay aloft and also the maximum height that it will reach.

The program operates in metric units only – so speed is always calculated in metres per second (m/s). For comparison, a hard hit tennis ball may reach 100 mph (about 45 m/s).

One of the easiest ways to visualize the effect of the different launch speeds in the program is by constructing a bar graph that plots height against velocity. Let the starting velocity



measured on the vertical axis.
You should soon be able to
predict the height the ball will
reach and the time it stays aloft
simply from the speed at which it

You will swiftly discover that height increases dramatically with the length of time the ball

stays in the air.

LINE NOTES

Line 110: Sets up the area of the text on the screen.

Line 120: Randomly selects

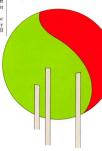
values for the initial velocity of the ball in a range from 11 m/s to 45 m/s.

Line 130: Prints a message showing the initial velocity.

Lines 150-180: Ask you to guess the time the ball is aloft and the maximum height it reaches; calculate the correct results.

Line 190: Stops the program.

Lines 200-230: Subroutine that compares your guesses with the actual values and then prints the answers.



SCIENCE PROBLEMS: FLY BALL

LISTING 110 GOSUB 1100

100 REM GRAVITY

120 LET N=35:GOSUB 1300:LET R=R+10

130 LET X=0:LET Y=2:GOSUB 1700: PRINT"BALL.THROWN: . ";R; ".M/S" 140 LET T=4: GOSUB 1400 150 PRINT: PRINT:

PRINT "TIME, IN, AIR?: ":: **GOSUB 2100**

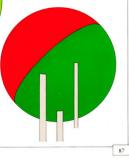
160 LET A=INT(2#R/9.81+0.5): GOSUB 200: PRINT ". SECONDS"

170 PRINT: PRINT "MAXIMUM. HEIGHT?: ":: 50SUB 2100

180 LET A=INT(R#R/19.62+0.5): GOSUB 200: PRINT ". METRES" 190 PRINT: STOP

200 IF ABS (K-A) >A/5 THEN GOTO 220 210 PRINT: PRINT "WELL. DONE" 220 PRINT: PRINT "ANSWER. IS. ": A:

230 RETURN



SCIENCE PROBLEMS:



INTRODUCTION

On a hot summer day, as the temperature rises to 30°C (86°F), we all head for the shade to find relief. In the dead of winter a cold spell of ~20°C (~4°F) keeps us huddled indoors for warmth. Yet these figures fade into insignificance compared to the extremes that exist on Earth. The thermometer in Libya has climbed to \$8°C (136°F) to set a record high while in Antarctica ~88°C (~127°F) of cold has been reported.

By contrast, man-made temperatures greatly exceed anything found on Earth. Laboratory temperatures low enough to freeze air are common while research scientists have come within a fraction of a degree of absolute zero —273°C/ (-460°F), the coldest temperature that can exist in the universe. At the other end of the scale core temperatures of the Sun reach 15 million °C (27 million °F).

In 1714, the Dutch physicist Cabriel Fahrenheit invented the first mercury thermometer. With it he worked out a temperature scale we know as Fahrenheit degrees. On this scale water freezes at 32° and boils at 212°. Not long after Fahrenheit (in 1742) a Swedish astronomer anamed Celsius developed a much

simpler temperature scale. His was divided into a 100 units between the freezing temperature of water and its boiling point.

However, converting from one scale to the other is a difficult and time-consuming calculation. The formula for changing Celsius degrees into Fahrenheit egrees is 95.8°C+32. Converting from Fahrenheit to Celsius is F-32×59. A computer is ideal for performing a mathematical for the such as this. The program complete the blanks in the conversion table below.



	°C	°F				
Absolute zero:	-273°					
Oxygen becomes liquid:	-183°					
Alcohol freezes:		-170°				
Coldest average temperature on Earth:		-72°				
Ice melts:		32°				
Highest average temperature on Earth:		94°				
Temperature of human body:		98.6°				
Water boils:	100°					
Paper burns:		451°				
Silver melts:	961°					
Gold melts:	1063°					
Core of the Earth:	2500°					
Surface of Sun:	6000°					

HOT & COLD TEMPERATURES



LINE NOTES

Line 110: Sets up the area of the text on screen.

Lines 130–180: Print a message on screen asking you to choose a specific conversion mode and to enter the temperature in which you are interested.

Lines 190-230: Input values as centigrade degrees, convert them to Fahrenheit and print the answers.

Lines 240-270: Input Fahrenheit values, convert them to centigrade and print the answers.

Line 280: Stops the program.

LISTING

- 100 REM TEMPS
 - 110 GOSUB 1100
- 120 LET X=0:LET Y=2:G0SUB 1700
- 130 PRINT "....TEMPERATURE": PRINT: PRINT"....CONVERSIONS"
- 140 PRINT: PRINT "A. =. CENTIGRADE. TO":
- PRINT:PRINT "....FAHRENHEIT"
 150 PRINT:PRINT "B.=.FAHRENHEIT.TO":
- PRINT:PRINT "...CENTIGRADE"
 160 PRINT:PRINT "PRESS..A..OR..B"
- 170 GOSUB 1500: IF G\$<>"A" AND G\$<>"B" THEN GOTO 170
- 180 GOSUB 2800: IF G\$="B" THEN GOTO 240
- 190 LET Y=4:GOSUB 1700: PRINT "CENTIGRADE.:.";
- 200 GOSUB 2100:LET C=K 210 LET F=INT(C\$1.8+32.5)
- 220 PRINT:PRINT "FAHRENHEIT.:.":F
- 230 GOTO 280
- 240 LET Y=4:GOSUB 1700: PRINT "FAHRENHEIT.:."; 250 GOSUB 2100:LET F=K
- 260 LET C=INT (5* (F-32) /9+0.5)
- 270 PRINT:PRINT "CENTIGRADE.:.":C
- 280 PRINT: PRINT: STOP

SCIENCE PROBLEMS: TUMBLEDICE



INTRODUCTION

In any game that involves throwing two dice, whether backgammon or craps, the number you get with any single roll is entirely a matter of chance. It is impossible to predict what will come up on the face of the dice.

However, the same does not hold true for a large number of throws. Here the laws of probability apply and it becomes possible to predict (roughly) what the results will be.

what the results will be.

There is a very simple explanation when you stop to think of it.

Over a hundred throws you will,
for example, obtain seven more
often than two or twelve. This is
because seven can be obtained
with six different combinations
of the dice:

5 & 2 2 & 5 4 & 3 1 & 6 Twelve can only be obtained one way – with double sixes. In other words, the odds of a seven being thrown are six times as great as

681 384

for a twelve.

Skilled players of backgammon always take the probability of different numbers into account when placing their pieces. They try to avoid moving into positions where likely dice combinations can throw them out, yet they will take greater risks where only hard to obtain

combinations can reach them.

In this program the roll of two dice is randomly simulated by the computer some 180 times

the computer some 180 times over and the distribution of the numbers thrown is plotted on the screen. The final result is also printed out.

You can compare the result to the average probability of the numbers occuring. As often as 2 and 12 are thrown:

3 & 11 are 2× as likely; 4, 5, 9 & 10 are 3× as likely; 6 & 8 are 5× as likely;

6 & 8 are 5 × as likely; 7 is 6 × as likely; Of course 1 can never occ

Of course 1 can never occur when only two dice are being rolled at the same time.

LINE NOTES

Line 110: Sets up an array to contain the numbers rolled by the dice.

Line 120: Sets up area of the text on screen.

Lines 130-150: Print on screen a list of the values the dice can take. Line 160: Establishes the initial values of the dice.

Line 170: Prints a message on screen to indicate the results of each throw.

Line 180: Calls up random numbers which simulate the throw of each dice.

Line 190: Prints each throw of the dice on screen.

Lines 200-240: Repeat the dice throw 180 times and show the results with a row of stars.

Lines 250-280: Print out the total result of all the throws in a double column list.

Line 290: Stops the program.



