

# 50

**COMPLETE  
PROGRAMS**

**for Texas Instruments  
TI-99/4A Computer**

**BLUE MOUNTAIN COMPUTER, INC.**

# **50 Complete Programs for Texas Instruments TI-99/4A Computer**

**BLUE MOUNTAIN COMPUTER, INC.**

This book was developed by  
Blue Mountain Computer, Inc.  
6818 Woodstream Circle  
Seabrook, Maryland 20706

©1983 by Blue Mountain Computer, Inc.  
All rights reserved. No part of this book may be  
reproduced in any form or by any means without per-  
mission in writing from the publisher.

Printed in the United States of America

#### IMPORTANT

Blue Mountain Computer, Inc. makes no expressed or implied warranty with regard to the program materials offered or their merchantability or their fitness for any particular purpose. The program materials are made available solely on an "as is" basis.

In no event shall Blue Mountain Computer, Inc. be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the furnishing, use, or performance of the program materials. The sole and exclusive liability to Blue Mountain Computer, Inc., regardless of the form of action, shall not exceed the purchase price of the book.

The word TI and TI-99/4A Computer are registered trademarks of Texas Instruments Incorporated. Texas Instruments, Inc. was not in any way involved in the preparation, writing or reviewing of this book. Use of the term TI should not be construed to represent any endorsement, official or otherwise, by Texas Instruments Incorporated.

# Contents

Introduction .....	5
--------------------	---

## Education

Arithmetics .....	6
Calendar .....	8
Change .....	10
Morse Code .....	12
Polish Notation .....	15
Scramble .....	18
Shopping List .....	20
Typing Lesson .....	23

## Finance

Break Even Analysis .....	25
Interest Rate .....	28
Loan Payment .....	30
Trend Forecasting .....	32

## Fun and Games

Biorhythms .....	34
Calling the Largest Number .....	36
Card Game .....	39
Chomp .....	41
Craps .....	43
Cryptography .....	45
Gold Bug .....	48
Guess Number .....	50
Hangman .....	51
Knight's Tour .....	55
Lottery .....	58
Moon Landing .....	61
Mouse and Cheese .....	64
Nicoma .....	66
Nimb .....	68
On the Money .....	70

Piglatin .....	72
Skiing .....	74
Timer .....	76
Tower of Hanoi .....	78
Typing Monkey .....	81

## Graphics

Christmas Tree .....	83
Colorful .....	84
Flag .....	85
3D Graph .....	87

## Mathematics

Base Conversion .....	88
Differential Equations .....	90
Factorial .....	92
Integration-Gaussian .....	94
Integration-Simpson .....	96
Linear Equations .....	98
Magic Square .....	101
Matrix Multiplication .....	103
Monte Carlo .....	106
Pascal Triangle .....	108
Prime Factors .....	110
Prime Numbers .....	111
Zeros of Functions .....	112

# Introduction

Congratulations! You have just found 50 exciting programs for your Texas Instruments TI-99/4A Computer. No accessories are needed for your computer in order to run the programs in this book, although a cassette tape recorder is highly recommended for storing programs.

All 50 programs are specifically written for TI-99/4A Computer. This book is designed to help you get the most from your computer. It provides programs to solve real world problems in finance and mathematics, as well as educational, graphics, and games programs of general interest.

This book is useful for both the novice and the person who has a good knowledge of computers. The programs include interesting techniques which you may find useful in writing your own programs. We hope this book will help you understand and enjoy your computer for many years to come.

# Arithmetics

This program teaches addition, subtraction and multiplication. The computer randomly selects two numbers and asks you for their sum, difference or product. Unless you give the correct answer, the computer will not let you go. To terminate this program, hold the key FCTN down, then press CLEAR.

## Sample Run

```
>RUN
WHAT IS 8+3 ?
? 11
RIGHT
WHAT IS 4*9 ?
? 35
WHAT IS 4*9 ?
? 36
RIGHT
WHAT IS 92-1 ?
? 91
RIGHT
```

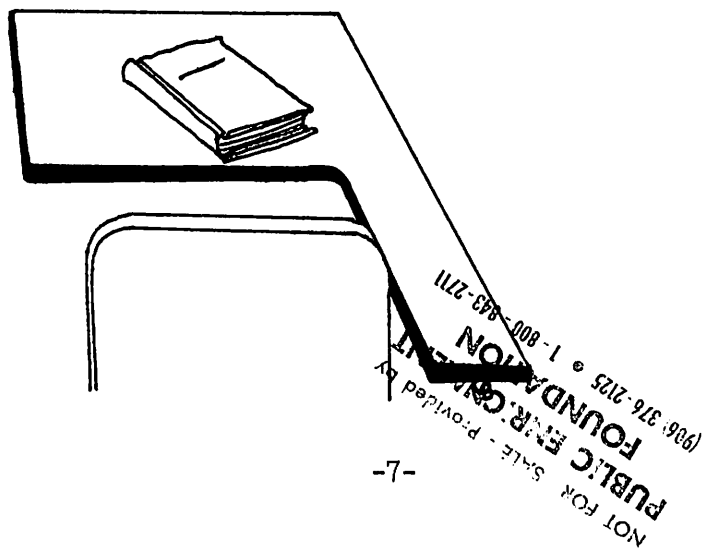
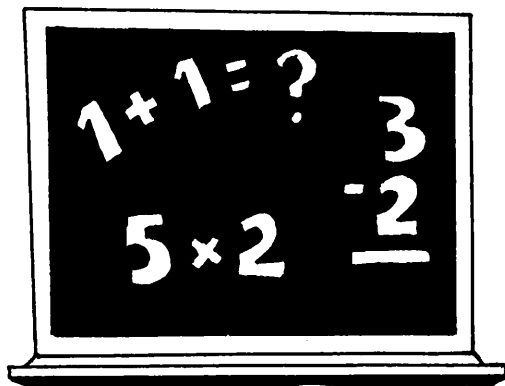
## Program Listing

```
100 REM ARITHMETICS
110 CALL CLEAR
120 RANDOMIZE
130 PRINT :::
140 I=INT(RND*100)
150 J=INT(RND*10)
160 ON INT(RND*3)+1 GOTO 170,200,230
```

```

170 PRINT "WHAT IS";I;"+";J;"?"
180 INPUT IPJ
190 IF IPJ=(I+J)THEN 260 ELSE 170
200 PRINT "WHAT IS";I;"-";J;"?"
210 INPUT IMJ
220 IF IMJ=(I-J)THEN 260 ELSE 200
230 PRINT "WHAT IS";I;"*";J;"?"
240 INPUT ITJ
250 IF ITJ=(I*J)THEN 260 ELSE 230
260 PRINT "RIGHT"
270 GOTO 130
280 END

```





# Calendar

Enter any year from 1700 to 2100. Then enter month and date. This program will tell you the day of the week. If you want to continue the program, press Y. Press any other key will terminate the program.

## Sample Run

```
>RUN
YEAR=?1983
MONTH=?5
DAY=?7
IT IS A SATURDAY
WANT TO CONTINUE?N
** DONE **
```

## Program Listing

```
100 REM CALENDAR
110 CALL CLEAR
120 DIM D$(7)
130 D$(1)="SUNDAY"
140 D$(2)="MONDAY"
150 D$(3)="TUESDAY"
160 D$(4)="WEDNESDAY"
170 D$(5)="THURSDAY"
180 D$(6)="FRIDAY"
190 D$(7)="SATURDAY"
200 INPUT "YEAR=?":Y
210 IF Y<1700 THEN 200
220 IF Y>2100 THEN 200
230 INPUT "MONTH=?":M
```

```

240 IF M<1 THEN 230
250 IF M>12 THEN 230
260 INPUT "DAY=?":D
270 I=D-621049
280 I1=Y*365.25
290 I2=30.6*(M+1)
300 IF M>2 THEN 330
310 I1=I1-365.25
320 I2=I2+12*30.6
330 I=I+INT(I1)+INT(I2)
340 IF Y>1800 THEN 370
350 IF M>2 THEN 370
360 I=I+1
370 IF Y>1900 THEN 400
380 IF M>2 THEN 400
390 I=I+1
400 I=I-INT(I/7)*7
410 PRINT "IT IS A ";D$(I+1)
420 INPUT "WANT TO CONTINUE?":A$
430 B$=SEG$(A$,1,1)
440 IF B$="Y" THEN 200
450 END

```



# Change

After entering the price of an item and the amount of payment, this program will print out the change.

## Sample Run

```
>RUN
PRICE OF ITEM?5.27
AMOUNT OF PAYMENT?10
AMOUNT OF CHANGE

ONE ONE ONE ONE
HALF-DOLLAR
DIME DIME
PENNY PENNY PENNY
** DONE **
```

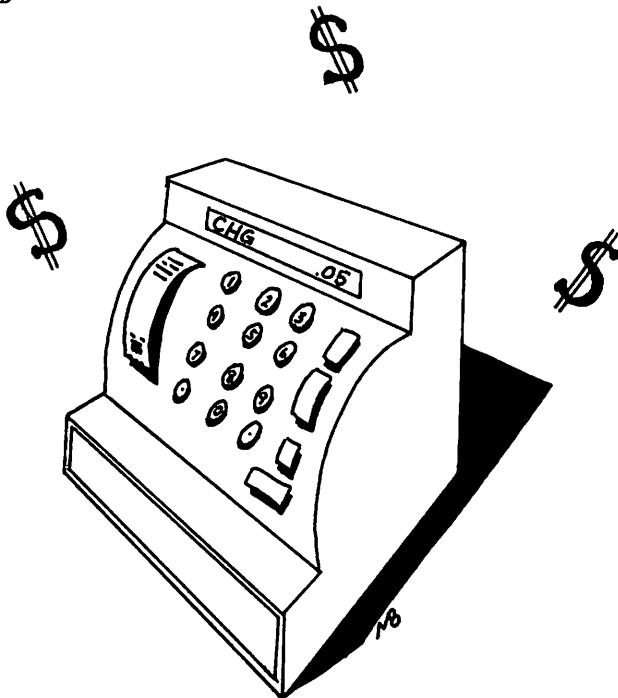
## Program Listing

```
100 REM CHANGE
110 CALL CLEAR
120 DATA HUNDRED,100,FIFTY
130 DATA 50,TEN,10,FIVE,5
140 DATA ONE,1,HALF-DOLLAR
150 DATA 0.5,QUARTER,0.25
160 DATA DIME,0.1,NICKEL
170 DATA 0.05,PENNY,0.01
180 DATA A$(10),B(10),R(10)
190 FOR I=1 TO 10
200 READ A$(I),B(I)
210 NEXT I
```

```

220 INPUT "PRICE OF ITEM?":P
230 INPUT "AMOUNT OF PAYMENT?":A
240 C=A-P
250 IF C<0 THEN 220
260 FOR I=1 TO 10
270 R(I)=INT(C/B(I))
280 C=C-R(I)*B(I)
290 NEXT I
300 PRINT "AMOUNT OF CHANGE",,,
310 FOR I=1 TO 10
320 FOR J=1 TO R(I)STEP 1
330 PRINT A$(I);" ";
340 NEXT J
350 IF R(I)=0 THEN 370
360 PRINT
370 NEXT I
380 END

```



# Morse Code

This program decodes Morse Code characters and prints out the corresponding English characters. The Morse Code .- is for character A, and -... is for character B. You can figure out the rest from program statements 120 to 180. Leave one blank between characters and another blank between words. To input a dash in Morse Code, hold down the SHIFT key, then press the key / .

## Sample Run

```
>RUN
  ENTER A LINE . . .
      .
      .
? -- --- .-. ... . -.- --- ... .
MORSE CODE
PRESS C TO CONTINUE
      .
      .
```

## Program Listing

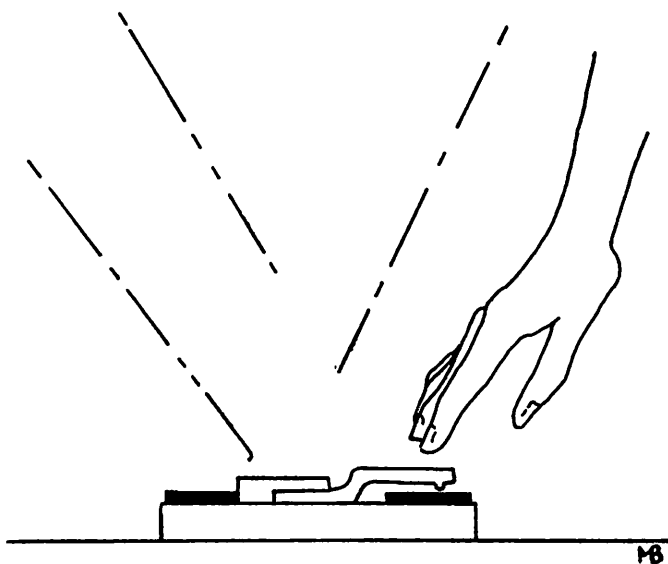
```
100 REM MORSE CODE
110 CALL CLEAR
120 DATA B.-B-...B-.-.B-..B
130 DATA .B...-B--.B....B
140 DATA ..B.---B-.B-...B
150 DATA --B-.B---B.--.B
160 DATA ---.B.-.B...B-B
170 DATA ...B...-B.--B
```

```

180 DATA -...-B-...-B---BB
190 MC$=""
200 FOR I=1 TO 7
210 READ A$
220 MC$=MC$&A$
230 NEXT I
240 CA$=""
250 J=65
260 FOR I=1 TO LEN(MC$)
270 CA$=CA$&CHR$(J)
280 IF SEG$(MC$,I,1)<>"B" THEN 330
290 IF J=32 THEN 330
300 J=J+1
310 IF J<=90 THEN 330
320 J=32
330 NEXT I
340 PRINT "ENTER A LINE OF MORSE
      CODE WITH BLANK BETWEEN CHARACTERS
      AND ANOTHER BLANK BETWEEN WORDS"
350 PRINT
360 INPUT ME$
370 ME$=ME$&" "
380 P=1
390 MG$=""
400 IF P>LEN(ME$) THEN 540
410 L=POS(ME$," ",P)
420 C$=""
430 IF P=L THEN 450
440 C$=SEG$(ME$,P,L-P)
450 C$="B"&C$&"B"
460 P=L+1
470 L=POS(MC$,C$,1)
480 IF L=0 THEN 490 ELSE 510
490 CR$="?"
500 GOTO 520
510 CR$=SEG$(CA$,L,1)
520 MG$=MG$&CR$
530 GOTO 400
540 PRINT MG$

```

```
550 PRINT
560 PRINT "PRESS C TO CONTINUE"
570 CALL KEY(0,KEY,STATUS)
580 IF STATUS=0 THEN 570
590 IF KEY=67 THEN 340
600 END
```



# Polish Notation

The most common computer representation for an arithmetic expression is known as postfix Polish notation, invented by the Polish mathematician Jan Lukasiewicz. The syntax of a conventional expression is Operand Operator Operand, also known as infix notation. The syntax for postfix notation is Operand Operand Operator.

This program will translate any conventional (infix) expression into postfix (Polish) notation. Operand must be denoted by single letter.

## Sample Run

```
>RUN
ENTER CONVENTIONAL EXPRESSION
? ((A+B)/(C-D))*F
POLISH NOTATION
A B + C D - / F *
```

## Program Listing

```
100 REM POLISH NOTATION
110 CALL CLEAR
120 DIM STK$(100)
130 PTR=1
140 PRINT "ENTER CONVENTIONAL EXPRESSION"
150 PRINT
160 INPUT IN$
170 G$=""
180 I=0
190 I=I+1
```



```

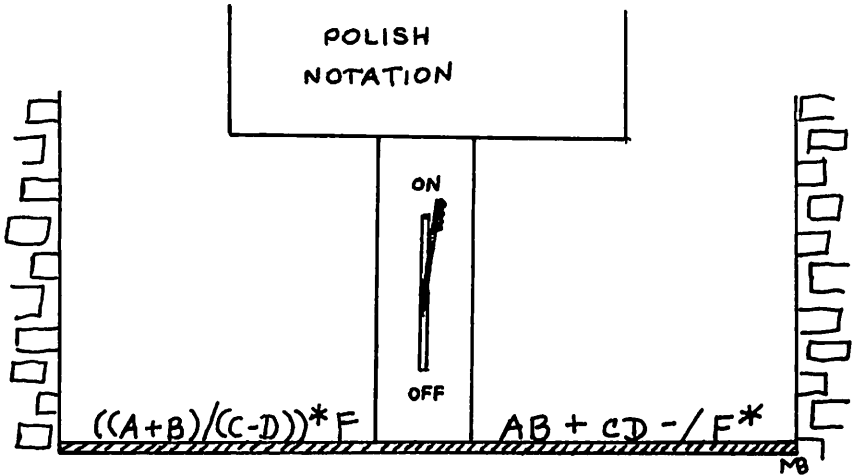
200 IF I>LEN(IN$)THEN 540
210 N$=SEG$(IN$,I,1)
220 N=ASC(N$)
230 IF (N>64)*(N<91)THEN 240 ELSE 260
240 G$=G$&N$&" "
250 GOTO 190
260 IF N=40 THEN 270 ELSE 300
270 STK$(PTR)=N$
280 PTR=PTR+1
290 GOTO 190
300 IF N=41 THEN 320
310 GOTO 380
320 IF STK$(PTR-1)<>"(" THEN 330
    ELSE 360
330 PTR=PTR-1
340 G$=G$&STK$(PTR)&" "
350 GOTO 320
360 PTR=PTR-1
370 GOTO 190
380 IF PTR=1 THEN 390 ELSE 410
390 PT=0
400 GOTO 440
410 L$=STK$(PTR-1)
420 GOSUB 620
430 PT=P
440 L$=N$
450 GOSUB 620
460 PG=P
470 IF PT>PG THEN 480 ELSE 510
480 PTR=PTR-1
490 G$=G$&STK$(PTR)&" "
500 GOTO 380
510 STK$(PTR)=N$
520 PTR=PTR+1
530 GOTO 190
540 FOR I=PTR-1 TO 1 STEP -1
550 G$=G$&STK$(I)&" "
560 NEXT I
570 PRINT

```

```

580 PRINT "POLISH NOTATION"
590 PRINT
600 PRINT G$
610 STOP
620 L=ASC(L$)
630 IF L=40 THEN 640 ELSE 660
640 P=0
650 RETURN
660 IF (L=43)+(L=45) THEN 670 ELSE 690
670 P=1
680 RETURN
690 IF (L=42)+(L=47) THEN 700 ELSE 710
700 P=2
710 RETURN
720 END

```



# Scramble

This program will randomly select a word defined in statements 130-160. It scrambles the word and then asks you what it is. You may put new words into statements 130-160. The numbers in statements 110 and 120 should be equal to the total number of words defined in statements 130-160.

To terminate this program, hold down the key FCTN, then press the key CLEAR.

Sample Run

>RUN

```
      FDNU
ANSWER ?FUND
RIGHT
```

Program Listing

```
100 REM SCRAMBLE
110 DIM A$(12),R$(12)
120 N=12
130 DATA APPLE,BOOK,DISH
140 DATA EAST,FUND,GIRL
150 DATA HORSE,IVORY,JACK
160 DATA KEY,LAMP,MONEY
170 FOR I=1 TO N
180 READ A$(I)
190 NEXT I
200 RANDOMIZE
```

```

210 J=INT(RND*N)+1
220 CALL CLEAR
230 B$=A$(J)
240 L=LEN(B$)
250 FOR I=1 TO L
260 R$(I)=SEG$(B$,I,1)
270 NEXT I
280 FOR I=1 TO 3
290 K=INT(RND*L)+1
300 M=INT(RND*L)+1
310 T$=R$(K)
320 R$(K)=R$(M)
330 R$(M)=T$
340 NEXT I
350 N$=""
360 FOR I=1 TO L
370 N$=N$&R$(I)
380 NEXT I
390 IF N$=B$ THEN 280
400 FOR I=1 TO L
410 A=ASC(R$(I))
420 CALL HCHAR(10,10+I,A)
430 NEXT I
440 INPUT "ANSWER ?":AN$
450 IF AN$=B$ THEN 460 ELSE 480
460 PRINT "RIGHT"
470 GOTO 490
480 PRINT "WRONG, IT IS ":B$
490 FOR I=1 TO 600
500 NEXT I
510 GOTO 210
520 END

```

# Shopping List

This program tests your ability of memorizing items from a shopping list. Each time the computer displays 5 items for a few seconds, then asks you to repeat them. It will then tell you how many mistakes you made.

## Sample Run

```
>RUN
YOUR SHOPPING LIST
FISH
EGG
BUTTER
VEGETABLE
OIL
WHAT ARE THOSE ITEMS?
EGG
BUTTER
SOAP
COKE
OIL
YOU MADE 2 MISTAKES
THE LIST WAS
.
.
.
```

## Program Listing

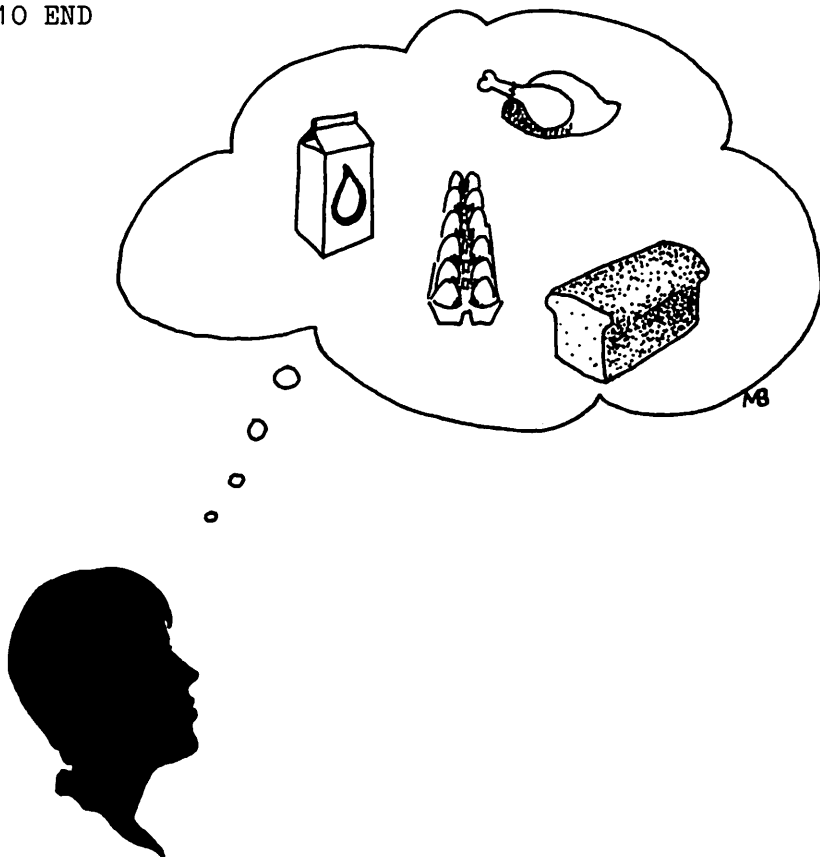
```
100 REM SHOPPING LIST
110 CALL CLEAR
```

```

120 DATA MILK,BEEF,SOAP,COKE
130 DATA BREAD,BUTTER,WINE
140 DATA ORANGE,SAUCE,NOODLE
150 DATA RICE,COOKIE,FISH
160 DATA CHICKEN,CREAM
170 DATA VEGETABLE,OIL
180 DATA EGG,TEA,GINGER
190 DIM A$(20),B(5),C$(5)
200 DIM D$(5)
210 FOR I=1 TO 20
220 READ A$(I)
230 NEXT I
240 RANDOMIZE
250 R=INT(RND*20)+1
260 B(1)=R
270 FOR I=2 TO 5
280 R=INT(RND*20)+1
290 FOR J=1 TO I-1
300 IF R=B(J)THEN 280
310 NEXT J
320 B(I)=R
330 NEXT I
340 PRINT "YOUR SHOPPING LIST"
350 FOR I=1 TO 5
360 R=B(I)
370 C$(I)=A$(R)
380 PRINT A$(R)
390 NEXT I
400 FOR I=1 TO 1000
410 NEXT I
420 CALL CLEAR
430 PRINT "WHAT ARE THOSE ITEMS?"
440 FOR I=1 TO 5
450 INPUT D$(I)
460 NEXT I
470 CT=0
480 FOR I=1 TO 5
490 FOR J=1 TO 5
500 IF C$(I)=D$(J)THEN 530

```

```
510 NEXT J
520 GOTO 540
530 CT=CT+1
540 NEXT I
550 PRINT "YOU MADE";5-CT;"MISTAKES"
560 PRINT
570 PRINT "THE LIST WAS"
580 FOR I=1 TO 5
590 PRINT C$(I)
600 NEXT I
610 END
```



# Typing Lesson

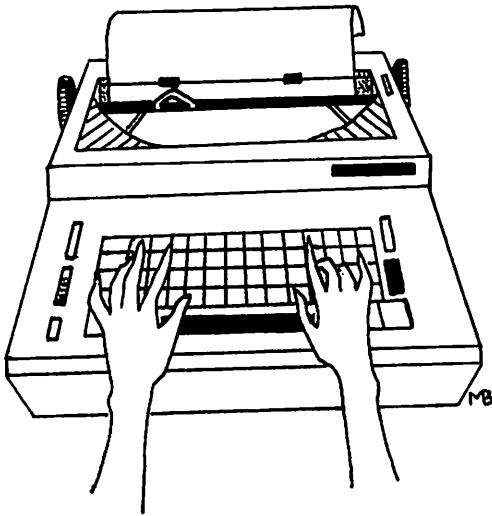
This program will display four lines of randomly selected letters and blanks. You should type letter or blank as pointed by the ^ symbol. Any error will be warned by a beep tone. No back space is allowed. At the end, computer will display the total number of errors. Also, press Y if you want to get another typing lesson. Press any other key will terminate the program.

## Program Listing

```
100 REM TYPING LESSON
110 RANDOMIZE
120 CALL CLEAR
130 FOR I=4 TO 19 STEP 5
140 FOR J=7 TO 26
150 N=65+INT(RND*27)
160 IF N<>91 THEN 180
170 N=32
180 CALL HCHAR(I,J,N,1)
190 NEXT J
200 NEXT I
210 CT=0
220 FOR I=4 TO 19 STEP 5
230 FOR J=7 TO 26
240 CALL HCHAR(I+2,J,94,1)
250 CALL GCHAR(I,J,N)
260 CALL KEY(0,KEY,STATUS)
270 IF STATUS=0 THEN 260
280 IF KEY=N THEN 320
290 IF KEY=141 THEN 260
300 CALL SOUND(10,294,2)
310 CT=CT+1
```



```
320 CALL HCHAR(I+2,J,KEY,1)
330 NEXT J
340 NEXT I
350 PRINT "NUMBER OF ERRORS=";CT
360 PRINT "DO YOU WANT TO CONTINUE?"
370 CALL KEY(0,KEY,STATUS)
380 IF STATUS=0 THEN 370
390 IF KEY=89 THEN 120
400 END
```



# Break Even Analysis

Base on the following information provided by the user,

F=Total Fixed Cost  
V=Variable Cost Per Unit  
S=Sales Price Per Unit  
U=Expected Sales in Units,

This program performs the following calculations:

Expected Sales in Dollars= $S*U$   
Break Even Point in Units= $F/(S-V)$   
Break Even Point in Dollars= $F*S/(S-V)$   
Margin of Safety Ratio= $(U-(F/(S-V)))/U$   
Profit or Loss= $(S-V)*U-F$

Sample Run

>RUN

.  
.  
.

1 TOTAL FIXED COST= 0  
2 VARIABLE COST PER UNIT= 0  
3 SALES PRICE PER UNIT= 0  
4 EXPECTED SALES IN UNITS= 0

ENTER SELECTION, 1 THROUGH 4

1

TOTAL FIXED COST=1000

.  
.  
.

2  
 VARIABLE COST PER UNIT=2  
 .  
 .  
 .  
 3  
 SALE PRICE PER UNIT=4  
 .  
 .  
 .  
 4  
 EXPECTED SALES IN UNITS=2000  
 EXPECTED SALES=\$ 8000  
 BREAK EVEN POINT= 500 UNITS  
 BREAK EVEN POINT=\$ 2000  
 MARGIN OF SAFETY RATIO= .75  
 PROFIT OR LOSS=\$ 3000  
 .  
 .  
 .