SCIENCE PROBLEMS: TUMBLEDICE



SCIENCE PROBLEMS: FAST BRAKES



INTRODUCTION

Driving along a country lane in your sports car you feel quite confident about doing 80km/h since the road ahead seems quite deserred. Suddenly, at the top of a rise and some ten metres ahead of you, you see a stop sign. By the time your foot hits the brakes can you still read the sign? How

while the car is stopping? By the time you have come to a complete halt how far have you gone since you first glimpsed the sign?

Stopping a car is an action that takes two separate stages — thinking about braking and actually braking. Although common sense tells us that the faster a car travels the longer it takes to stop, what is less well known is that the amount of time spent thinking varies tremendously compared to the time spent braking.

As a rule of thumb the Highway Code of Britain reckons that thinking distance is roughly 20cm for every kilometre per hour a car is moving. At 10km/h you will have gone two metres before you hit the pedal. At 100km/h this distance increases.

to 20 metres. However, braking distances rise dramatically a speed increases – from 39m at 80km/h to 76m at 112km/h (the legal speed limit). At very slow speeds you spend more time thinking than braking; at higher speeds the reverse holds true.

In the program which follows you are asked to enter different speeds of travel. These are converted by the computer into the distances you travel while thinking, braking and coming to rest. At what speed does thinking distance equal braking distance?

LINE NOTES

Line 110: Sets up the area of the text on screen.

Line 120: Prints the title of the



SCIENCE PROBLEMS: FAST BRAKES

program on the screen.

Lines 130-160: Ask you to enter the speed at which you are travelling. If this is more than 112km/h the program lets you know that you are breaking the speed limit.

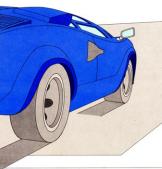
Line 170: Calculates the distance taken to stop.

Lines 180-210: Print out answers for thinking and braking distances and total stopping distance.

Lines 220-240: Plot the graphics symbols which compare distance travelled while thinking (T) with the distance travelled while braking (B), then end the program.

LISTING

- 100 REM BRAKES
- 110 BOSUB 1100 120 LET X=0:LET Y=3:GDSUB 1700:
 - PRINT "BRAKING. DISTANCES"
- 130 PRINT: PRINT "HOW. FAST. (KM/H).";
- 140 GOSUB 2100: IF K<113 THEN GOTO 170
- 150 PRINT: PRINT "OVER. THE. SPEED. LIMIT"
- 160 LET T=6:GOSUB 1400:GOTO 110
- 170 LET A=INT(K/5+0.5):
- LET B=INT (K*K/166+0.5)
- 180 PRINT: PRINT "DISTANCES. (IN.M) "
- 190 PRINT: PRINT "THINKING. "; A 200 PRINT: PRINT "BRAKING.."; B
- 210 PRINT:PRINT "TOTAL....": A+B
- 220 PRINT: FOR I=1 TO INT(A/5+0.5): PRINT "T":: NEXT I
- 230 FOR I=1 TO INT(B/5+0.5):
- PRINT "B"::NEXT I 240 PRINT: PRINT: STOP



Answers on page 118

SCIENCE PROBLEMS: PAY AS YOU GO



INTRODUCTION

Tremendous news! You have just won £25,000 in the football pools and are now faced with the happy problem of deciding whether to spend it all at once (say buying 5,000 software cassettes) or else investing it in order to become a millionaire by the time you are forty. You choose the latter option.

The question is how best to invest your money. Putting it on Lady Belle in the fifth race simply because the horse is starting at 40-1 odds takes more nerve than you can stomach.

The bank manager advises you to put everything in a deposit account earning eight percent. At this rate how many years will it take for your loan to the bank to grow to a million pounds?

By contrast, a friend who is

opening a computer shop and rates his chances of success as very high tells you he needs a loan of £25,000 to start up. He offers to pay you 30% interest provided that you leave the loan in the business for £2 years. How much will he eventually repay?

much will be eventually repay?

A third option arises when a friend of a friend, who is a bit vague about his business except to say that it involves international flights in small planes to unmarked airstrips, sacks of used notes and some very tough-looking business associates, would like to borrow your money for 2 years at 16% interest.

Which of the three loans do you choose if you want to make

your million pounds?

The thing to bear in mind is that simple interest accumulates by the same amount year by year whereas compound interest takes the interest earned each year and reinvests it together with the original sum. Interest is earned on a growing pool of money.

In the program which follows you are asked to enter the value of your investment, the interest rate and the time for which it will be loaned.



SCIENCE PROBLEMS: PAY AS YOU GO

LISTING

- 100 REM INTEREST 110 GOSUB 1100
- 120 LET X=0:LET Y=D/2:GOSUB 1700
 - 130 PRINT "SIMPLE.AND.COMPOUND":
 PRINT:PRINT "....INTEREST"
 140 LET T=3:GOSUB 1400:GOSUB 2800
- 150 LET X=0:LET Y=2:GOSUB 1700: PRINT "SIZE.OF.LOAN?.";: GOSUB 2100:LET L=K
- 160 PRINT:PRINT "INTEREST(%)?.";; GOSUB 2100:LET I=K
- GOSUB 2100:LET I=K
 170 PRINT:PRINT "TIME.IN.YEARS."::
- GOSUB 2100: LET Y=K
- 180 LET R=INT(L+L*I*Y/100+0.5)
 190 PRINT:PRINT "AMOUNT.TO.REPAY.IS"
- 200 PRINT:PRINT :R:".(SIMPLE)"
- 210 LET R=L:FOR N=1 TO Y:
- LET R=R*(1+I/100):NEXT N 220 LET R=INT(R+0.5)
- 230 PRINT:PRINT :R:". (COMPOUND)"

Answers on page 118

240 PRINT: STOP

LINE NOTES

Line 110: Sets up the area of the text on screen.

Lines 120-140: Print the title of the program on the screen.

Lines 150-170: Ask you to input the values which will be used as raw data to calculate interest payments.

Lines 180-240: Work out the amount that has to be repaid if simple interest is being calculated and then the amount of repayment if compound interest is being calculated, then stop the program when it has finished.



SCIENCE PROBLEMS: MOON



INTRODUCTION

Ever since man first looked to the heavens and observed the night sky he has been fascinated by the mysteries to be found there. Even before the invention of the most primitive telescopes, astronomers were able to recognize and name the larger and nearer planets of the solar system. Mercury, Venus, Mars, Saturn and Jupiter were all known before the 17th century. With the development of the telescope by Galileo in 1609 astronomy was transformed; although it was not until 1781 that the next planet,

Uranus, was found by Herschel. Since then the number of planets known to orbit the Sun has increased to nine and astronomy has become a science in

which mathematics and now computers, play a vital part. Probably the most prominent certainly the most striking, is the Moon. Its phases have been observed since time immemorial. From one new moon to the next, if we watch night by night, we first see a thin crescent appear. Then, over the next two weeks, the crescent gradually thickens until a full moon fills the sky. Thereafter it slowly wanes for a further two weeks until another new moon begins. The exact period between one new moon and the next is 29 days. 12 hours.

44 minutes and 2.9 seconds. We will use a simplified method here to calculate the phases (age) of the Moon, Don't worry about understanding the mathematics involved. They are rather obscure. But the result is accurate to within two days.

The program gives the Moon's age in days since the last new moon. By referring to the diagram you can find the corresponding shape it will have.

DATA

The accuracy of this program can be tested if we look at the dates of some eclipses of both the Sun and Moon. These dates can

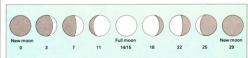
be found in astronomy books. Now it is a fact that an eclipse

object in the night sky, and of the Moon can only occur when the Moon is full. An eclipse of the Sun can only occur when the Moon is new. If you are in doubt about this, any good book on astronomy will explain why.

Listed below are eclipse dates - past and future. Can you find out which are eclipses of the Sun and which of the Moon?