# Program Listing

```

100 REM BREAK EVEN ANALYSIS
110 CALL CLEAR
120 F=0
130 V=0
140 S=0
150 U=0
160 D=0
170 R=0
180 BEPU=0
190 MU=0
200 BEPD=0
210 MD=0
220 IF (S=V)THEN 240
230 BEPU=F/(S-V)

```

```

240 IF U=0 THEN 260
250 MU=(U-BEPU)/U
260 IF S=0 THEN 280
270 R=(S-V)/S
280 IF R=0 THEN 300
290 BEPD=F/R
300 IF D=0 THEN 320
310 MD=(D-BEPD)/D
320 PD=D*R-F
330 CALL CLEAR
340 PRINT "EXPECTED SALES=$";D
350 PRINT "BREAK EVEN POINT=";BEPU;"UNITS"
360 PRINT "BREAK EVEN POINT=$";BEPD
370 PRINT "MARGIN OF SAFETY RATIO=";MD
380 PRINT "PROFIT OR LOSS=$";PD,,,
390 PRINT "1 TOTAL FIXED COST=";F
400 PRINT "2 VARIABLE COST PER UNIT=";V
410 PRINT "3 SALES PRICE PER UNIT=";S
420 PRINT "4 EXPECTED SALES IN UNITS=";U,,,
430 INPUT "ENTER SELECTION, 1 THROUGH 4":I
440 IF (I>4)+(I<1)THEN 430
450 ON I GOTO 460,480,500,520
460 INPUT "TOTAL FIXED COST=":F
470 GOTO 220
480 INPUT "VARIABLE COST PER UNIT=":V
490 GOTO 220
500 INPUT "SALE PRICE PER UNIT=":S
510 GOTO 220
520 INPUT "EXPECTED SALES IN UNITS=":U
530 D=S*U
540 GOTO 220
550 END

```

# Interest Rate

This program deals with the same problem as in the Loan Payment program. In this program, assume the amount of the loan, the number and amount of periodic payments are known, then the program will calculate the periodic interest rate.

If you borrow 9100 dollars, and you pay back the loan in 48 months with monthly payment of 222.33 dollars, then the monthly interest rate is 0.0067002934%.

## Sample Run

```
>RUN
AMOUNT OF LOAN 9100
PERIODIC PAYMENT?222.33
NUMBER OF COMPOUNDING PERIODS?48
10
5.
2.5
1.25
.
.
.
.0067002934

PERIODIC INTEREST RATE IS
.0067002934
```

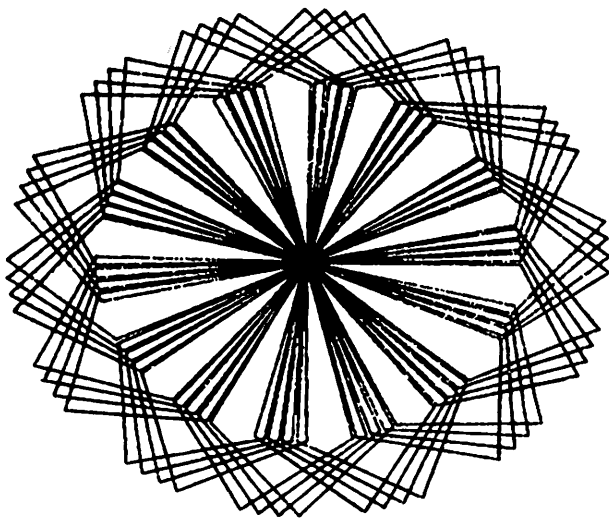
## Program Listing

```
100 REM INTEREST RATE
```

```

110 DEF F(I)=PV-PMT*(1-(1+I)^(-N))/I
120 INPUT "AMOUNT OF LOAN ":PV
130 INPUT "PERIODIC PAYMENT?":PMT
140 INPUT "NUMBER OF COMPOUNDING PERIODS?":N
150 L=1.0E-10
160 H=10
170 IF (F(L)*F(H)<=0.0)THEN 200
180 PRINT "REENTER INPUT VALUES"
190 GOTO 120
200 IF (ABS(H-L)<=1.0E-10)THEN 290
210 X=(H+L)/2
220 PRINT H
230 IF (F(X)*F(H)>=0.0)THEN 260
240 L=X
250 GOTO 270
260 H=X
270 GOTO 200
280 PRINT ::
290 PRINT ::
300 PRINT "PERIODIC INTEREST RATE IS";H
310 END

```



# Loan Payment

Let PV be the amount of the loan. This loan will be paid back in N equal payments (PMT). Let the interest rate per compounding period be I%. Then the following relationship holds:

$$PMT = PV * (I/100) / (1 - (1 + I/100)^{-N})$$

For example, assume that you borrow 9100 dollars. The annual interest rate is 8%. This loan will be repaid in 48 equal monthly payments. In this case, the monthly interest rate is  $I = 8/12 = 0.67$  percent. As you can see from the sample run, your monthly payment is 222.32 dollars.

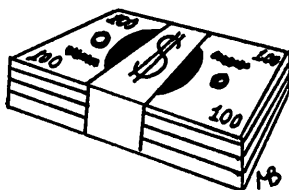
## Sample Run

```
>RUN
WHAT IS THE AMOUNT OF THE LOAN? 9100
WHAT IS THE NUMBER OF COMPOUNDING
PERIODS? 48
WHAT IS THE PERIODIC INTEREST RATE? 0.67
PERIODIC PAYMENT IS
222.3284954
```

## Program Listing

```
100 REM LOAN PAYMENT
110 INPUT "WHAT IS THE AMOUNT OF THE
    LOAN?":PV
120 INPUT "WHAT IS THE NUMBER OF COMPOUNDING
    PERIODS?":N
```

```
130 INPUT "WHAT IS THE PERIODIC INTEREST  
    RATE?":I  
140 I=I/100  
150 PMT=PV*I/(1-(1+I)^(-N))  
160 PRINT "PERIODIC PAYMENT IS":PMT  
170 STOP  
180 END
```





# Trend Forecasting

Suppose that you are running a successful business. For each of the first four months this year, your profits are \$2550, \$3579, \$4490 and \$5600, respectively. You may use this program to forecast profits for the following two months. This program is based on the least-square regression methodology.

## Sample Run

>RUN

NO. OF KNOWN DATA POINTS?4

NO. OF DATA POINTS TO FORECAST?2

DATA POINT 1 ? 2550

DATA POINT 2 ? 3579

DATA POINT 3 ? 4490

DATA POINT 4 ? 5600

LEAST-SQUARE FORECASTING

PERIOD	ACTUAL	FORECAST
1	2550	2545.6
2	3579	3551.7
3	4490	4557.8
4	5600	5563.9
5	?	6570
6	?	7576.1

## Program Listing

```
100 REM TREND FORECASTING
110 CALL CLEAR
120 DIM Y(100)
130 SI=0
```

```

140 SSI=0
150 SYI=0
160 SIYI=0
170 INPUT "NO. OF KNOWN DATA POINTS?":M
180 INPUT "NO. OF DATA POINTS TO
    FORECAST?":N
190 FOR I=1 TO M
200 PRINT "DATA POINT";I;
210 INPUT YI
220 Y(I)=YI
230 SI=SI+I
240 SSI=SSI+I*I
250 SYI=SYI+YI
260 SIYI=SIYI+I*YI
270 NEXT I
280 N1=SSI*SYI-SI*SIYI
290 N2=-SI*SYI+M*SIYI
300 DELTA=M*SSI-SI*SI
310 C1=N1/DELTA
320 C2=N2/DELTA
330 PRINT "LEAST-SQUARE FORECASTING"
340 PRINT "PERIOD  ACTUAL  FORECAST"
350 FOR I=1 TO M
360 PRINT I;TAB(9);Y(I);TAB(17);C1+C2*I
370 NEXT I
380 FOR I=M+1 TO M+N
390 PRINT I;TAB(10);"?";TAB(17);C1+C2*I
400 NEXT I
410 END

```

# Biorhythms

The biorhythm theory assumes that human body functions according to some inner clocks with constant cycles. At birth, these cycles start in a positive direction. The physical cycle (23-day) relates with physical vitality. The sensitivity (28-day) cycle relates sensitivity, creativity and cheerfulness. The cognitive (33-day) cycle relates with alertness, memory and judgement.

Enter the year, month and date of your birth and of today. This program will output the levels of these three indicators. Positive values ( $0 \leq x \leq 1$ ) are energetic times. Negative values are recuperative periods. Zero values are regarded as accident prone days.

Sample Run

```
>RUN
WHAT YEAR WERE YOU BORN?1960
MONTH?6
DAY?3
THIS YEAR IS ?1983
MONTH ?5
TODAY ?7
PHYSICAL VITALITY LEVEL=
-.0853448295
SENSITIVITY LEVEL=
-.5671228279
MENTAL ALERTNESS LEVEL=
.9627925265
** DONE **
```

## Program Listing

```
100 REM BIORHYTHMS
110 INPUT "WHAT YEAR WERE YOU BORN?":Y
120 INPUT "MONTH?":M
130 INPUT "DAY?":D
140 GOSUB 290
150 I1=I
160 INPUT "THIS YEAR IS ?":Y
170 INPUT "MONTH ?":M
180 INPUT "TODAY ?":D
190 GOSUB 290
200 N=I2-I
210 N=N*6.28318
220 P=SIN(N/23)
230 S=SIN(N/28)
240 C=SIN(N/33)
250 PRINT "PHYSICAL VITALITY LEVEL=":P
260 PRINT "SENSITIVITY LEVEL=":S
270 PRINT "MENTAL ALERTNESS LEVEL=":C
280 STOP
290 I=D-621049
300 I1=Y*365.25
310 I2=30.6*(M+1)
320 IF M>2 THEN 350
330 I1=I1-365.25
340 I2=I2+12*30.6
350 I=I+INT(I1)+INT(I2)
360 RETURN
370 END
```

# Calling the Largest Number

You first specify the number of cards. The computer then secretly writes down one number on each card. One at a time, the computer reveals to you the number on each card. After examining each number, you have to decide if that number is the largest. If so, you press the key Y to stop the game. Otherwise, you press any other key to continue. If you press key Y, the entire set of numbers is revealed to you immediately. If your choice is correct, you win 3 dollars, otherwise you lose 1 dollar.

## Sample Run

```
>RUN
YOU ARE GIVEN 100 DOLLARS
ENTER NUMBER OF CARDS 10
WHEN YOU SEE THE LARGEST NUMBER, PRESS Y
CARD # 1 = 63
CARD # 2 = 92
    (press Y)

THE REST OF THE CARDS

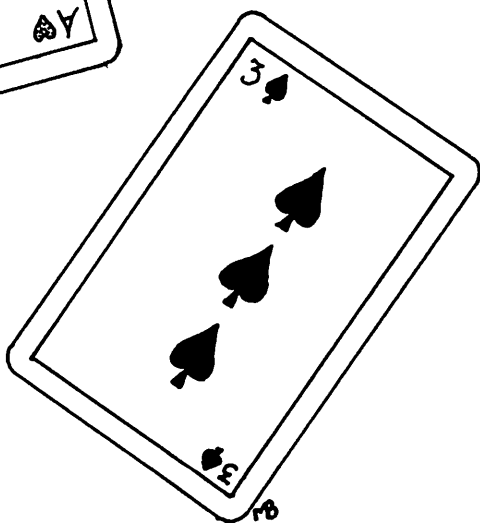
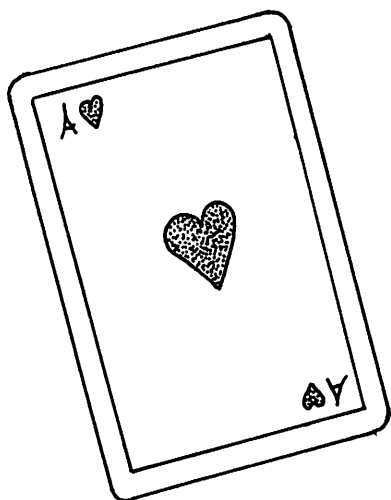
CARD # 3 = 52
    .
    .
    .
CARD # 6 = 93
    .
    .

YOU LOSE 1 DOLLAR
TOTAL AMOUNT=$ 99
TO CONTINUE PRESS SPACE BAR
```

## Program Listing

```
100 REM CALLING THE LARGEST NUMBER
110 RANDOMIZE
120 CALL CLEAR
130 DIM R(1000)
140 T=100
150 PRINT "YOU ARE GIVEN";T;"DOLLARS"
160 PRINT
170 INPUT "ENTER NUMBER OF CARDS ":N
180 PRINT
190 PRINT "WHEN YOU SEE THE
    LARGEST NUMBER, PRESS Y"
200 PRINT
210 FOR I=1 TO N
220 R(I)=INT(RND*10^2)+1
230 NEXT I
240 FOR I=1 TO N
250 PRINT "CARD #";I;"=";R(I)
260 CALL KEY(0,KEY,STATUS)
270 IF STATUS=0 THEN 260
280 IF KEY=69 THEN 340
290 NEXT I
300 T=T-1
310 PRINT
320 PRINT "YOU LOSE 1 DOLLAR"
330 GOTO 460
340 PRINT
350 PRINT "THE REST OF THE CARDS"
360 PRINT
370 FOR J=I+1 TO N
380 PRINT "CARD #";J;"=";R(J)
390 NEXT J
400 FOR J=1 TO N
410 IF R(I)<R(J) THEN 300
420 NEXT J
```

```
430 T=T+3
440 PRINT
450 PRINT "YOU WIN 3 DOLLARS"
460 PRINT "TOTAL AMOUNT=$";T
470 PRINT
480 PRINT "TO CONTINUE PRESS
    SPACE BAR"
490 PRINT
500 CALL KEY(0,KEY,STATUS)
510 IF STATUS=0 THEN 500
520 IF KEY=32 THEN 210
530 END
```



# Card Game

You start with 100 dollars. Each time the computer draws two cards face up. You have the option to bet or not to bet depending on if you think the value of the next card will be between the first two. The game is over when you do not have any money left.