- (1) 20 FEB 1924 (2) 14 JAN 1926
- (3) 27 NOV 1928
- (4) 9 MAY 1929 (5) 28 APR 1930
- (6) 2 APR 1931
- (7) 30 IIII 1935
- (8) 3 MAY 1939
- (9) 19 DEC 1945
- (10) 26 IUL 1953 (11) 25 DEC 1954
- (12) 8 IUN 1956
- (13) 30 DEC 1963
- (14) 4 IUN 1974
- (15) 18 APR 1977
- (16) 16 FEB 1980
- (17) 6 IUL 1982
- (18) 4 DEC 1983
- (19) 30 MAY 1984
- (20) 4 MAY 1985
- (21) 7 OCT 1987
 - (22) 6 AUG 1990
 - (23) 21 MAY 1993 (24) 27 SEP 1996

 - (25) 11 AUG 1999
 - (26) 16 IUL 2000



Phases of the Moon

SCIENCE PROBLEMS: MOON



LISTING

- 100 REM MOON
- 110 GOSUB 1100
- 120 LET X=0:LET Y=4:GOSUB 1700: PRINT "ENTER. THE. DAY: . ":
- 130 GOSUB 2100: LET D=K
- 140 PRINT: PRINT "NOW. MONTH. (1-12):.";
- 150 GOSUB 2100: LET M=K
- 160 PRINT:PRINT "YEAR(4 FIGURES):."; 170 GOSUB 2100: LET Y=K
- 180 LET Z=Y/19: LET Z=INT(209#(Z-INT(Z)))
- 190 LET Z=Z+6+INT(Y/400)+
- INT (Y/300) INT (Y/100) +M+D
- 200 LET A=INT(Z/30):LET Z=Z-30*A
- 210 PRINT: PRINT "MOON'S. AGE. = . ": Z: "- DAYS"
- 220 PRINT: STOP

LINE NOTES

terested.

Line 110: Sets up area of the text on screen.

Lines 120-160: Print messages asking you to enter the day, month and year (as a four-figure number) in which you are in-

Lines 170-200: Calculate the age of the Moon.

Lines 210-220: Print out the Moon's age for a particular date and stop the program.

Answers on page 118

SCIENCE PROBLEMS: PINBALL



INTRODUCTION

This program recreates an oldfashioned pinball machine that can sometimes still be found in amusement arcades or antique shops. A ball enters at the top and travels toward the bottom. Along the way it bounces against pins and rolls between them adding to the score.

The interesting thing about this sort of machine is that the chances of a ball ending up in a particular slot at the bottom of a pin table can easily be predicted by using Pascal's Triangle, a mathematical rule discovered in the 17th century.

Blaise Pascal (1623-1662), a brilliant French philosopher and mathematician, first became interested in the problem as a result of meeting the Chevalier de Mere, a renowned gambler who was less interested in numbers than the chances of throwing different combinations of dice. Pascal was able to show that a triangle of probability could be created as shown above. If this triangle is applied to a pinball table it can be used to predict the chances of a ball ending up in different slots along the bottom. For example, there is only one chance in 64 that it will end up either on the extreme right or left but 20 chances in 64 that it will fall into the centre.

Predicted results	Number of possibilities					
1	1					
1 1	2					
1 2 1	4					
1 3 3 1	8					
1 4 6 4 1	16					
1 5 10 10 5 1	32					
6 15 20 15 6 1	64					

In the program which follows Pascal's Triangle is created by the computer and tested to see if the actual results of dropping a ball 64 times are comparable to the results predicted by Pascal.

LINE NOTES Line 110: Sets up an array with

Line 110: Sets up an array seven spaces.

Line 120: Establishes the text area of the screen.

Line 130: Initializes the value of the number of trials.

Lines 140-220: Print the title of the program on the screen and also the layout of the pinball table.

Lines 230-240: Start the ball at the top of the table.

Lines 250-260: Erase the ball after it has moved on to the next



SCIENCE PROBLEMS: PINBALL

Lines 270-290: Move the ball down through the pins.

Line 300: Chooses randomly whether the ball goes right or left as it falls.

Line 310: At the position where the ball lands one is added to the score.

Lines 320-330: Print totals of each of the seven scores.

Line 340: Adds one to the counter for the number of trials.

Lines 350-410: Print the actual results after 64 trials and compare them to the predicted results; then end the program.

LISTING

- 100 REM PINBALL
- 110 DIM T(7) 120 GOSUB 1100
- 130 PRINT: LET G=0
- 140 LET X=1:LET Y=D/2:G0SUB 1700
- 150 PRINT ".....PINBALL"
- 160 LET T=2:60SUB 1400:60SUB 2800 170 LET X=8:LET Y=2:60SUB 1700:
- PRINT"+"
 180 LET X=7:LET Y=4:GOSUB 1700:
- PRINT"+.+"

 190 LET X=6:LET Y=6:GOSUB 1700:
- PRINT"+.+.+"
- 200 LET X=5:LET Y=8:GOSUB 1700: PRINT"+.+.+"
- 210 LET X=4:LET Y=10:GOSUB 1700: PRINT"+.+.+.+
- PRINT"+.+.+.+"
 220 LET X=3:LET Y=12:60SUB 1700:
- PRINT"+.+.+.+"
 230 LET X=8:LET Y=1
- 240 GOSUB 1700: PRINT"o"
- 250 LET T=.2:60SUB 1400
- 260 GDSUB 1700:PRINT"."
- 270 IF Y=13 THEN GOTO 310
- 280 LET Y=Y+2 290 IF Y=1 THEN GOTD 240
- 300 LET N=2:GOSUB 1300:LET X=X+2*R-3:
- LET Z=60+2*R:GOSUB 2400:GOTO 240 310 LET T(X/2)=T(X/2)+1
- 320 LET Y=13:LET X=1:GOSUB 1700 330 FOR I=1 TO 7:PRINT ".";T(I);:
- NEXT I 340 LET T=3:60SUB 1400:LET G=G+1:
- IF 6<64 THEN GOTO 230 350 LET T=6:GOSUB 1400:GOSUB 2800
- 360 LET X=0:LET Y=2:GOSUB 1700
- 370 PRINT"PASCAL'S.PREDICTION"
 380 PRINT:PRINT".1.6.15.20.15.6.1"
 390 PRINT:PRINT"YOUR.RESULTS"
- 400 PRINT:FOR I=1 TO 7:PRINT ".";
- 410 PRINT: PRINT: STOP

WORD PROBLEMS: CANUTE



INTRODUCTION

Nearly 1000 years ago Canute ruled as King of England, Denmark and Norway. Annoyed by the flattery of his courtiers and their awe of his power. Canute decided to teach them a lesson. He sat by the edge of the sea one day and ordered the incoming tide not to wet him. As the rising waters lapped over his feet he soon convinced his followers that even a mighty ruler such as he, at the end of the day, was merely a man.

In this game for two or more players King Canute risks another version of his famous soaking. One player, the wordchooser, secretly keys in a word of up to 13 letters. The computer then displays a picture of King Canute on the screen and, next to him, a line of blank spaces, one for each letter of the unknown word.

Now each word-seeker in turn must press a letter key. If the hidden word contains the letter that has been selected, the computer prints it in the correct space on the screen. However, bad guesses cause the tide to come in and slowly cover Canute.

The aim of the game is to discover the secret word before Canute is completely covered by the tide.

LINE NOTES

Lines 110-120: Define the text area on the screen and print the first message.

Lines 130-140: Take secret word and store it as KS

Lines 150-240: Draw figure of Canute and print the title of the

Line 250: Sends program to a subroutine

Lines 260-330: Print message to player who is guessing the word and show the number of letters it contains with a row of spaces.

Lines 340-530: Set up a loop so that with each wrong guess the tide advances one line.

Line 360: Stops program when tide covers Canute or when the secret word has been found.

Line 370: Takes the letter that has been guessed and stores it.

Lines 380-470: Check the letter against the secret word. If correct it is printed on the screen.

Line 480: Stops rising tide if the right letter is found and prepares for the next letter to be guessed.

Lines 490-520: Produce a rise in the tide of asterisks that slowly engulf Canute.

Line 550: Stops the program.

Lines 560-590: Form a subroutine that sets up the print position of the text.



WORD PROBLEMS: CANUTE

VARIATIONS

If you would like to add 'win or lose' messages to the program, then insert the following lines into the main listing:

352 IF T=# THEN GOTO 6## 354 IF N=1 THEN GOTO 7## 6## PRINT:PRINT *>>HIGH.TIDE((*

618 PRINT 628 PRINT "THE.WORD.IS ";KS

630 STOP 788 PRINT:PRINT *>>WELL.DONE<<

718 PRINT 728 IF T>5 THEN GOTO 758

730 PRINT "BUT.CANUTE" 740 PRINT "IS.VERY.NET" 750 STOP

As the program now runs King Canute vanishes under the tide after 13 wrong guesses. You can make the game easier or harder by changing the value of 13 in line 340 to a number that is greater or smaller.

For expert players, change line

528 PRINT "#####"

The person guessing the secret word will then have to remember which letters have already been

which letters have already been tried.

If you want the computer to ensure that a word of more than 13 letters cannot be chosen use

135 TE LEN(KE))13 THEN BOTO 128

the following line:

LISTING

100 REM CANUTE

120 PRINT"ENTER.A. SECRET. WORD."

130 GOSUB 2200 140 GOSUB 2800: PRINT: PRINT

150 PRINT"...MMM.....CANUTE"

160 PRINT". (+.+)....======" 170 PRINT".. (-).."

*180 PRINT"0..V..O" 190 PRINT"00HHH00"

200 PRINT"..HHH" 210 PRINT"..HHH"

220 PRINT"..===" 230 PRINT"..W.W"

240 PRINT".WW.WW" 250 GOSUB 560

260 PRINT"PICK.A.LETTER" 270 LET X=(W-LEN(K\$)+6)/2+1

280 LET Y=F+2 290 GOSUB 1700

300 FOR N=1 TO LEN(K\$) 310 PRINT"=";

320 NEXT N 330 DIM W(N-1)

340 FOR T=13 TO 0 STEP -1 350 GOSUB 560

360 IF T=0 OR N=1 THEN GOTO 540 370 GOSUB 1500

380 FOR P=1 TO LEN(K\$) 390 GOSUB 2600

400 IF G\$<>P\$ THEN GOTO 470 410 LET X=P+(W-LEN(K\$)+6)/2

420 LET Y=F+1 430 GOSUB 1700 440 PRINT G\$

450 IF W(P)=0 THEN LET N=N-1 460 LET W(P)=1

470 NEXT P 480 IF Y=F+1 THEN GOTO 350 490 LET X=0

500 LET Y=T+F-7 510 GDSUB 1700 520 PRINT G\$:"******

530 NEXT T 540 PRINT"....

560 LET X=W/2-3 570 LET Y=F+7 580 GDSUB 1700

590 RETURN
*This is the only use of 0 as a letter - not zero.

WORD PROBLEMS: LASTWORD



INTRODUCTION

Since Victorian times making word ladders has been a popular pastime. For instance, can you change 'last' to 'word' in five moves? You can only change one letter at a time, and you have to form a new four-letter word at each step on the way. One possible answer is:

LAST: 1) LOST 4) WORE 2) LOSE 5) WORD

2) LOSE 5) WO 3) LORE

Now try a more difficult word ladder such as changing CHIP to BYTE. There is an answer that takes only seven moves.

LASTWORD is a word ladder game for two or more players. The first player chooses any word with four letters. Everybody then takes it in turn to add to the ladder, starting with the first word chosen. Whoever cannot make a new word passes and mortal to the ladder of the ladder of ENTER or NEWLINE, When new word the winner is announced.

There is only one special rule to remember. If a letter has been used eight times in one position then the next player must change that letter rather than any other. For example, if a player enters 'tale' and the word list already

Answer on page 118

LISTING

100 REM LASTWORD

110 GOSUB 1100 120 PRINT"ENTER.NO..OF.PLAYERS"

130 GOSUB 2100 140 GOSUB 2900

150 PRINT"ENTER NAMES....."

160 FOR N=1 TO K 170 PRINT "PLAYER. ";N:GOSUB 2200

170 PRINT "PLAYER. "; N: GOSUB 22 180 LET V\$(N)=K\$

190 NEXT N 200 GOSUB 1300 210 DIM L (4, 26)

220 GOSUB 3000 230 LET T=0

240 GOSUB 1200 250 JE TO THEN PRINT

250 IF T>O THEN PRINT "LAST. WORD. BY: "

260 IF T>O THEN PRINT V\$(LW) 270 PRINT

280 LET R=R+1

290 IF R>K THEN LET R=1 300 PRINT"ENTER.A.WORD:"

310 PRINT V\$(R)

330 IF K\$<>"" THEN LET L=0

340 IF T=0 OR K\$="" THEN GOTO 400

contains at least seven words ending in 'e' then the computer will print 'NEW LETTER 4'. The next person to go must change the fourth letter (the e).

LINE NOTES

Line 110: Positions text area on the screen.

Lines 120–150: Print messages to the players.

Line 130: Stores the number of players as variable K.

Line 140: Sets up an array V\$ to contain the players' names.

Lines 160-190: Form a loop then print message.

which stores the players' names as elements of V\$.

Line 200: Stores a random number R. This number will determine which player starts the game.

Line 210: Sets up an array L to record the number of times the 26 letters of the alphabet are used in the four columns of the word list.

Lines 220-230: Set up an array W\$ to store up to 60 words entered by the players and a variable T to count the words.

Lines 240-250: Clear screen

WORD PROBLEMS: LASTWORD

350 FOR P=4 TO 1 STEP -1 360 BOSHB 2700 370 LET L (P.A-64)=L (P.A-64)+1 380 IF L(P, A-64) >7 THEN LET L=P 390 NEXT P 400 LET X=4+W/2 A10 EDR V=F TO F+A 420 BOSUB 1700 430 IF T+Y>F+6 THEN PRINT W\$ (T+Y-F-6) 440 NEXT Y 450 IF L>O THEN PRINT"NEW.LETTER."; STR# (L); ">. "; 460 GOSUB 2200 470 IF K\$="" THEN GOTO 520 480 LET T=T+1 490 LET WS (T)=KS 500 LET LW=R 510 LET P=0 520 LET P=P+1 530 IF PKK AND TK60 THEN GOTO 240 540 PRINT 550 PRINT V\$ (LW) : ".WINS....." SAO STOP

player who supplied the most recent last word.

*For Spectrum, change line 2900 to DIM V\$(26,7): RETURN and line 3000 to DIM W\$(60,4):RETURN

Lines 280-320: Determine the number of the next player, then print his or her name and request that a word be entered.

Lines 330-390: Form a routine that checks the last word entered for letter repetitions, Line 340 ensures that the routine is not carried out at the start of the game or if the last player has passed.

Lines 400-440: List the last 15 words entered

Line 450: Prints a message that a

Line 260: Prints the name of the | new letter must be played as the old one has already been used in a given position eight times.

> Line 460: Stores the new word entered as K\$.

> Lines 470-500: Add the new word to the list in WS, provided that K\$ is not the empty string " " which shows that someone has passed on their last turn.

Line 550: Announces winner.

Lines 530-560: Ensure the game ends when 60 words are reached, otherwise the program returns to line 240.



WORD PROBLEMS: CODE



INTRODUCTION

In this program the computer creates a random code which it substitutes for the message which you type in. Each letter is replaced by another letter so that A might become Q, B becomes T and so on.

The number of possible codes is enormous, greater than 4 with 26 zeros after it! With this program you can soon find out if you have the genius to become a master at code cracking.

Persuade a friend or member of your family to enter a secret message for you. A fairly short message of 20 letters or so works best. Only enter letters however. Numbers will not work.

The computer will display the message in code. Underneath it will print blank spaces for the decoded version. Look for one-letter words first, then two-letter words and so forth. You might deduce that Q in the code stands for A in the original. Type Q and then A. Press RETURN (or ENTIER or NEWI INF).

If the guess is wrong you will be given two penalty points. If it is right then the letter you guessed will be printed in the correct space or spaces. Continue until you have deciphered the entire message.

It may help you to know that the ten letters used most often in written English are E, T, A, O, N, R, I, S, H, D in that order. These may match up with some of the most common letters in

These may match up with some of the most common letters in the coded message.

VARIATIONS In order to bring the program to an end when the message has

been deciphered add the following lines to the listing and make the necessary replacements.