## Sample Run

```
>RUN
NOW YOU HAVE $ 100
YOUR TWO CARDS ARE
      2      4
WHAT IS YOUR BET?10
THIRD CARD IS
3
YOU WIN
NOW YOU HAVE $ 110
YOUR TWO CARDS ARE
      4      5
WHAT IS YOUR BET?0
CHICKEN
.
.
.
```

## Program Listing

```
100 REM CARD GAME
110 DIM CARDS$(14)
120 CARDS$(11)="JACK"
130 CARDS$(12)="QUEEN"
```



```

140 CARD$(13)="KING"
150 CARD$(14)="ACE"
160 FOR I=2 TO 10
170 CARD$(I)=STR$(I)
180 NEXT I
190 CALL CLEAR
200 T=100
210 PRINT "NOW YOU HAVE $";T
220 RANDOMIZE
230 N1=INT(13*RND)+2
240 N2=INT(13*RND)+2
250 IF N1>=N2 THEN 230
260 PRINT "YOUR TWO CARDS ARE";TAB(5);
    CARD$(N1);TAB(12);CARD$(N2)
270 PRINT
280 INPUT "WHAT IS YOUR BET?":M
290 IF M=0 THEN 300 ELSE 320
300 PRINT "CHICKEN"
310 GOTO 230
320 IF M<=T THEN 350
330 PRINT "YOU ONLY HAVE $";T
340 GOTO 280
350 N3=INT(13*RND)+2
360 PRINT "THIRD CARD IS":CARD$(N3)
370 PRINT
380 IF (N3>N1)*(N3<N2)THEN 430
390 T=T-M
400 IF T>0 THEN 210
410 PRINT "THAT'S ALL, FOLKS"
420 STOP
430 PRINT "YOU WIN"
440 T=T+M
450 GOTO 210
460 END

```

# Chomp

This program displays a cookie with 5 rows and 7 columns. The piece on the upper left corner is poisonous. Two players alternately chomp the cookie from lower right by specifying a row number and a column number. The one who eats the poisonous piece loses.

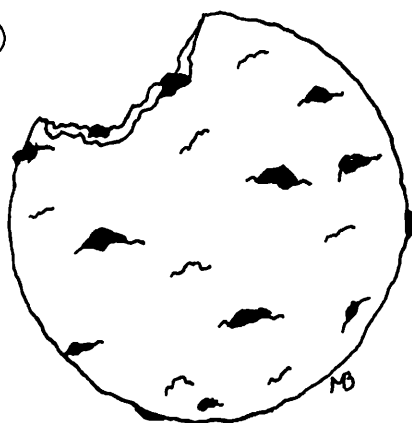
## Program Listing

```
100 REM CHOMP
110 CALL CLEAR
120 FOR J=10 TO 22 STEP 2
130 A=(J/2)+44
140 CALL HCHAR(8,J,A)
150 NEXT J
160 FOR I=10 TO 18 STEP 2
170 A=(I/2)+44
180 CALL HCHAR(I,8,A)
190 NEXT I
200 FOR I=10 TO 18 STEP 2
210 FOR J=10 TO 22 STEP 2
220 CALL HCHAR(I,J,42)
230 NEXT J
240 NEXT I
250 CALL HCHAR(10,10,80)
260 M1$="PLAYER 1'S MOVE="
270 M2$="PLAYER 2'S MOVE="
280 M$=M1$
290 P=1
300 FOR I=1 TO LEN(M$)
310 A=ASC(SEG$(M$,I,1))
320 CALL HCHAR(22,3+I,A)
330 NEXT I
340 GOSUB 610
```

```

350 IF (KEY>53)+(KEY<49)THEN 300
360 CALL HCHAR(22,22,KEY)
370 ROW=KEY*2-88
380 GOSUB 610
390 CALL HCHAR(22,23,44)
400 GOSUB 610
410 IF (KEY>56)+(KEY<49)THEN 300
420 CALL HCHAR(22,24,KEY)
430 COL=KEY*2-88
440 CALL GCHAR(ROW,COL,A)
450 IF A=32 THEN 300
460 IF A=80 THEN 470 ELSE 490
470 PRINT " PLAYER ";P;"LOST"
480 STOP
490 FOR I=ROW TO 18 STEP 2
500 FOR J=COL TO 22 STEP 2
510 CALL HCHAR(I,J,32)
520 NEXT J
530 NEXT I
540 IF P=1 THEN 550 ELSE 580
550 P=2
560 M$=M2$
570 GOTO 300
580 P=1
590 M$=M1$
600 GOTO 300
610 CALL KEY(0,KEY,STATUS)
620 IF STATUS=0 THEN 610
630 RETURN
640 END

```



# Craps

You start with 100 dollars. You are asked to bet before you roll two dice. If your first roll is a 7 or 11, you win. If your first roll is a 2, 3 or 12, you lose. If your first roll is a 4, 5, 6, 8, 9 or 10, then this number is called your "point". If you roll a point on your first roll, then you must continue rolling until you either roll your point again or roll a 7. If you make your point, you win. If you roll a 7, then you lose. To terminate this program before you run out of money, hold the key FCTN down, then press CLEAR.

## Sample Run

```
>RUN
NOW YOU HAVE $100 TO PLAY
HOW MUCH DO YOU BET?10
PRESS ANY KEY TO ROLL
YOU ROLL 2 AND 6
AND YOUR POINT IS 8
PRESS ANY KEY TO ROLL
YOU ROLL 2 AND 6
AND MAKE YOUR POINT
NOW YOU HAVE $110 TO PLAY
```

## Program Listing

```
100 REM CRAPS
110 CALL CLEAR
120 RANDOMIZE
130 T=100
140 IF T=0 THEN 150 ELSE 170
```

```

150 PRINT "THAT'S ALL, FOLKS"
160 STOP
170 PRINT "NOW YOU HAVE $";T;" TO PLAY",,
180 INPUT "HOW MUCH DO YOU BET?":M
190 IF (M<=0)+(M>T)THEN 180
200 N1=INT(7*RND)+1
210 N2=INT(7*RND)+1
220 N=N1+N2
230 PRINT "PRESS ANY KEY TO ROLL",,,
240 CALL KEY(0,KEY,STATUS)
250 IF STATUS=0 THEN 240
260 IF (N=7)+(N=11)THEN 270 ELSE 300
270 PRINT "YOU ROLL";N1;"AND";N2;"AND WIN",,,
280 T=T+M
290 GOTO 140
300 IF (N<4)+(N=12)THEN 310 ELSE 340
310 PRINT "YOU ROLL";N1;"AND";N2;"AND LOSE",,,
320 T=T-M
330 GOTO 140
340 PRINT "YOU ROLL";N1;"AND";N2;"AND
    YOUR POINT IS";N,,,
350 N1=INT(7*RND)+1
360 N2=INT(7*RND)+1
370 P=N1+N2
380 PRINT "PRESS ANY KEY TO ROLL",,,
390 CALL KEY(0,KEY,STATUS)
400 IF STATUS=0 THEN 390
410 IF P=N THEN 420 ELSE 440
420 PRINT "YOU ROLL";N1;"AND";N2;
    "AND MAKE YOUR POINT"
430 GOTO 280
440 IF (P=7)THEN 450 ELSE 470
450 PRINT "YOU ROLL";N1;"AND";N2;
    "AND LOSE",,,
460 GOTO 320
470 PRINT "YOU ROLL";N1;"AND";N2;
    "AND CONTINUE",,,
480 GOTO 350
490 END

```

# Cryptography

The concept of secret writing has been in existence as far back as written language existed. To preserve the security of private information, you can employ a special form of process called data encryption. Data encryption scrambles sensitive information using a sequence of characters called keyword. The input to the encoding algorithm are your plaintext (your original message in unscrambled form) and the keyword. The encoding algorithm scrambles the message. The output is scrambled data, called ciphertext.

The encryption process is reversible. You can decipher the scrambled message if you have the keyword and the decoding algorithm to reverse the process to get the plaintext.

In the sample run, we use TOY as the keyword. If you want to terminate this program, hold down the key FCTN, then press CLEAR.

## Sample Run

```
>RUN
 1 ENTER KEYWORD
 2 ENTER PLAINTEXT
 3 ENTER CIPHERTEXT

ENTER SELECTION, 1 THROUGH 3
 1
KEYWORD=?TOY
ENTER SELECTION, 1 THROUGH 3
 2
PLAINTEXT=?THIS IS FUN
KEYWORD= TOY
```

PLAINTEXT  
THIS IS FUN  
CIPHERTEXT  
ELAZBPNCCLR  
(press any key to continue)

### Program Listing

```
100 REM CRYPTOGRAPHY
110 CALL CLEAR
120 PRINT "1 ENTER KEYWORD":
130 PRINT "2 ENTER PLAINTEXT":
140 PRINT "3 ENTER CIPHERTEXT":
150 INPUT "ENTER SELECTION, 1 THROUGH 3 ":S
160 S=INT(S)
170 IF S<1 THEN 110
180 IF S>3 THEN 110
190 ON S GOTO 200,280,480
200 INPUT "KEYWORD=?":KD$
210 K1=1
220 FOR I=1 TO LEN(KD$)
230 C$=SEG$(KD$,I,1)
240 N=ASC(C$)
250 K1=K1*N
260 NEXT I
270 GOTO 110
280 IF LEN(KD$)=0 THEN 110
290 INPUT "PLAINTEXT=?":EG$
300 CP$=""
310 RANDOMIZE K1
320 FOR I=1 TO LEN(EG$)
330 R=INT(RND*27)
340 E$=SEG$(EG$,I,1)
350 E=ASC(E$)
360 F=0
370 IF E>90 THEN 400
380 IF E<65 THEN 400
```

```

390 F=E-64
400 G=(F+R)-INT((F+R)/27)*27
410 H=32
420 IF G=0 THEN 440
430 H=G+64
440 H$=CHR$(H)
450 CP$=CP$&H$
460 NEXT I
470 GOTO 660
480 IF LEN(KD$)=0 THEN 110
490 INPUT "CYPHERTEXT=?":CP$
500 EG$=""
510 RANDOMIZE K1
520 FOR I=1 TO LEN(CP$)
530 R=INT(RND*27)
540 H$=SEG$(CP$,I,1)
550 H=ASC(H$)
560 G=0
570 IF H=32 THEN 590
580 G=H-64
590 F=G-R+27-INT((G-R+27)/27)*27
600 E=32
610 IF F=0 THEN 630
620 E=F+64
630 E$=CHR$(E)
640 EG$=EG$&E$
650 NEXT I
660 PRINT " ":"KEYWORD=",KD$
670 PRINT " ":"PLAINTEXT"
680 PRINT " ":"EG$
690 PRINT " ":"CIPHERTEXT"
700 PRINT " ":"CP$
710 PRINT " ":"PRESS ANY KEY TO CONTINUE"
720 CALL KEY(0,KEY,STATUS)
730 IF STATUS=0 THEN 720
740 GOTO 110
750 END

```



# Gold Bug

You may speculate in the gold market without risking a penny. This program does just that. Every-day you have the option of buying, selling or doing nothing. Gold price varies from day to day. How long does it take to double your initial investment of one hundred thousand dollars?

## Sample Run

```
>RUN
DAY NO: 1
.
1 TO BUY GOLD
2 TO SELL GOLD
3 DO NOTHING
ENTER SELECTION, 1 THROUGH 3
1
AMOUNT OF BUYING IN OUNCES?20

DAY NO: 2
GOLD PRICE:$ 495
AMOUNT OF GOLD: 20 OUNCES
CASH:$ 91000
TOTAL WORTH:$ 100900
```

## Program Listing

```
100 REM GOLD BUG
110 RANDOMIZE
120 D=1
```

```

130 C=100000
140 G=0
150 P=450
160 T=P*G+C
170 CALL CLEAR
180 PRINT "DAY NO: ";D
190 PRINT
200 PRINT "GOLD PRICE:$";P
210 PRINT
220 PRINT "AMOUNT OF GOLD:";
    G;"OUNCES"
230 PRINT
240 PRINT "CASH:$";C
250 PRINT
260 PRINT "TOTAL WORTH:$";T
270 PRINT
280 IF T<=100 THEN 490
290 PRINT "1 TO BUY GOLD"
300 PRINT "2 TO SELL GOLD"
310 PRINT "3 DO NOTHING"
320 INPUT "ENTER SELECTION, 1 THROUGH 3":S
330 S=INT(S)
340 IF (S<1)+(S>3)THEN 320
350 ON S GOTO 360,410,450
360 INPUT "AMOUNT OF BUYING IN OUNCES?":N
370 IF (N*P)>C THEN 160
380 C=C-N*P
390 G=G+N
400 GOTO 450
410 INPUT "AMOUNT OF SELLING IN OUNCES?":N
420 IF N>G THEN 160
430 G=G-N
440 C=C+N*P
450 P=P+INT(RND*99)-49
460 IF P<1 THEN 450
470 D=D+1
480 GOTO 160
490 PRINT "SORRY, YOU ONLY HAVE $";T
500 END

```

# Guess Number

This program plays the game of guessing numbers between 0 and a limit specified by the user. Computer also helps you with some clues.

## Sample Run

```
>RUN
LARGEST NUMBER=10
YOUR GUESS=5
TOO LARGE
YOUR GUESS=3
TOO LOW
YOUR GUESS=4
YOU GOT IT
```

## Program Listing

```
100 REM GUESS NUMBER
110 CALL CLEAR
120 RANDOMIZE
130 INPUT "LARGEST NUMBER=":N
140 R=INT(RND*N)+1
150 INPUT "YOUR GUESS=":G
160 IF G>R THEN 170 ELSE 190
170 PRINT "TOO LARGE",,,,
180 GOTO 150
190 IF G=R THEN 200 ELSE 220
200 PRINT "YOU GOT IT",,,,
210 GOTO 130
220 PRINT "TOO LOW",,,,
230 GOTO 150
240 END
```

# Hangman

This program plays the popular word game of Hangman. It randomly selects a word from statements 160 to 190. You may put other words in these statements, as long as the total number of words is correctly put in statements 260 and 270. The computer then displays a few dashes and asks you to guess the letters for the word. This program allows you 11 chances to guess the letters. Moreover, even if you are hung, no rope is drawn on the screen.

## Sample Run

>RUN

-----

LETTERS USED

YOUR GUESS?A

.

.

A-----

LETTERS USED

A

YOUR GUESS?I

.

.

A-----

LETTERS USED

AI

YOUR GUESS?P

.

.  
APP--

LETTERS USED  
AIP  
YOUR GUESS?  
.