155 LET 0=# 41# IF M(P)<>32 THEN PRINT "-";:

LET Q=Q+1 578 IF HS=XS AND KS=YS THEN LET H=1:LET Q=Q-1:60TO 598

LET H=1:LET G=Q-1:50T0 598 628 IF 0>8 THEN 50T0 438 638 LET T=3:50SUB 1488:SOSUB 1288:STOP

LINE NOTES

Lines 110-120: Define the text size and position and print the first message on the screen.

Lines 130&150: Store the secret message as KS and its length as L. Lines 140&610: Set up a coun-

ter for points gained with wrong guesses.

Line 160: Clears the screen.

Lines 170-190: Set up three arrays C, J and M.

LISTING

100 REM CODE

110 GOSUB 1100

120 PRINT"ENTER. SECRET. MESSAGE"

130 PRINT: GOSUB 2200 140 LET S=0

140 LET S=0 150 LET L=LEN(K\$)

160 GOSUB 1200

170 DIM C(26)

180 DIM J(L)

200 FOR N=1 TO 26

210 LET C(N)=N+64

230 FOR N=25 TO 2 STEP -1

240 GOSUB 1300 250 LET T=C(N+1)

250 LET T=C(N+1) 260 LET C(N+1)=C(R)

270 LET C(R)=T 280 NEXT N

290 FOR P=1 TO L

300 GOSUB 2700 310 LET M(P)=A

320 IF A=32 THEN LET J(P)=A 330 IF A<>32 THEN LET J(P)=C(A-64)

340 LET R=J(P)-64:GDSUB 2500: PRINT R\$;

350 NEXT P 360 LET X=0

370 LET Y=F+2

WORD PROBLEMS: CODE

puter's code numbers (65, 66, 67, etc) for the 26 capital letters of the alphabet.

Lines 230-280: Randomly rearrange the letters within array C. Lines 290-350: Form a loop to

print the encoded message letter by letter leaving spaces in all the right places. Lines 360-470: Clear a line of

position.

620 GOTO 430

Lines 200-220: Store the com- | announcing points gained by wrong guesses.

> Lines 480-490: Store as GS the first letter you press (taken from the coded message), then asks which original letter this stands

> Line 500: Stores the second letter you press.

Lines 510-600: Print only correctly decoded letters placing them in the position they occuthe screen and reset the print pied in the original message.

Line 450: Prints the message | Line 620: Goes to next guess.

380 GOSUB 1700 390 FOR P=1 TO L 400 IF M(P)=32 THEN PRINT".": 410 IF M(P)<>32 THEN PRINT"-": 420 NEXT P 430 LET Y=F+4 440 GOSUB 1700 450 PRINT"YOU. HAVE. ": STR\$ (S): ".POINTS" 460 PRINT: PRINT"...." 470 LET Y=F+6: GOSUB 1700 480 GOSUB 1500 490 PRINT"TRY. ": G\$: ". STANDS. FOR ": 500 LET H\$=G\$:GOSUB 2200 510 LET Y=F+2 520 GOSUB 1700 530 LET H=0 540 FOR P=1 TO L 550 LET R=J(P)-64: GDSUB 2500: LET X\$=R\$ 560 LET R=M(P)-64:60SUB 2500:LET Y\$=R\$ 570 IF HS=XS AND KS=YS THEN LET H=1: GOTO 590 580 GDTD 600 590 LET X=P-1:GOSUB 1700:PRINT K\$:: LET X=0 AOO NEXT P 610 IF H=0 THEN LET S=S+2



WORD PROBLEMS: WORDGRID



INTRODUCTION

This is a game for two players using a 7 × 7 grid. Each player in turn enters a column number and then chooses any letter of the alphabet to place inside the grid. If this letter completes one or more words (whether across or down) of at least three letters. points are scored as follows:

- 3 letters = 1 point
- 4 letters = 3 points 5 letters = 5 points
- 6 letters = 10 points
- 7 letters = 20 points The illustration below shows

how a typical game would look after 12 moves

			Α		R		N				
	E		Y		E		C				
	T										
2	3	4	4	*	5	0	6	10	7	4	10

If the next player puts a B in column three, five points in all will be scored for making the words 'bet', 'barn' and 'bar'. The player must then enter this score so that the computer can keep track of and print the totals and identify the winner once the grid is full and the game has ended after 49 moves.

LINE NOTES

Line 110: Sets the text area

Line 120: Sets up an array S to record the players' scores.

Line 130: Sets up an array H to record the heights of columns one to seven.

Line 140: Introduces variable T to count the total number of letters in the grid.

Lines 150-210: Print the grid.

Line 220: Introduces variable P for the players: it takes the values 1 and 2

Line 230: Calls a subroutine (lines 490-540) that clears any messages and resets the print position.

Lines 240-250: Ask player to choose a column and store the column number as K.

Line 260: Ensures that column K is not already full with seven letters

Lines 270-290: Ask player to choose a letter and store the letter entered as KS.

Lines 300-310: Increase the letter count by one and increase the column height.

correct position. Line 370: Asks player to enter

the score

Line 380: Waits for the score to be entered.

Line 390: Adds the score to the player's total.

Line 400: Moves to the next player.

Line 410: Returns program to the next move, provided that the grid is not full.

Lines 430-480: Print the final scores and stop the program; final lines are a subroutine.



WORD PROBLEMS: WORDGRID

VARIATIONS

In this game the computer refers to the players as 1 and 2. Perhaps you could write a few extra lines to input the names of the players. You could use these names instead of the variable P in lines 240, 280, 370 and elsewhere. If you are not sure how to do this the LASTWORD program may give you some idea.

A harder version of WORD-GRID that needs no extra programming is one in which you score points for words formed in all eight directions; across, up and down, and diagonally.



LISTING

- 100 REM WORDGRID
- 120 DIM S(2)
- 130 DIM H(7)
- 140 LET T=0 150 LET X=W/2-B
- 160 FOR Y=1 TO 12
- 170 GOSUB 1700 180 PRINT"\$.....
- 190 PRINT"
- 200 GUSUB 1700 210 PRINT"**1*2*3*4*5*6*7**"
- 220 LET P=1
- 230 BOSUB 490
- 240 PRINT "PLAYER.";P;".COLUMN.";
- 250 GOSUB 2100 260 IF H(K)=7 THEN GOTO 230
- 260 IF H(K)=7 THEN 270 GOSUB 490
- 280 PRINT "PLAYER. ":P:".LETTER. ":
- 290 GOSUB 2200
- 300 LET T=T+1
- 310 LET H(K)=H(K)+1 320 LET X=W/2+K#2-B
- 330 LET Y=F-H(K) \$2+6
- 340 GOSUB 1700
- 350 PRINT K\$ 360 GOSUB 490
- 370 PRINT "PLAYER. "; P; ". SCORE. ";
- 380 GOSUB 2100 390 LET S(P)=S(P)+K
- 400 LET P=3-P 410 IF T<49 THEN GOTO 230
- 420 GOSUB 490
- 430 LET T=4:60SUB 1400:60SUB 2800
- 450 GOSUB 1700 460 PRINT"PLAYER.1.SCORED.";S(1):PRINT
- 470 PRINT"PLAYER. 2. SCORED. "; S(2): PRINT 480 STOP
- 490 LET X=W/2-10
- 510 GOSUB 1700
- 540 RETURN

WORD PROBLEMS: ANAGRAM



INTRODUCTION

Most word addicts love to solve anagrams. These puzzles, in which one word is formed from the letters of another, often appear as clues to crossword puzzles. With other anagrams, the letters are scrambled into nonsensical combinations that the puzzle solver must simply decode.

This program is designed to form anagrams from a list of up to ten words. In this game for two, one person enters the words (which may be up to seven letters long). The computer then displays the anagrams one at a time and waits for the solution to be entered. A wrong answer forces you to go again. If the answer is correct the next word in the list can be tackled (after pressing any key to continue). When all ten anagrams in the list are solved the computer displays a stop statement announcing that a round is over.

LINE NOTES

Line 110: Sets area of text on the screen

Line 120: Prints message on screen.

Line 130: Stores the number asked for in line 120 as K.

Lines 140-160: Set up arrays | Lines 400-410: Space out the VS. WS. and L.

Line 180: Prints message to first player.

Lines 190-230: Form a loop to deal with the words that are entered.

Lines 240-510: Set up a loop to display the anagrams one at a time.

Lines 250-300: Clear the screen and store the letters of the word to be jumbled as the elements of VS.

Lines 310-360: Rearrange the elements of V\$ into a jumbled sequence.

Lines 370-390: Print out the anagram.

messages on the screen.

Line 420: Prints message on screen.

Line 430: Waits for the input of the solution and stores it as KS. If the solution is correct line 450 jumps the program to a 'win' message in line 490. Otherwise the program goes to line 460 and asks the player to try again to unscramble the anagram.

Line 500: Waits for the key to be pressed. If there are further words to be unscrambled the program goes to line 510 and at once returns to line 240. If not. the program goes to line 520 and ends



WORD PROBLEMS: ANAGRAM

VARIATIONS

The anagram game can be made more exciting if it becomes a race against the clock. The extra lines below add a timing routine to the program. When you think you know what the solution to the anagram is, press the first letter of the answer to stop the clock. Only if the correct letter is entered will it come to a halt. You can then key in the whole word.

If you make a mistake, the computer will add penalty points to your score and restart the clock. The winner is the player with the fewest total points after completing ten anagrams.

```
265 LET S=8:LET T=1
481 GOSUB 1488
482 LET S=S+1
```

483 GOSUB 1688 484 IF K\$ () ** AND K\$ () P\$ THEN LET \$2529

405 LET Y=12:LET I=0:60SUB 1700 406 PRINT "YOUR.SCORE.IS.";S

487 IF K\$ () P\$ THEN GOTO 481 488 PRINT 475 LET S=S+58



LISTING

100 REM ANAGRAM 110 GOSUB 1100

120 PRINT"HOW, MANY, WORDS?...."

130 PRINT: GOSUB 2100 140 GOSUB 2900

150 GOSUB 3000 160 DIM L(K)

170 PRINT

180 PRINT"NOW.ENTER.THE.WORDS.":
PRINT

190 FOR N=1 TO K 200 GOSUB 2200

210 LET W\$(N)=K\$ 220 LET L(N)=LEN(K\$)

230 NEXT N 240 FOR J=1 TO K

250 GOSUB 1200 260 LET K\$=W\$(J)

270 FOR P=L(J) TO 1 STEP -1 280 GOSUB 2600

290 LET V\$(P)=P\$

310 FOR N=L(J)-1 TO 2 STEP -1 320 GOSUB 1300 330 LET T\$=Y\$(L(J)-N)

340 LET V\$(L(J)-N)=V\$(L(J)-R+1) 350 LET V\$(L(J)-R+1)=T\$

360 NEXT N 370 FOR P=1 TO L(J)

380 PRINT V\$(P); 390 NEXT P

430 PRINT:GOSUB 2200:LET W\$(26)=K\$
440 PRINT"...."
450 IF W\$(26)=W\$(J) THEN GOTD 490

460 PRINT "TRY AGAIN....." 470 LET T=2:GOSUB 1400:GOSUB 1200

480 PRINT:GOTO 370 490 PRINT"CORRECT.-.WELL.DONE."

500 GOSUB 1500 510 NEXT J

520 STOP

WORD PROBLEMS: WORDKING



INTRODUCTION

The mysterious King of Wordomania could be a valuable friend. However, if you make him angry you will have a dangerous enemy for life. You now have a chance to visit Wordomania and to impress its eccentric ruler.

tric ruler. The King loves presents. You may offer him gifts of any description but the King has laid down a strict rule. He will only the control of the co

what it is. As you play the game the computer will calculate your score. This score is a measure of the King's rising anger. The computer adds one point to the score for an accepted gift which contains the secret letter. A gift which doesn't, and is refused, will earn you two points. Ten points will be added to your score if you press 'Y' to say that you have guessed the King's secret letter, only to give the wrong answer. If you guess correctly you will be rewarded. If your score reaches 30 points you have lost the game and most likely will be executed.

LINE NOTES

Line 110: Sets the area of the text as it appears on screen.

Lines 120–130: Select a random number from 1 to 26 and store it as R. The number stands for the King's favourite letter.

Line 140: Changes the random number into a letter of the alphabet.

Line 150: Sets position of the screen message.

Lines 160–180: Print message on screen.

Line 190: Stores the 'gift' that the player enters as variable K\$.

Lines 200-230: Set up loop that checks K\$ letter by letter.

Lines 240-290: Print messages on screen and add to the score if the gift is refused.

Lines 310-320: Print message on the screen.

Lines 350-360: Depending on your answer the computer goes back or leaps ahead.

Lines 370–380: Print message

Line 390: Stores reply as K\$.

Line 400: If K\$ is the same as the secret letter stored at R, then the program jumps to the 'win' messages at lines 510–530.

Lines 410-450: Points are added and a 'wrong' message printed if K\$ is not the secret letter. Line 460: Sets time delay then gives the player another chance to offer a gift.

Lines 470–490: If score is more than 30 this message and the correct answer is printed. Line 540: The program ends.

VARIATIONS

If you want to make the game harder try changing 30 in line 460 to 25, or even to 20.

For expert players the King is even more demanding. Gifts must contain his favourite letter and must not contain the letter he dislikes most. Try this version:

142 LET BS=RS 144 GOSUB 1366

146 GOSUB 2500

148 IF B\$=R\$ THEN GOTO 144 195 LET F=1:LET G=8 215 IF P\$=B\$ THEN LET F=8

228 IF PS-RS THEN LET G-1 498 PRINT "LETTERS.WERE.";

Rs;",";Bs 51# PRINT "THE.LETTER.DISLIKED?" 512 GOSUB 22##

514 IF KS()BS THEN GOTO 418

Also make the following changes:

In line 250 replace P=100 by

In lines 260, 290 replace P<100 by F*G=0 In line 460 replace 30 by 50

HINT

You have probably realized that the computer treats the gifts you key in merely as strings of letters.

WORD PROBLEMS: WORDKING

If you offered the King 'RUB-BISH' or 'QBYX' instead of 'GOLD' or 'SILVER' the computer would not protest.

Assuming that you enter genuine words your gifts, no matter how unlikely, should cover all the letters of the alphabet. Here is one list of suitable presents – you could probably do even better.

JIGSAW BRONZE PICKAXE YACHT LIQUORICE SILVER GOLDFISH MUSHROOMS



LISTING

- 100 REM WORDKING
- 110 GOSUB 1100 120 LET N=26
- 130 GOSUB 1300 140 GOSUB 2500: LET S=0
- 150 GOSUB 1200:LET X=0:LET Y=6:
- 160 PRINT"WHAT.GIFT.HAVE.YOU.":PRINT 170 PRINT".BROUGHT.FOR.THE...":PRINT 180 PRINT"KING.OF.WORDOMANIA?":PRINT
- 190 GOSUB 2200
- 200 FOR P=1 TO LEN(K\$)
- 210 GOSUB 2600 220 IF P\$=R\$ THEN LET P=99
- 230 NEXT P
- 240 PRINT"....." 250 IF P=100 THEN PRINT
- ". *. GIFT. ACCEPTED. *"
- 260 IF PC100 THEN PRINT
- 270 PRINT
- 280 LET S=S+1 290 IF P<100 THEN LET S=S+1
- 300 PRINT
- 320 PRINT"KING'S.LETTER.Y/N?":PRINT
- 330 GOSUB 1500 340 GOSUB 1200:LET X=0:LET Y=4:
- GOSUB 1700
- 350 IF G\$="N" THEN GOTO 440 360 IF G\$<>"Y" THEN GOTO 310
- 370 PRINT"..ENTER.THE.KING'S..":PRINT
- 380 PRINT"..FAVOURITE.LETTER..":PRINT 390 GOSUB 2200
- 400 IF KS=RS THEN GOTO 510
- 420 GOSUB 1200 430 PRINT"BAD.LUCK.-.NOT."; K*; "..."
- 440 PRINT"YOUR.SCORE.IS."; STR*(S);"..."
- 460 LET T=2:60SUB 1400:1F S<30 THEN GOTO 150 470 PRINT"THE.KING.IS.ANGRY..":PRINT
- 480 PRINT "RUN. FOR. YOUR. LIFE.. ": PRINT 490 PRINT "THE. LETTER. WAS. "; R\$; ":
- PRINT 500 GDTD 540
- 500 GOTO 540 510 PRINT".<<QUITE CORRECT>>.":PRINT 520 PRINT"YOU.WIN.THE.FREEDOM":PRINT
- 530 PRINT"...OF.WORDOMANIA...":PRINT 540 STOP

WORD PROBLEMS: JOTTO



INTRODUCTION

JOTTO is a classic test of deductive thinking. Like a sleuth you must eliminate the wrong answers one by one until the solution to the problem is in your grasp.