### Program Listing

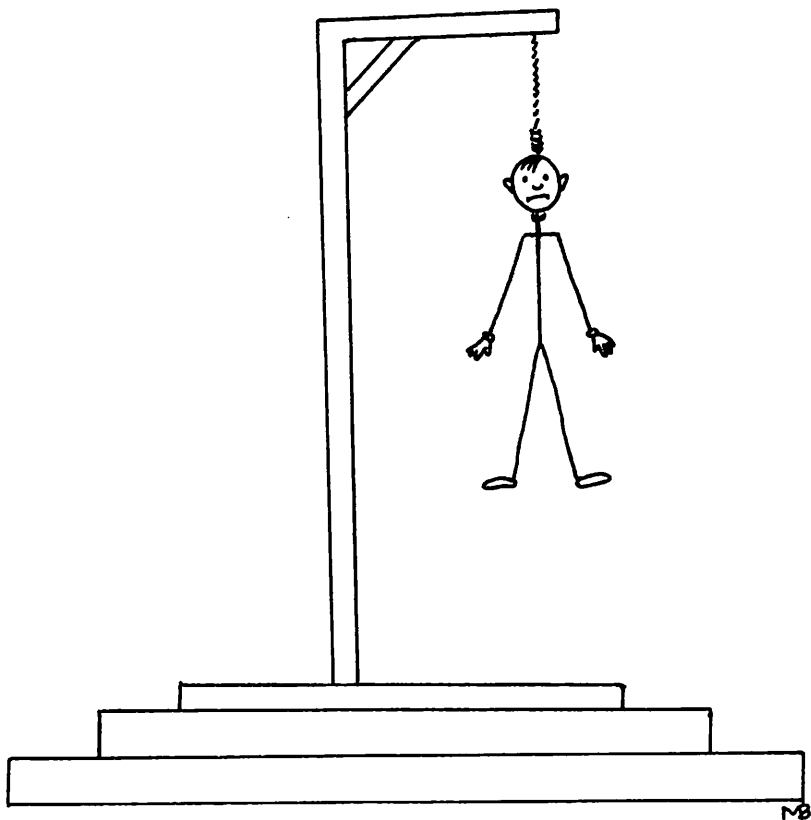
```
100 REM HANGMAN
110 RANDOMIZE
120 M1$="LETTERS USED"
130 M2$=""
140 M3$="YOUR GUESS?"
150 CALL CLEAR
160 DATA TABLE,CHAIR,HOUSE
170 DATA BOOK,UNIVERSITY
180 DATA AIRPLANE,SHIP,MONEY
190 DATA HORSE,APPLE
200 DATA 15,21,79,16,21,73
210 DATA 17,21,73,16,20,95
220 DATA 16,22,95,18,20,47
230 DATA 18,22,92,16,19,42
240 DATA 16,23,42,19,19,42
250 DATA 19,23,42
260 N=10
270 DIM W$(10)
280 FOR I=1 TO N
290 READ W$(I)
300 NEXT I
310 H=0
320 P=INT(RND*N)+1
330 B$=W$(P)
340 L=LEN(B$)
350 CALL HCHAR(5,10,45,L)
360 ROW=7
```

```

370 COL=3
380 M$=M1$
390 GOSUB 770
400 ROW=9
410 COL=3
420 M$=M2$
430 GOSUB 770
440 ROW=11
450 COL=3
460 M$=M3$
470 GOSUB 770
480 CALL KEY(0,KEY,STATUS)
490 IF STATUS=0 THEN 480
500 CALL HCHAR(11,14,KEY)
510 M2$=M2$&CHR$(KEY)
520 MAR=0
530 FOR J=1 TO LEN(B$)
540 K$=SEG$(B$,J,1)
550 K=ASC(K$)
560 IF KEY=K THEN 570 ELSE 590
570 MAR=1
580 CALL HCHAR(5,9+J,K)
590 NEXT J
600 IF MAR=0 THEN 610 ELSE 640
610 H=H+1
620 READ RR,CC,AA
630 CALL HCHAR(RR,CC,AA)
640 IF H=11 THEN 650 ELSE 680
650 PRINT "YOU ARE HUNG"
660 CALL SOUND(100,294,2)
670 STOP
680 BLANK=0
690 FOR I=1 TO L
700 CALL GCHAR(5,9+I,XX)
710 IF XX=45 THEN 720 ELSE 730
720 BLANK=1
730 NEXT I
740 IF BLANK=0 THEN 750 ELSE 400
750 PRINT "YOU GOT IT"

```

```
760 STOP
770 FOR I=1 TO LEN(M$)
780 LT$=SEG$(M$,I,1)
790 LT=ASC(LT$)
800 CALL HCHAR(ROW,COL+I-1,LT)
810 NEXT I
820 RETURN
830 END
```



# Knight's Tour

In the classical chessboard problem, the knight is allowed to make a sequence of moves to any square which is two rows and one column or two columns and one row away from its current position, each square on the board can be landed only once. In the beginning, you are asked to specify a row number between 1 and 8, and to specify a column number between 1 and 8. Then computer draws an 8 by 8 chessboard. Starting from the row and column specified, a sequence of moves is made randomly by the computer until no more move is possible. We use the set of characters, starting from letter A, to indicate the sequence of moves.

## Sample Run

>RUN

INITIAL ROW?3

INITIAL COL?5

(you will see a Knight's Tour on the screen)

## Program Listing

```
100 REM KNIGHT'S TOUR
110 CALL CLEAR
120 DIM RR(8),CC(8)
130 DATA 6,3,-6,3,6,-3
140 DATA -6,-3,3,6,-3,6
150 DATA 3,-6,-3,-6
160 FOR I=1 TO 8
```



```

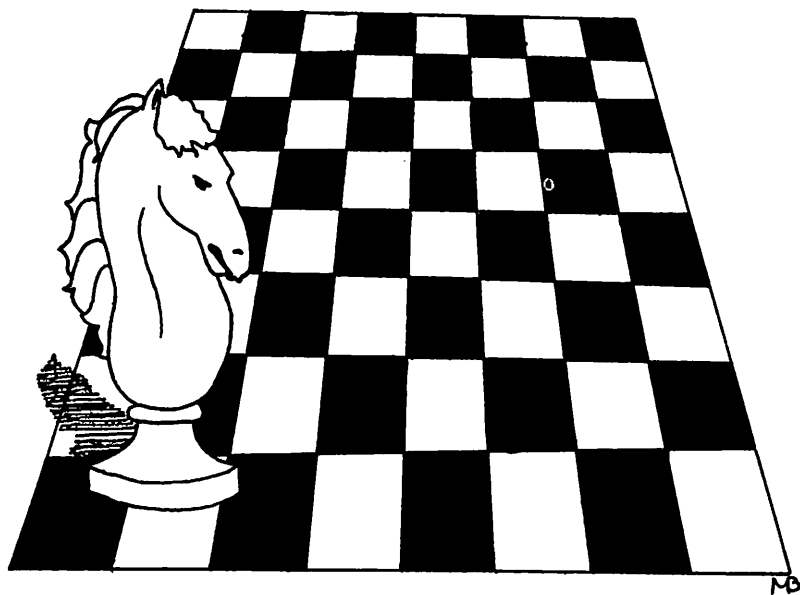
170 READ R,C
180 RR(I)=R
190 CC(I)=C
200 NEXT I
210 RANDOMIZE
220 BAR$="FFFFFFFFFFFFFFF"
230 CALL CHAR(128,BAR$)
240 CALL CHAR(136,BAR$)
250 CALL COLOR(13,7,1)
260 CALL COLOR(14,2,1)
270 INPUT "INITIAL ROW?":R
280 IF (R<1)+(R>8)THEN 270
290 INPUT "INITIAL COL?":C
300 IF (C<1)+(C>8)THEN 290
310 CALL CLEAR
320 C1=128
330 C2=136
340 FOR II=0 TO 0.5 STEP 0.5
350 FOR I=II TO 3+II
360 FOR J=1 TO 3
370 FOR K=5 TO 23 STEP 6
380 CALL HCHAR(I*6+J,K,C1,3)
390 CALL HCHAR(I*6+J,K+3,C2,3)
400 NEXT K
410 NEXT J
420 NEXT I
430 CT=C1
440 C1=C2
450 C2=CT
460 NEXT II
470 ROW=R*3-1
480 COL=3+C*3
490 SY=65
500 CALL HCHAR(ROW,COL,SY)
510 SY=SY+1
520 N=INT(RND*8)+1
530 FOR I=1 TO 8
540 IF N<=8 THEN 560
550 N=N-8

```

```

560 NROW=ROW+RR(N)
570 NCOL=COL+CC(N)
580 N=N+1
590 IF (NROW>0)*(NROW<25)*
      (NCOL>4)*(NCOL<29)THEN 600
      ELSE 650
600 CALL GCHAR(NROW,NCOL,X)
610 IF (X=128)+(X=136)THEN
      620 ELSE 650
620 ROW=NROW
630 COL=NCOL
640 GOTO 500
650 NEXT I
660 CALL SOUND(2000,-3,5)
670 CALL KEY(0,KEY,STATUS)
680 IF STATUS=0 THEN 670
690 STOP
700 END

```



# Lottery

This program simulates N.Y. state lottery. Each ticket has a 6-digit number. Amount of winning is as follows:

All 6 digits correct	\$50,000.
First 5 or last 5 digits correct	\$ 2,000.
First 4 or last 4 digits correct	\$ 125.
First 3 or last 3 digits correct	\$ 25.
First 2 and last 2 digits correct	\$ 5.
First 2 or last 2 digits correct	\$ 2.

You first specify the total number of tickets. Then enter each ticket number in 6 digits. The computer will tell you the amount of winning, if any.

## Sample Run

```
>RUN
NO. OF TICKETS?1
ENTER TICKET NO. IN 6 DIGITS
TICKET NO. 1 ? 257790
WINNING NO:799890
TICKET NO:257790
AMOUNT OF WINNING:$ 2
TOTAL WINNING:$ 2
```

## Program Listing

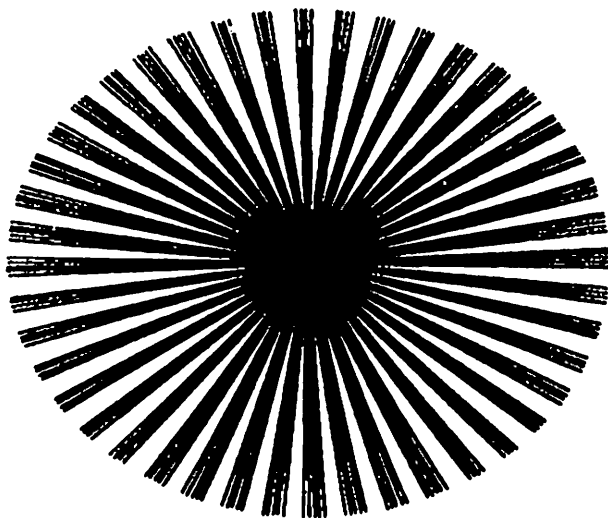
```
100 REM LOTTERY
110 CALL CLEAR
120 RANDOMIZE
```

```

130 DIM T$(100),P(4)
140 P(1)=2000
150 P(2)=125
160 P(3)=25
170 P(4)=2
180 DATA 1,5,2,5,1,4,3,4
190 DATA 1,3,4,3,1,2,5,2
200 W$=""
210 FOR I=1 TO 6
220 R=INT(RND*10)
230 W$=W$&STR$(R)
240 NEXT I
250 INPUT "NO. OF TICKETS?":N
260 PRINT
270 PRINT "ENTER TICKET NO. IN
    6 DIGITS"
280 PRINT
290 FOR I=1 TO N
300 PRINT "TICKET NO.";I;
310 INPUT T$(I)
320 PRINT
330 NEXT I
340 PRINT "WINNING NO: ";W$
350 PRINT
360 FOR I=1 TO N
370 K$=T$(I)
380 PAY=0
390 IF K$=W$ THEN 400
    ELSE 420
400 PAY=50000
410 GOTO 520
420 IF (SEG$(K$,1,2)=SEG$(W$,1,2))*
    (SEG$(K$,5,2)=SEG$(W$,5,2))
    THEN 430 ELSE 450
430 PAY=5
440 GOTO 520
450 RESTORE
460 FOR J=1 TO 8
470 READ A,B

```

```
480 IF SEG$(K$,A,B)=SEG$(W$,A,B)
    THEN 510
490 NEXT J
500 GOTO 520
510 PAY=P(INT((J+1)/2))
520 PRINT "TICKET NO:";K$
530 PRINT "AMOUNT OF WINNING:$";PAY
540 PRINT
550 T=T+PAY
560 NEXT I
570 PRINT
580 PRINT "TOTAL WINNING:$";T
590 END
```



# Moon Landing

Imagine you are landing a spacecraft on the surface of the moon. You start with 100 units of fuel. The initial rate of fuel consumption is 50%. You can increase the rate of fuel consumption by pressing key H, or reduce it by pressing L. If the fuel rate is too low, you may crash. If your fuel rate is too high, you may run out of fuel too early. This also leads to crash landing. So be careful and have a happy moon landing. Press any key to start a new game. To terminate the program, hold down key FCTN, then press key CLEAR

## Program Listing

```
100 REM MOON LANDING
110 CALL CLEAR
120 CALL SCREEN(6)
130 BAR$="FFFFFFFFFFFFFFF"
140 CALL CHAR(128,BAR$)
150 CALL COLOR(13,15,1)
160 CALL CHAR(136,BAR$)
170 CALL COLOR(14,9,1)
180 CALL CHAR(144,BAR$)
190 CALL COLOR(15,12,1)
200 DATA 2,4,70,2,5,85,2,6,69
210 DATA 2,7,76,4,4,82,4,5,65
220 DATA 4,6,84,4,7,69,3,11,48
230 DATA 3,20,53,3,28,49,3,21,48
240 DATA 6,18,79
250 DATA 3,29,48,3,30,48
260 DATA 67,82,65,83,72,32
270 DATA 76,65,78,68,73,78
280 DATA 71
```

```

290 DATA 83,79,70,84,32,76
300 DATA 65,78,68,73,78,71
310 RESTORE
320 CALL HCHAR(12,11,32,13)
330 F=100
340 CALL HCHAR(22,18,32,1)
350 T=0
360 R=50
370 XOLD=6
380 FOR I=1 TO 15
390 READ ROW,C,A
400 CALL HCHAR(ROW,C,A)
410 NEXT I
420 CALL HCHAR(2,11,136,20)
430 CALL HCHAR(4,11,144,10)
440 CALL HCHAR(23,1,128,64)
450 CALL KEY(0,KEY,STATUS)
460 IF STATUS=0 THEN 450
470 IF F>0 THEN 490
480 R=0
490 XX=50-R-T*(F=0)*30
500 XNEW=INT(XX/20)+XOLD
510 IF (XNEW<23) THEN 530
520 XNEW=23
530 IF XNEW>6 THEN 550
540 XNEW=6
550 CALL HCHAR(XOLD,18,32)
560 CALL HCHAR(XNEW,18,79)
570 IF XNEW=22 THEN 940
580 IF XNEW=23 THEN 860
590 XOLD=XNEW
600 CALL KEY(0,KEY,STATUS)
610 IF STATUS=0 THEN 710
620 IF KEY=72 THEN 680
630 IF KEY<>76 THEN 710
640 R=R-20
650 IF R>=0 THEN 710
660 R=0
670 GOTO 710

```

```

680 R=R+20
690 IF R<101 THEN 710
700 R=100
710 T=T+1
720 F=F-R/5
730 IF F>=0 THEN 750
740 F=0
750 COL=INT(F/5)+11
760 N=32-COL
770 CALL HCHAR(2,COL,32,N)
780 COL=INT(R/5)+11
790 N=32-COL
800 C1=COL-10
810 IF (C1>0)THEN 830
820 GOTO 840
830 CALL HCHAR(4,11,144,C1)
840 CALL HCHAR(4,COL,32,N)
850 GOTO 470
860 FOR I=1 TO 13
870 READ A
880 CALL HCHAR(12,I+10,A)
890 NEXT I
900 CALL SOUND(500,-1,2)
910 CALL KEY(0,KEY,STATUS)
920 IF STATUS=0 THEN 910
930 GOTO 310
940 RESTORE 290
950 FOR I=1 TO 12
960 READ A
970 CALL HCHAR(12,I+10,A)
980 NEXT I
990 CALL SOUND(200,200,10)
1000 CALL KEY(0,KEY,STATUS)
1010 IF STATUS=0 THEN 1000
1020 GOTO 310
1030 END

```



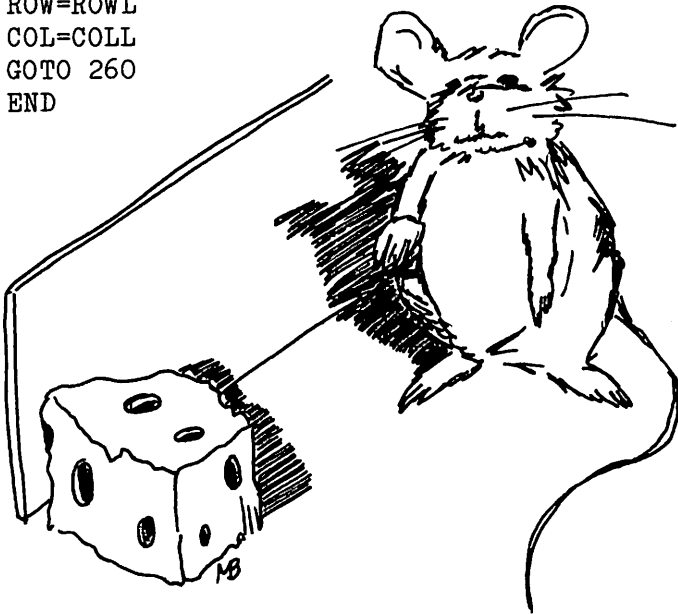
# Mouse and Cheese

This program draws a maze with a mouse and a piece of cheese. The mouse runs randomly in the maze to find the cheese. This is an interesting example of the "random walk" problem known by statistician.

## Program Listing

```
100 REM MOUSE AND CHEESE
110 CALL CLEAR
120 RANDOMIZE
130 BAR$="FFFFFFFFFFFFFFF"
140 CALL CHAR(128,BAR$)
150 CALL COLOR(13,7,1)
160 CALL COLOR(5,2,7)
170 CALL COLOR(6,2,7)
180 FOR I=4 TO 20 STEP 4
190 CALL HCHAR(I,9,128,17)
200 CALL VCHAR(4,I+5,128,17)
210 NEXT I
220 CALL HCHAR(8,25,67)
230 ROW=16
240 COL=9
250 CALL HCHAR(ROW,COL,77)
260 N=INT(RND*4)+1
270 ROWL=ROW
280 COLL=COL
290 SR=1
300 SC=1
310 ON N GOTO 320,350,370,390
320 ROWL=ROW-4
330 SR=-1
340 GOTO 410
350 COLL=COL+4
```

```
360 GOTO 410
370 ROWL=ROW+4
380 GOTO 410
390 COLL=COL-4
400 SC=-1
410 IF (COLL>8)*(COLL<26)*(ROWL>3)*
    (ROWL<21)THEN 420 ELSE 260
420 CALL GCHAR(ROWL,COLL,X)
430 PR=ROW
440 PC=COL
450 FOR R=ROW TO ROWL STEP SR
460 FOR C=COL TO COLL STEP SC
470 CALL HCHAR(PR,PC,128)
480 CALL HCHAR(R,C,77)
490 PR=R
500 PC=C
510 NEXT C
520 NEXT R
530 IF X=67 THEN 540 ELSE 560
540 PRINT "MOUSE FOUND CHEESE"
550 STOP
560 ROW=ROWL
570 COL=COLL
580 GOTO 260
590 END
```



# Nicoma

You first think of a whole number between 1 and 100. Then the computer asks you three questions. Based on your answer, the computer can figure out what your number is.

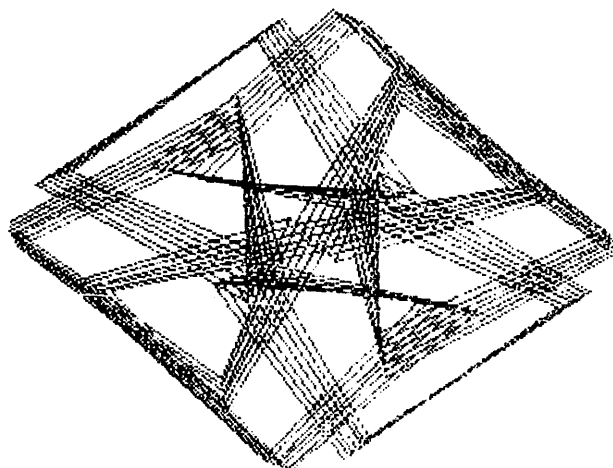
## Sample Run

```
>RUN
THINK OF A WHOLE NUMBER BETWEEN
1 AND 100
YOUR NUMBER DIVIDED BY 3 HAS A
REMINDER OF 1
YOUR NUMBER DIVIDED BY 5 HAS A
REMINDER OF 1
YOUR NUMBER DIVIDED BY 7 HAS A
REMINDER OF 2
YOUR NUMBER IS 16
```

## Program Listing

```
100 REM NICOMA
110 CALL CLEAR
120 PRINT "THINK OF A WHOLE NUMBER
    BETWEEN 1 AND 100",,,
130 INPUT "YOUR NUMBER DIVIDED
    BY 3 HAS A REMINDER OF ":P
140 INPUT "YOUR NUMBER DIVIDED
    BY 5 HAS A REMINDER OF ":Q
150 INPUT "YOUR NUMBER DIVIDED
    BY 7 HAS A REMINDER OF ":R
160 S=70*P+21*Q+15*R
```

```
170 IF S>105 THEN 180 ELSE 200
180 S=S-105
190 GOTO 170
200 PRINT "YOUR NUMBER IS ";S,,,
210 GOTO 120
220 END
```



# Nimb

This program plays the game of Nimb. In the beginning, you are asked to specify the total number of objects and the maximum number of objects that can be taken each time. You are allowed to make the first move. Then it alternates between you and the computer. The player forced to take the last one loses.

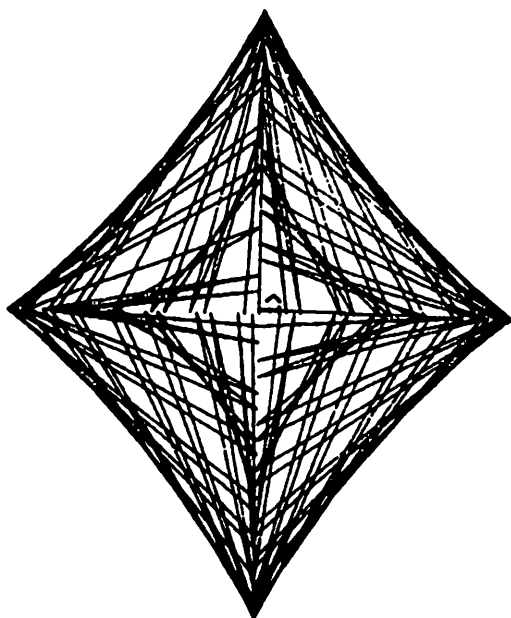
## Sample Run

```
>RUN
TOTAL NUMBER OF OBJECTS?6
MAXIMUM NUMBER TAKEN EACH TIME?3
YOUR MOVE?1
COMPUTER MOVE 1
NUMBER LEFT 4
YOUR MOVE?3
YOU WIN
** DONE **
```

## Program Listing

```
100 REM NIMB
110 CALL CLEAR
120 INPUT "TOTAL NUMBER OF OBJECTS?":N
130 INPUT "MAXIMUM NUMBER TAKEN EACH
    TIME?":M
140 IF (M<0)+(N<M) THEN 130
150 INPUT "YOUR MOVE?":Y
160 IF (Y<1)+(Y>M)+(Y>N) THEN 150
170 N=N-Y
180 IF (N>0)THEN 210
```

```
190 PRINT "COMPUTER WINS"  
200 STOP  
210 P=INT((N-1)/(M+1))  
220 R=N-1-P*(M+1)  
230 IF (R=0)THEN 240 ELSE 280  
240 IF (N>1)THEN 250 ELSE 270  
250 R=1  
260 GOTO 280  
270 R=1+INT(M*RND)  
280 N=N-R  
290 IF (N<=0)THEN 300 ELSE 320  
300 PRINT "YOU WIN"  
310 STOP  
320 PRINT "COMPUTER MOVE";R  
330 PRINT "NUMBER LEFT";N,,  
340 GOTO 150  
350 END
```



# On the Money

A pile of money is on the ground. You try to shoot at it from the left corner of the screen. You enter the velocity and angle, the computer will do the rest. If you hit on the money, you will hear a loud sound.

To terminate this program, hold down FCTN key and press CLEAR key.

## Sample Run

```
>RUN
VELOCITY=?3.5
ANGLE IN DEGREE=?40
(hold your breath and see what happens)
```

## Program Listing

```
100 REM ON THE MONEY
110 RANDOMIZE
120 CALL CLEAR
130 P=RND*25+3
140 P=INT(P)
150 CALL HCHAR(22,P,36,1)
160 CALL HCHAR(23,P-1,36,3)
170 R=5
180 C=5
190 M$="VELOCITY=?"
200 GOSUB 570
210 C=16
220 GOSUB 470
230 V=M
240 R=10
```

```

250 C=5
260 M$="ANGLE IN DEGREE=?"
270 GOSUB 570
280 C=24
290 GOSUB 470
300 Z=M
310 Z=Z/57.29
320 T=0
330 ROW=INT(24-SIN(Z)*T*V+0.1*T*T)
340 COL=INT(V*COS(Z)*T)+1
350 IF ROW<>23 THEN 390
360 IF COL<P-1 THEN 390
370 IF COL>P+1 THEN 390
380 CALL SOUND(2000,-3,5)
390 IF ROW<=0 THEN 110
400 IF ROW>24 THEN 110
410 IF COL<=0 THEN 110
420 IF COL>32 THEN 110
430 CALL HCHAR(ROW,COL,42,1)
440 T=T+1
450 GOTO 330
460 GOTO 210
470 M$=""
480 CALL KEY(O,KEY,STATUS)
490 IF STATUS=0 THEN 480
500 IF KEY=13 THEN 550
510 CALL HCHAR(R,C,KEY,1)
520 C=C+1
530 M$=M$&CHR$(KEY)
540 GOTO 480
550 M=VAL(M$)
560 RETURN
570 FOR I=1 TO LEN(M$)
580 C$=SEG$(M$,I,1)
590 N=ASC(C$)
600 CALL HCHAR(R,C+I,N,1)
610 NEXT I
620 RETURN
630 END

```



# Piglatin

This program translates English text into Piglatin. The rule of translation is the following: Move all consonants before the first vowel in an English word to the end of the word, and add the suffix "ay." This program is fun.

## Sample Run

```
>RUN
TYPE IN ENGLISH TEXT
? THIS BOOK IS GREAT
ENGLISH TEXT

THIS BOOK IS GREAT

PIGLATIN

ISTHAY OOKBAY ISAY EATGRAY
```

## Program Listing

```
100 REM PIGLATIN
110 CALL CLEAR
120 DIM B(5)
130 DATA 65,69,73,79,85
140 FOR I=1 TO 5
150 READ B(I)
160 NEXT I
170 PRINT "TYPE IN ENGLISH TEXT"
180 INPUT M$
190 J1=1
200 IF J1>LEN(M$)THEN 510
210 FOR I=J1 TO LEN(M$)
```

```

220 C$=SEG$(M$,I,1)
230 C=ASC(C$)
240 IF C<>32 THEN 270
250 NEXT I
260 GOTO 510
270 J1=I
280 FOR J3=J1 TO LEN(M$)
290 C$=SEG$(M$,J3,1)
300 C=ASC(C$)
310 IF C=32 THEN 330
320 NEXT J3
330 J3=J3-1
340 FOR J2=J1 TO J3
350 C$=SEG$(M$,J2,1)
360 C=ASC(C$)
370 FOR I=1 TO 5
380 IF C=B(I) THEN 420
390 NEXT I
400 NEXT J2
410 J2=J2-1
420 IF (J1=J2)+(J2=J3) THEN
430 ELSE 460
430 N$=N$&SEG$(M$,J1,J3-J1+1)
440 N$=N$&"AY "
450 GOTO 490
460 N$=N$&SEG$(M$,J2,J3-J2+1)
470 N$=N$&SEG$(M$,J1,J2-J1)
480 N$=N$&"AY "
490 J1=J3+1
500 GOTO 200
510 PRINT "ENGLISH TEXT"
520 PRINT
530 PRINT M$
540 PRINT
550 PRINT "PIGLATIN"
560 PRINT
570 PRINT N$
580 STOP
590 END

```

# Skiing

Would you enjoy a skiing trip to Aspen, Colorado? Well, let us first try one on your computer. This program will randomly draw 12 trees on the screen. While you ski downward, you may move right by pressing key D and move left by pressing key S. Just be careful and not hit any tree.