In this two-player version of the game, one person keys in a secret word. The word may be of any length, but four or five letters are probably the best.

The other player then tries to discover the word taking as few turns as possible. With each turn a test word of the same length as the secret word is entered. The computer prints the score for this guess on the screen. It gives one point for each correct letter that has been found regardless of its position.

The score acts as a clue to the correct letters. However, you may find that even when you get a full score (say 5 with a five-letter word) you may not have won. In this case you will have won in this case you will have continues until the correct word is found with all the letters in proper order.

LINE NOTES

Line 110: Sets the text area.

Line 120: Sets up an array VS in which the secret word is stored.

Line 270: E right length.



Line 130: Prints a message on the screen.

Lines 140-160: Store the secret word as variable A\$ with its length given as L.

Lines 170-180: Clear screen and print message giving length of test words that may be tried.

Lines 190–200: Introduce variables T, the number of turns, and S, the score for each guess.

Lines 210-250: Store the letters of the secret word A\$ as elements of the array V\$.

Line 260: Stores the test word as K\$.

Line 270: Ensures that K\$ is the right length.

Lines 290-320: Print KS.

Line 330: If KS is the secret word then this line jumps the program to the final 'win' messages at the end.

Lines 340–420: Compare K\$ letter by letter with the secret word stored in V\$. Whenever a match is found, line 380 adds one point to the score.

Line 430: Puts the score on the screen.

Line 450: Returns the program to line 200 and resets array VS for the next test word.

Lines 460-480: Print 'win' message and report the number of tries taken, then end the program.

WORD PROBLEMS: JOTTO



VARIATIONS

Some players of JOTTO enjoy a scoring system which helps them to determine whether or not letters are in their correct positions. If you add the following lines to the program the computer will score two points for a correct letter in the correct position, but only one point for a correct letter in a wrong position.

332 FOR P=1 TO L 334 BOSUB 2588

336 IF PS=VS(P) THEN LET S=S+1 TTR NETT P

LISTING

100 REM JOTTO 110 GOSUB 1100 120 GOSUB 2900

130 PRINT"ENTER.A. SECRET. WORD. " 140 PRINT: 605UR 2200

150 LET A\$=K\$ 160 LET L=LEN(A\$):GOSUB 1200

170 LET X=0:LET Y=6:GOSUB 1700 180 PRINT"TRY, ": STR\$ (L):

"-LETTER. WORDS .. " 190 LET T=0

200 LET S=0 210 LET KS=AS 220 FOR P=1 TO L

230 GOSUB 2600 240 LET V\$ (P) =P\$

250 NEXT P 260 LET X=0:LET Y=8:GOSUB 1700:

PRINT"....": GOSUB 1700: GOSUB 2200 270 IF LEN(K\$)<>L THEN GOTO *260

280 LET T=T+1 290 LET X=0:LET Y=10:GOSUB 1700

300 PRINT ". ": K\$ 310 LET X=0:LET Y=12

320 GOSUB 1700 330 IF K\$=A\$ THEN GOTO 460

340 FOR P=1 TO L 350 BOSUB 2600

360 FOR N=1 TO L 370 IF P\$<>V\$(N) THEN GOTO 410

380 LET S=S+1 390 LET V\$(N)="#"

400 LET N=L 410 NEXT N

420 NEXT P 430 PRINT"SCORE. "; STR\$ (S); ". "

440 PRINT: PRINT: PRINT 450 GOTO 200 ". TURNS .. "

460 GOSUB 1200: PRINT"CORRECT. " 470 PRINT: PRINT"YOU, TOOK, ": STR\$ (T):

480 STOP

WORD PROBLEMS: SQUARES



INTRODUCTION

Sam Loyd's 14-15 Puzzle was the Rubik Cube of the 1870s. It consisted of square blocks numbered from 1 to 15 and set in a 4 × 4 grid. It was sold to the public with the empty block positioned in the bottom right-hand corner. Blocks 1 to 13 were in the correct sequence but numbers 14 and 15 were interchanged. The goal of the puzzle was to slide the blocks into the correct order ending with the space back in its original position. Loyd offered a large prize for the first correct solution. He soon succeeded in driving the world half mad for, as it turned out, the puzzle was mathematically impossible.

In this computer version of Sam Loyd's puzzle we use 15 letters instead of numbers. The aim of the game is to arrange the letters in alphabetical order, left to right and starting from the top, in as few moves as possible. Don't worry — the program is organised so that the computer will produce only puzzles that can be solved.

To move a letter into the space simply use the keys L (Left), R (Right), U (Up), and D (Down).

LINE NOTES

Line 110: Sets area of text.

Line 120: Determines the size of the grid; in this case 4 × 4.

Lines 130–190: Set up two arrays, V\$ and W\$, each with 15 letters of the alphabet and then a space.

Lines 200-250: Produce a random rearrangement of the 15 letters in WS.

Line 260: Calls a subroutine (lines 450 to 540) which prints the elements of W\$ in the 4 × 4 grid. The variable P stores the location of the empty space.

Line 270: Introduces S to count the number of moves taken.

Line 280: Introduces M to record the movements of the space as it travels forwards or backwards along array WS.

Line 290: When player presses a key, stores the move as G\$.

Lines 300–370: Carry out the required letter movement. Here, lines 340–350 interchange the space with the letter.

Lines 380–440: Check letters in WS against VS and print number of moves taken if puzzle has been solved, then stop the program.

Lines 450-540: Subroutine which puts letters into the grid.



WORD PROBLEMS: SQUARES

LISTING

100 REM SOLIARES

110 GOSUB 1100

120 IFT 0=4 130 GDSUB 2900

140 GOSUB 3000

150 FOR R=1 TO Q*Q-1

160 GDSUB 2500

170 LET V\$ (R)=R\$ 180 NEXT R

190 LET V\$ (Q\$Q) ="+"

200 FOR N=Q*Q-2 TO Q-1 STEP -1

210 GOSUB 1300 220 LET T\$=V\$ (N+1)

230 LET V\$ (N+1)=V\$ (R)

240 LET V\$ (R) =T\$

250 NEXT N 260 GOSUB 450

270 LET S=0

280 LET M=0

290 GDSUB 1500

300 IF G\$="U" AND P<Q\$Q-Q+1 THEN

LET M=Q 310 IF G\$="D" AND P>Q THEN LET M=-Q 320 IF G\$="L" AND P/Q<>INT(P/Q) THEN

LET M=1

330 IF G\$="R" AND (P+Q-1)/Q

<>INT((P+Q-1)/Q) THEN LET M=-1 340 LET V\$(P)=V\$(P+M)

350 LET V\$ (P+M) ="+"

360 IF MC>0 THEN LET S=S+1

370 GOSUB 450

380 FOR N=1 TO 0*0-1 390 IF V\$(N)<>CHR\$(N+64) THEN LET N=30

400 NEXT N

410 IF N=31 THEN GOTO 280

420 PRINT 430 PRINT "YOU, TOOK, ": STR\$ (S):

". MOVES "

440 PRINT: STOP

450 LET N=1 460 FOR Y=F-3 TO F+Q\$2-5 STEP 2

470 FOR X=W/2-4 TO W/2+Q\$2-6 STEP 2 480 GDSUB 1700

490 PRINT V\$ (N)

500 IF V\$(N)="+" THEN LET P=N 510 LET N=N+1

520 NEXT X

530 NEXT Y

540 RETURN



VARIATIONS

If you would like to change the size of the grid to 3×3 or to $5 \times$ 5 then alter the value of Q in line 120 to 3 or 5.