## Program Listing

```
100 REM SKIING
110 CALL CLEAR
120 CALL COLOR(1,13,1)
130 CALL SCREEN(16)
140 DATA -1,0,94,0,-1,47
150 DATA 0,0,94,0,1,92
160 DATA 1,-1,47,1,0,94
170 A$="1898FF3D3D3DE505"
180 CALL CHAR(111,A$)
190 B$="FF00000000000000"
200 CALL CHAR(95,B$)
210 DATA 1,1,92
220 RANDOMIZE
230 FOR I=1 TO 12
240 ROW=INT((10*RND)/4)*4+10
250 COL=INT((26*RND)/4)*4+3+(ROW=14)
260 RESTORE
270 FOR J=1 TO 7
280 READ R,C,A
290 CALL HCHAR(ROW+R,COL+C,A)
300 NEXT J
310 NEXT I
320 FOR PR=1 TO 24
330 PR1=PR+1
340 FOR PC=COL TO 26
350 CALL GCHAR(PR,PC,X)
```

```

360 CALL GCHAR(PR1,PC,Y)
370 IF (X=32)+(Y=32)THEN 410
380 NEXT PC
390 NEXT PR
400 GOTO 220
410 CALL HCHAR(PR,PC,111)
420 CALL HCHAR(PR1,PC,95)
430 IF PR1=24 THEN 730
440 PRN=PR+1
450 PR1N=PRN+1
460 PCN=PC
470 CALL KEY(O,KEY,STATUS)
480 IF STATUS=0 THEN 530
490 IF KEY<>83 THEN 510
500 PCN=PC-1
510 IF KEY<>68 THEN 530
520 PCN=PC+1
530 IF PCN>4 THEN 550
540 PCN=4
550 IF PCN<26 THEN 570
560 PCN=26
570 IF PR1N<24 THEN 600
580 PR1N=24
590 PRN=23
600 CALL GCHAR(PRN,PCN,X)
610 CALL GCHAR(PR1N,PCN,Y)
620 CALL HCHAR(PR,PC,32)
630 CALL HCHAR(PR1,PC,32)
640 PR=PRN
650 PC=PCN
660 PR1=PR1N
670 IF (X=32)+(Y=32)THEN 410
680 CALL HCHAR(PR,PC,42)
690 CALL HCHAR(PR1,PC,42)
700 CALL SOUND(500,-1,2)
710 PRINT "LOST"
720 STOP
730 PRINT "WIN"
740 END

```

# Timer

This program is a count down timer. User enters hours, minutes and seconds. Computer will start to count time and display the remaining time on the screen. At the end, it will also play a few musical tones. This timer is not very accurate. Do not use it for serious business.

## Sample Run

```
>RUN
HOURS?0
MINUTES?1
SECONDS?15
      (you will see remaining time on screen)
```

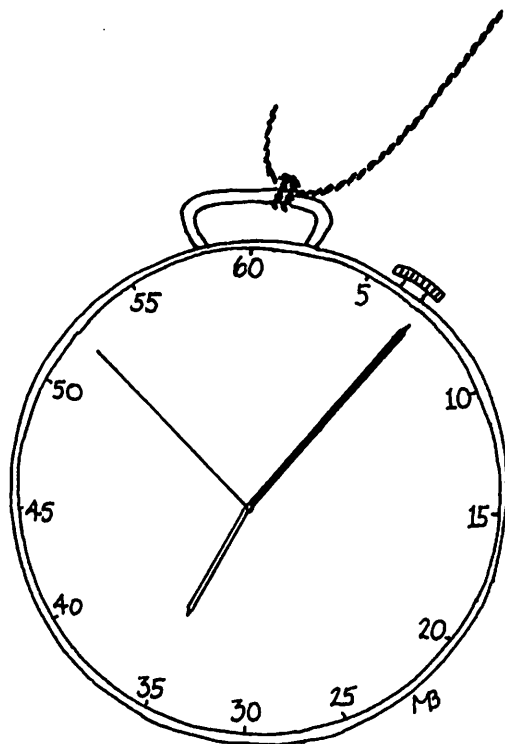
## Program Listing

```
100 REM TIMER
110 INPUT "HOURS?":H
120 INPUT "MINUTES?":MIN
130 INPUT "SECONDS?":SEC
140 COUNT=3600*H+60*MIN+SEC
150 CALL CLEAR
160 COUNT=COUNT-1
170 IF COUNT=-1 THEN 320
180 H=INT(COUNT/3600)
190 C=COUNT-H*3600
200 H$=STR$(H)
210 MIN=INT(C/60)
220 SEC=C-60*MIN
230 MIN$=STR$(MIN)
```

```

240 SEC$=STR$(SEC)
250 TIME$=H$&":"&MIN$&":"&SEC$&" "
260 FOR I=1 TO LEN(TIME$)
270 L$=SEG$(TIME$,I,1)
280 N=ASC(L$)
290 CALL HCHAR(7,10+I,N)
300 NEXT I
310 GOTO 160
320 TONE=110
330 FOR CT=1 TO 10
340 CALL SOUND(1000,TONE,10)
350 TONE=TONE+110
360 NEXT CT
370 END

```



# Tower of Hanoi

This program solves the well known puzzle, called the Tower of Hanoi. There are three posts. On one post, there are a number of rings of various diameters, with the largest on the bottom and the smallest on the top. The puzzle is to transfer the set of rings to one of the other two posts, moving one ring at a time and never placing a ring on top of a smaller one.

In the following sample run, we start with 3 rings on post 1. The solution is to move ring 1 (the smallest one) from post 1 to post 2, then to move ring 2 from post 1 to post 3, etc.

## Sample Run

```
>RUN
NO. OF RINGS?3
FROM POST NUMBER?1
TO POST NUMBER?2
MOVE RING  FROM POST TO POST
      1      1      2
      2      1      3
      1      2      3
      3      1      2
      1      3      1
      2      3      2
      1      1      2
** DONE **
```

## Program Listing

```
100 REM TOWER OF HANOI
```

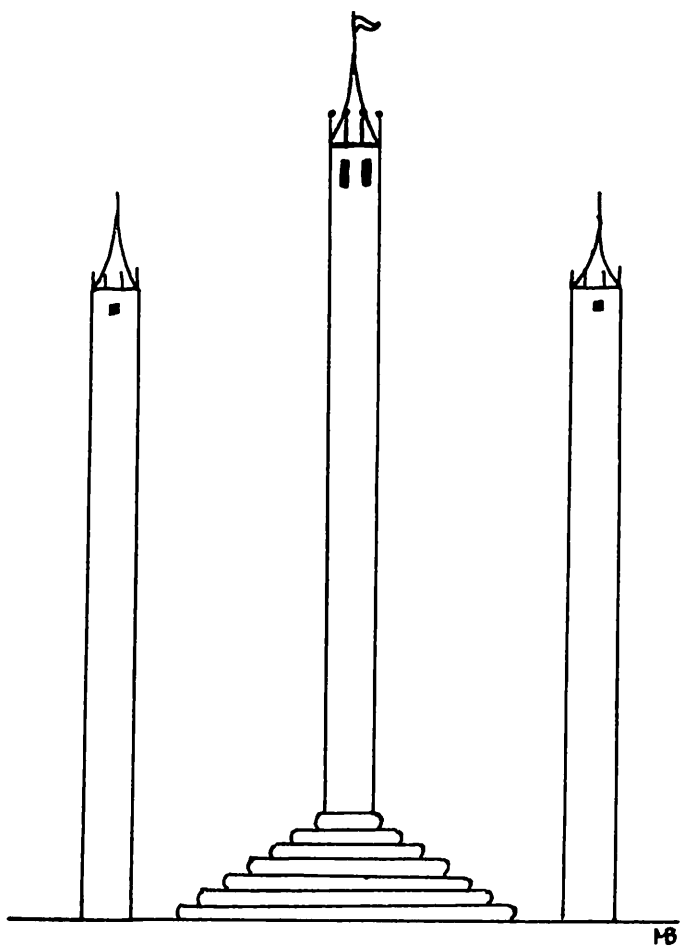
```

110 CALL CLEAR
120 DIM SN(10),SP1(10)
130 DIM SP2(10),SR(10)
140 INPUT "NO. OF RINGS?":NR
150 PRINT
160 INPUT "FROM POST NUMBER?":F
170 PRINT
180 INPUT "TO POST NUMBER?":T
190 PRINT
200 PRINT "MOVE RING FROM POST TO POST"
210 SP=0
220 N=NR
230 P1=F
240 P2=T
250 SP=SP+1
260 SN(SP)=N
270 SP1(SP)=P1
280 SP2(SP)=P2
290 SR(SP)=3
300 IF N<=0 THEN 390 ELSE 310
310 SP=SP+1
320 SN(SP)=N
330 SP1(SP)=P1
340 SP2(SP)=P2
350 SR(SP)=1
360 N=N-1
370 P2=6-P1-P2
380 GOTO 300
390 N=SN(SP)
400 P1=SP1(SP)
410 P2=SP2(SP)
420 R=SR(SP)
430 SP=SP-1
440 ON R GOTO 450,390,540
450 PRINT TAB(5);N;TAB(16);P1
    ;TAB(25);P2
460 SP=SP+1
470 SN(SP)=N
480 SP1(SP)=P1

```



```
490 SP2(SP)=P2
500 SR(SP)=2
510 N=N-1
520 P1=6-P1-P2
530 GOTO 300
540 END
```



# Typing Monkey

In 1927, Eddington made the statement: "If an army of monkeys were strumming on typewriters they might write all the books in the British Museum." This program does just that. Try it.

## Program Listing

```
100 REM TYPING MONKEY
110 CALL CLEAR
120 DATA .058, .0696, .0862
130 DATA .1174, .2104, .2283
140 DATA .2418, .2922, .3415
150 DATA .3424, .3497, .3848
160 DATA .41, .4595, .5327
170 DATA .5449, .5457, .5909
180 DATA .6436, .7162, .7450
190 DATA .7538, .7741, .7747
200 DATA .7969, .7973, .9942
210 DATA 1.0
220 DIM A(28)
230 RANDOMIZE
240 FOR I=1 TO 28
250 READ A(I)
260 NEXT I
270 FOR R=3 TO 21 STEP 3
280 FOR C=5 TO 28
290 X=RND
300 FOR I=1 TO 28
310 IF (X<=A(I))THEN 330
320 NEXT I
330 IF I=27 THEN 340 ELSE 360
340 B=32
350 GOTO 400
```



# Christmas Tree

This program draws a beautiful Christmas tree on your TV screen. To terminate this program, hold down FCTN key, then press CLEAR.

## Program Listing

```
100 REM CHRISTMAS TREE
110 RANDOMIZE
120 DIM A(10)
130 DATA 64,65,66,42,43
140 DATA 64,79,94,74,75
150 CALL CLEAR
160 CALL SCREEN(11)
170 CALL COLOR(2,16,13)
180 CALL COLOR(5,12,6)
190 CALL COLOR(6,9,13)
200 CALL COLOR(8,12,13)
210 CALL VCHAR(20,17,94,3)
220 CALL VCHAR(20,16,94,3)
230 CALL HCHAR(23,3,35,61)
240 FOR I=1 TO 10
250 READ A(I)
260 NEXT I
270 FOR I=1 TO 10
280 R1=INT(16*RND)+4
290 N=INT(R1/2-2)
300 C1=INT((R1-3)*RND)-N+16
310 CALL HCHAR(R1,C1,A(I))
320 NEXT I
330 GOTO 270
340 END
```

# Colorful

This program will continuously draw colorful patterns on your TV screen. To terminate this program, hold the key FCTN down, then press the key CLEAR.

## Program Listing

```
100 REM COLORFUL
110 CALL CLEAR
120 FOR I=1 TO 16
130 K=INT(16*RND)+1
140 J=INT(16*RND)+1
150 CALL COLOR(I,J,K)
160 NEXT I
170 RANDOMIZE
180 FOR R=1 TO 24
190 FOR C=1 TO 32
200 A=INT(80*RND)+32
210 CALL HCHAR(R,C,A)
220 NEXT C
230 NEXT R
240 GOTO 120
250 END
```

# Flag

This program will draw a beautiful flag of the United States of America on your TV screen.

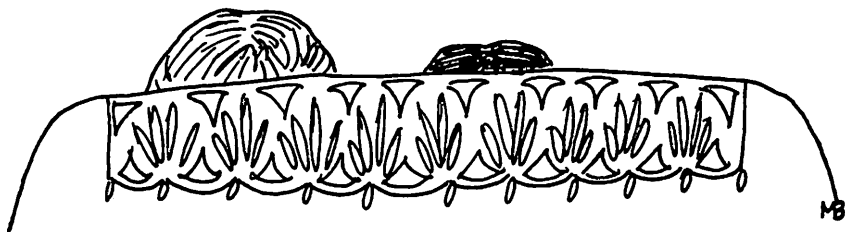
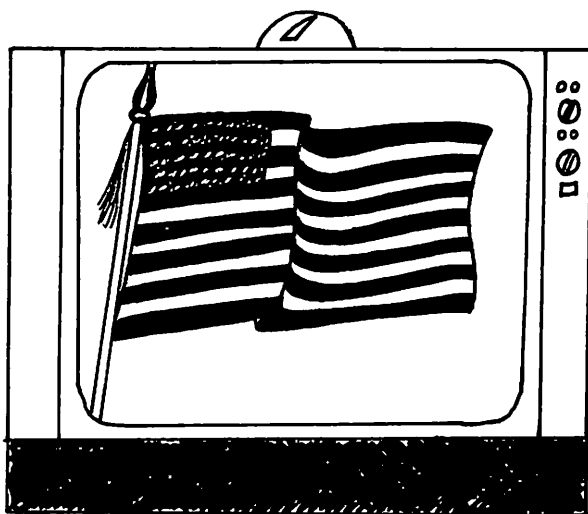
## Program Listing

```
100 REM FLAG
110 STAR$="FFFFFF7E3C1E3EBFF"
120 HSTP$="00000000FFFFFFFF"
130 STP$="FFFFFFFFFFFFFFFF"
140 CALL CLEAR
150 CALL CHAR(35,STAR$)
160 CALL CHAR(33,HSTP$)
170 CALL CHAR(34,STP$)
180 CALL COLOR(1,5,16)
190 CALL CHAR(40,STP$)
200 CALL CHAR(41,HSTP$)
210 CALL COLOR(2,7,16)
220 CALL SCREEN(16)
230 FOR I=0 TO 3
240 CALL HCHAR(1+I*3,16,41,15)
250 CALL HCHAR(2+I*3,16,40,15)
260 NEXT I
270 FOR I=4 TO 6
280 CALL HCHAR(1+I*3,3,41,28)
290 CALL HCHAR(2+I*3,3,40,28)
300 NEXT I
310 CALL HCHAR(1,3,33,13)
320 FOR I=2 TO 11
330 CALL HCHAR(I,3,34,13)
340 NEXT I
350 FOR I=2 TO 10 STEP 2
360 FOR J=4 TO 14 STEP 2
370 CALL HCHAR(I,J,35,1)
```

```

380 NEXT J
390 NEXT I
400 FOR I=3 TO 9 STEP 2
410 FOR J=5 TO 13 STEP 2
420 CALL HCHAR(I,J,35,1)
430 NEXT J
440 NEXT I
450 CALL KEY(0,KEY,STATUS)
460 IF STATUS=0 THEN 450
470 CALL CLEAR
480 END

```



# 3D Graph

This program will plot a family of curves on the screen. It looks 3-dimensional.