By adding the following lines to the program the computer will print out your best score after a sequence of games.

125 LET 8=10000

448 IF S(8 THEN LFT 8=S

441 PRINT '2FST.SCORE. = . ": B: "....." 442 FRINT "ANOTHER. GAME-Y. OR. N?"

447 BOSHR 1588

444 IF GS="N" THEN STOP 445 IF 6\$(>"Y" THEN 60TO 443

446 GOSUB 1288

447 GOTO 15#

If you want to convince yourself that some versions of the puzzle really are impossible to solve then replace Q-1 in line 200 with Q.

WORD PROBLEMS: LETTERS



INTRODUCTION

This is a solo variation of a popular family word game. In this version the computer and the player choose letters in turn, although it is the player whose letters in turn, although it is the player who decides where every letter is placed in the 5×5 grid. The object of the game is to fill the grid with as many high-scoring words as possible, either vertically or horizontally, and so win the highest score.

highest score. Each letter is chosen and then placed in the grid by entering the row number first and then the column number. The player may cover over letters that were previously placed but if this happens more than once there will be space left in the grid at the end of the game.

Once the grid is filled, points are calculated for completed words in each line and column as follows:

> 3 letter word: 3 points 4 letter word: 5 points 5 letter word: 10 points

You can only score points for one word in any single line or column, even if this word contains shorter words within it. Therefore the maximum possible score that can be achieved is 100. However, any score over 50 is excellent.

LISTING

100 REM LETTERS 110 GOSUB 1100

120 LET X=0:LET Y=1

130 GOSUB 1700 140 PRINT"..1.2.3.4.5"

150 FOR N=1 TO 5

160 PRINT

170 PRINT; STR\$ (N)

190 LET N=26

200 FOR T=13 TO 1 STEP -1

220 GOSUB 2500: LET K\$=R\$

230 LET X=0:LET Y=12:G0SUB 1700 240 PRINT"MY.CHDICE.IS.":K4:"....."

250 GOSUB 410

260 LET X=0:LET Y=12:GOSUB 1700

270 PRINT"YOUR.CHOICE.IS?."; 280 GOSUB 2200

290 GOSUB 410 300 NEXT T 310 FOR C=1 TO 5 320 GOSUB 480

LINE NOTES

Lines 120-180: Print the grid of row numbers and column numbers in the right position on the screen,

Lines 200-300: Set up a loop that is executed 13 times, during which the grid is filled with letters.

Selects a random

number R.

Line 210:

Line 220: Assigns to K\$ the letter of the alphabet determined by the random number R.

Line 240: Prints message showing computer's choice of letter. Line 250: Calls a subroutine (lines 410–520) in which the player is asked to enter the line and column of the position he has chosen for KS.

Lines 270-290: Require the player's choice of letter to be input, then call subroutine 410 again.

Lines 310-380: Help the player to calculate the score for the game.

Lines 390-400: Print the total score and end the game.

Lines 410-520: Form a subroutine that asks the player to enter line and column instructions, then prints the letter in the correct position, clears the bot-

WORD PROBLEMS: LETTERS

330 PRINT"LINE.":C:".SCORE?" 340 GOSUB 2100

350 PRINT"COLUMN. ": C: ". SCORE?"

360 GOSUB 2200 370 LFT T=T+K+VAL (K\$)

380 NEXT C

390 PRINT"YOUR. SCORE. IS. "; T

400 STOP 410 LET L\$=K\$:PRINT"LINE...";

420 GOSUB 2100

430 LET Y=1+K#2 440 PRINT "COLUMN. ":: GOSUB 2100

450 LET X=K#2

460 GDSUB 1700

470 PRINT LS: 480 LET X=0:LET Y=12:GOSUB 1700

490 FOR N=1 TO 3 500 PRINT"....."

510 NEXT N 520 RETURN

tom messages off the screen and I 'best score' routine to this pronext letter choice.

VARIATIONS

You may want to increase your chances of making a big score. If so, change the initial value of T in line 200 to 14 or 15. The bigger the value of T the more chances you will have to replace useless letters with ones that make up words.

For less experienced players it might be a good idea to stop the computer choosing difficult letters such as O or I. Add:

225 IF R=10 OR R=17 THEN **GOTO 210**

If you are going to play several games, you could try adding a

repositions the cursor for the gram. If you are not sure how to do it look at the program lines suggested for SQUARES. But take care. In that game the best score was the lowest. In this game you want to record your

highest score.





ANSWERS

THINKING GAMES

Dungeon

The combinations of numbers which ensure that the princess survives are:

Princess 1; Monster 4 Princess 2; Monster 2, 5, 8, 9 Princess 3; Monster 1, 5, 8 Princess 4; Monster 4 Princess 5; Monster 1, 5, 6, 8 Princess 6; Monster 2, 6, 9 Princess 7: Monster 7

Princess 8; Monster 1, 5, 9 Princess 9; Monster 1 Sum Trouble

Basil's answer: 1681 Cynthia's answer: 9376 and the result of multiplying it by itself is 87,909,376

Darrell's answer: 54 years for himself and 45 years for his wife. Ernest's answer: 168 right and 132 answers wrong.

Primes
1) 25 including the number 2
2) For 100–200: 21 primes

For 300–400: 16 primes Primes decrease with higher number ranges.

3) The common difference is 210. The numbers are: 199, 409,

619, 829, 1039, 1249, 1459, 1669, 1879, 2089. 4) 1983 – not prime

1983 – not prime 1979 – last prime 1987 – next prime

5) 1327–1361; the longest gap is 34 years. 6) 2, 3, 5, 7, 11, 101, 131, 151, 181, 191, 313, 353, 383 are

the smallest palindrome primes.

7) The largest prime under a

million is 999,983.

Hannah and Eve discovered that the lowest perfect square palindromes, which do not have square roots that are palindromes too, are: 676 (or 26²)

> 69696 (or 264²) 94249 (or 307²)

The lowest perfect square that is a palindrome and also has an

even number of digits was found by Anna and Otto to be:

698896 (or 836²) Crossnumber

Age is 73; year of birth is 1907.

2) The number of pieces of

nougat is either 63 (3969 being the square) or 87 (7569 being the square).

 The original number of employees was 30 – the present number is 960.



Midsummer's Eve

There are just four values of the square root which could fit the bill. If the letters are re-substituted for the numbers all but one can be eliminated.

'TWELVE' SQUARE ROOT 256,036 506 = WLE 361,201 601 = WVE 439,569 663 = VVW

456,976 676 = EVE The square root of 'TWELVE' is 'EVE' therefore – did you spot

NUMBER PUZZLES

Magic Squares 16 2 3 13 5 11 10 8 9 7 6 12 4 14 15 1

the clue in the title?

SCIENCE PROBLEMS

Conversion

The innkeeper's wife is a towering 6ft 6in.

However, there is no need to

worry about the fleas; they can only leap eight inches.

The cord the innkeeper's wife offers you is almost half an inch thick – useless for bootlaces.

s Fast Brakes By the time you hit the brakes

you will have travelled 16m – well past the sign. While braking you continue another 39m. By the time you come to a full stop you will have gone 55m in all since you first saw the sign.

Pay As You Go Provided that you leave the in-

Provided that you leave the interest you earn in the account it will only take 48 years for your money to grow to a million pounds.

Your friend will have paid you £212,500 in interest and you get the initial loan back as well making £237,500 in all.

The friend of a friend will repay you £1,021,856 after 25 years — and of course your original loan as well!

Moon

Eclipses of the Moon occur on dates numbered: 1, 3, 6, 8, 9, 10, 13, 14, 17, 20, 21, 22, 24and 26. Eclipses of the Sun occur on dates numbered: 2, 4, 5, 7, 11, 12, 15, 16, 18, 19, 23 and 25.

WORD PROBLEMS

Lastword

CHIP CHIN COIN CORN BORN BORE BYRE BYTE (7 moves).

THE RAINBOW BOOK OF BASIC PROGRAMS

GAMES, PROBLEMS AND PUZZLES FOR PECTRUM, BBC, ELECTRON. **DRAGON AND VIC20 COMPUTERS** 盘 魚