## Program Listing

```
100 REM 3D GRAPH
110 CALL CLEAR
120 FOR I=-14 TO 14
130 A=0
140 FOR J=-25 TO 25 STEP 2.5
150 R=I*I+J*J
160 K=INT(25*EXP(-R/100))
170 IF K<=A THEN 220
180 A=K
190 R=(31-K)*0.8
200 C=I+16
210 CALL HCHAR(R,C,46)
220 NEXT J
230 NEXT I
240 CALL KEY(0,KEY,STATUS)
250 IF STATUS=0 THEN 240
260 END
```



# Base Conversion

Numbers can be represented in different bases. This program will convert a positive integer in base A,  $x_A$ , to its equivalent representation in base B,  $x_B$ . The bases A and B can be any integer from 2 to 16. For example, consider number in base 8,  $107_8$ . From the following sample run, we can see that the same number in base 16 is  $47_{16}$ .

## Sample Run

```
>RUN
ENTER BASE=8
ENTER NUMBER IN BASE 8=107
BASE 2 =1000111
BASE 3 =2122
.
.
.

BASE 15 =4B
BASE 16 =47
```

## Program Listing

```
100 REM BASE CONVERSION
110 DIM M$(16),R(20)
120 DATA 0123456789ABCDEF
130 READ P$
140 FOR I=1 TO 16
150 M$(I)=SEG$(P$,I,1)
160 NEXT I
```

```

170 CALL CLEAR
180 INPUT "ENTER BASE=":B
190 IF (B<2)+(B>16)THEN 180
200 PRINT "ENTER NUMBER IN BASE";B;
210 INPUT "=":N$
220 N=0
230 FOR I=1 TO LEN(N$)
240 P$=SEG$(N$,I,1)
250 Q=-1
260 FOR J=1 TO B STEP 1
270 IF P$=M$(J)THEN 280 ELSE 290
280 Q=J-1
290 NEXT J
300 IF Q<0 THEN 200
310 N=N*B+Q
320 NEXT I
330 FOR B=2 TO 16
340 I=1
350 T=N
360 R(I)=T-INT(T/B)*B
370 T=INT(T/B)
380 I=I+1
390 IF T<>0 THEN 360
400 PRINT "BASE";B;"=";
410 FOR J=I-1 TO 1 STEP -1
420 K=R(J)+1
430 PRINT M$(K);
440 NEXT J
450 PRINT
460 NEXT B
470 END

```

# Differential Equations

This program solves first-order differential equations by the fourth-order Runge-Kutta method. The equation is of the form

$$dy/dx=f(x,y)$$

The function  $f(x,y)$  is defined in statement 120. The user is required to specify the initial values for  $x$ ,  $y$ , and the stepsize of  $x$ .

## Sample Run

```
>RUN
INITIAL X=0
INITIAL Y=0
STEP SIZE=0.1
X          Y
0          0
.1         .0048375
.2         .0187309014
.
.
.
.
1.9        1.049568877

(press any key to continue)
```

## Program Listing

```
100 REM DIFFERENTIAL EQUATIONS
```

```

110 CALL CLEAR
120 DEF F=-Y+X
130 INPUT "INITIAL X=":XI
140 INPUT "INITIAL Y=":YI
150 INPUT "STEP SIZE=":H
160 PRINT " X";TAB(16);"Y",,,
170 FOR I=1 TO 20
180 PRINT XI,YI
190 X=XI
200 Y=YI
210 C1=H*F
220 X=XI+H/2
230 Y=YI+C1/2
240 C2=H*F
250 X=XI+H/2
260 Y=YI+C2/2
270 C3=H*F
280 X=XI+H
290 Y=YI+C3
300 C4=H*F
310 XI=XI+H
320 YI=YI+(1/6)*(C1+2*C2+2*C3+C4)
330 NEXT I
340 CALL KEY(O,KEY,STATUS)
350 IF STATUS=0 THEN 340 ELSE 170
360 END

```

# Factorial

For any positive whole number N, the factorial of N is defined to be  $1*2*3*...*N$ . This program will calculate the factorial of any number up to 34. The result from this program can be 40 digits long.

## Sample Run

```
>RUN
N=20
FACTORIAL OF 1
1
FACTORIAL OF 2
2
.
.
.
FACTORIAL OF 20
2432902008176640000
```

## Program Listing

```
100 REM FACTORIAL
110 CALL CLEAR
120 DIM FAC(40)
130 LMT=40
140 INPUT "N=":N
150 N=INT(N)
160 IF (N<0)+(N>34)THEN 140
170 FOR I=1 TO LMT
180 FAC(I)=0
190 NEXT I
```

```

200 FAC(LMT)=1
210 FOR K=1 TO N
220 C=0
230 FOR L=LMT TO 1 STEP -1
240 FAC(L)=FAC(L)*K+C
250 C=INT(FAC(L)/10)
260 FAC(L)=FAC(L)-10*C
270 NEXT L
280 FOR I=1 TO LMT
290 IF FAC(I)<>0 THEN 310
300 NEXT I
310 G$=""
320 FOR L=I TO LMT
330 G$=G$&STR$(FAC(L))
340 NEXT L
350 PRINT "FACTORIAL OF";K
360 PRINT G$
370 PRINT
380 NEXT K
390 END

```

# Integration-Gaussian

Based on the six point Gaussian Integration method, this program computes an approximation of the integral for  $f(x)$  from any given lower limit of  $x$  to infinity. The function  $f(x)$  is defined in statement 120.

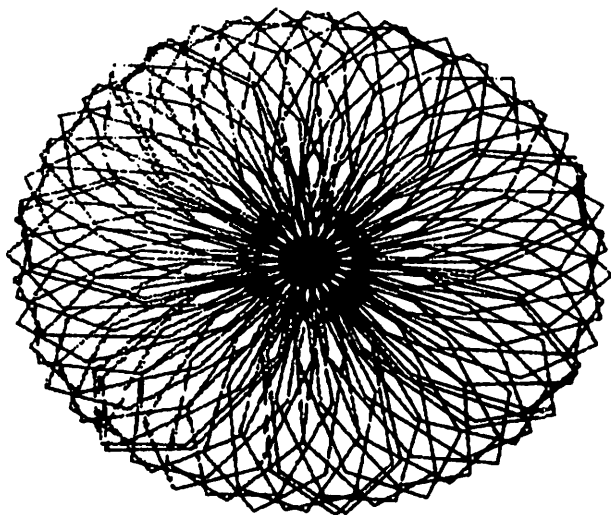
## Sample Run

```
>RUN
  LOWER LIMIT OF X=2.0
  INTEGRAL= .4999999938
```

## Program Listing

```
100 REM INTEGRATION-GAUSSIAN
110 CALL CLEAR
120 DEF F(X)=1/X^2
130 DATA .2386191861,.4679139346
140 DATA .6612093865,.360761573
150 DATA .9324695142,.1713244924
160 DIM X(6),Y(6),W(6)
170 INPUT "LOWER LIMIT OF X=":XZ
180 FOR I=1 TO 5 STEP 2
190 READ Y(I),W(I)
200 Y(I+1)=-Y(I)
210 W(I+1)=W(I)
220 NEXT I
230 FOR I=1 TO 6
240 X(I)=2/(1+Y(I))+XZ-1
250 NEXT I
260 S=0
```

```
270 FOR I=1 TO 6
280 S=S+W(I)/((1+Y(I))^2)*F(X(I))
290 NEXT I
300 S=S*2
310 PRINT "INTEGRAL=";S
320 END
```





# Integration-Simpson

This program uses Simpson's Rule to approximate the area under the curve  $f(x)$  over the interval  $(A,B)$ . The function  $f(x)$  is specified in statement 130. The user specifies the end points of the interval  $(A,B)$  and the total number of intervals used in Simpson's approximation.

## Sample Run

```
>RUN
A=0
B=2
NUMBER OF INTERVALS=10
RESULT=
 12.66666667
** DONE **
```

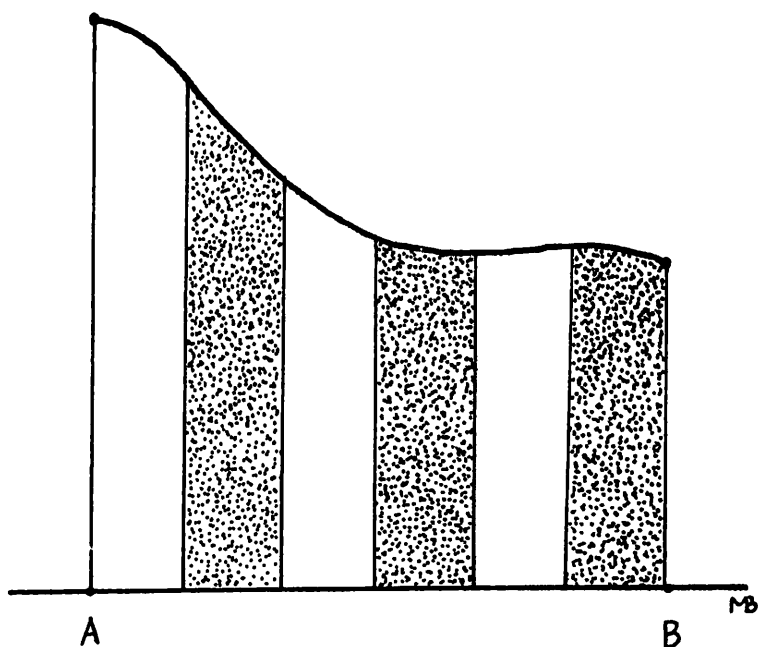
## Program Listing

```
100 REM INTEGRATION
110 REM SIMPSON'S RULE
120 CALL CLEAR
130 DEF F(X)=X^2+3*X+2
140 INPUT "A=":A
150 INPUT "B=":B
160 IF B<A THEN 140
170 INPUT "NUMBER OF INTERVALS=":K
180 IF K<0 THEN 170
190 L=(B-A)/K
200 LH=L/2
210 S=0.0
```

```

220 S1=F(A+LH)
230 FOR I=1 TO K-1
240 X=A+I*L
250 S=S+F(X)
260 S1=S1+F(X+LH)
270 NEXT I
280 RT=L*(F(A)+2*S+F(B)+4*S1)/6
290 PRINT "RESULT=":RT
300 END

```



# Linear Equations

This program solves a set of linear equations in the following form:

$$\begin{bmatrix} A(1,1) & A(1,2) & \dots & A(1,N) \\ A(2,1) & A(2,2) & \dots & A(2,N) \\ \vdots & \vdots & \ddots & \vdots \\ A(N,1) & A(N,2) & \dots & A(N,N) \end{bmatrix} \begin{bmatrix} X(1) \\ X(2) \\ \vdots \\ X(N) \end{bmatrix} = \begin{bmatrix} A(1,N+1) \\ A(2,N+1) \\ \vdots \\ A(N,N+1) \end{bmatrix}$$

where the number of equations N and all the elements A(I,J) are provided by the user.

## Sample Run

```
>RUN
NUMBER OF EQUATIONS?2
A(1,1)=? 1
A(1,2)=? 3
A(1,3)=? 2
A(2,1)=? 4
A(2,2)=? 2
A(2,3)=? 4
A MATRIX
  1 3 2
  4 2 4
X(1)= .8
X(2)= .4
** DONE **
```

## Program Listing

```

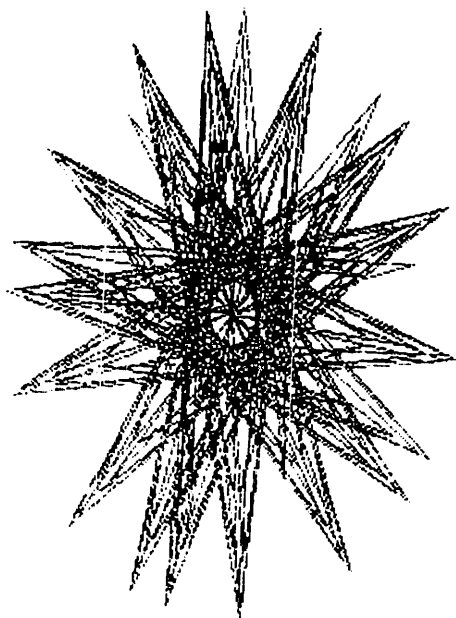
100 REM LINEAR EQUATIONS
110 DIM A(30,31)
120 INPUT "NUMBER OF EQUATIONS?":N
130 FOR I=1 TO N
140 FOR J=1 TO N+1
150 PRINT "A(";I;",";J;")=";
160 INPUT A(I,J)
170 NEXT J
180 NEXT I
190 PRINT "A MATRIX"
200 FOR I=1 TO N
210 FOR J=1 TO N+1
220 PRINT A(I,J);
230 NEXT J
240 PRINT
250 NEXT I
260 FOR I=1 TO N
270 R=I
280 B=ABS(A(I,I))
290 FOR J=I+1 TO N
300 IF ABS(A(J,I))>B THEN
310 ELSE 330
310 R=J
320 B=ABS(A(J,I))
330 NEXT J
340 IF B=0 THEN 350 ELSE 370
350 PRINT "NO WELL DEFINED SOLUTION"
360 STOP
370 IF R<>I THEN 380 ELSE 430
380 FOR J=1 TO N+1
390 B=A(I,J)
400 A(I,J)=A(R,J)
410 A(R,J)=B
420 NEXT J
430 B=A(I,I)
440 FOR J=I+1 TO N+1
450 A(I,J)=A(I,J)/B
460 NEXT J

```

```

470 FOR J=I+1 TO N
480 E=A(J,I)
490 FOR R=I+1 TO N+1
500 A(J,R)=A(J,R)-A(I,R)*E
510 NEXT R
520 NEXT J
530 NEXT I
540 FOR J=N-1 TO 1 STEP -1
550 E=0.0
560 FOR R=J+1 TO N
570 E=E+A(R,N+1)*A(J,R)
580 NEXT R
590 A(J,N+1)=A(J,N+1)-E
600 NEXT J
610 FOR I=1 TO N
620 PRINT "X(";I;")=";
      A(I,N+1)
630 NEXT I
640 END

```



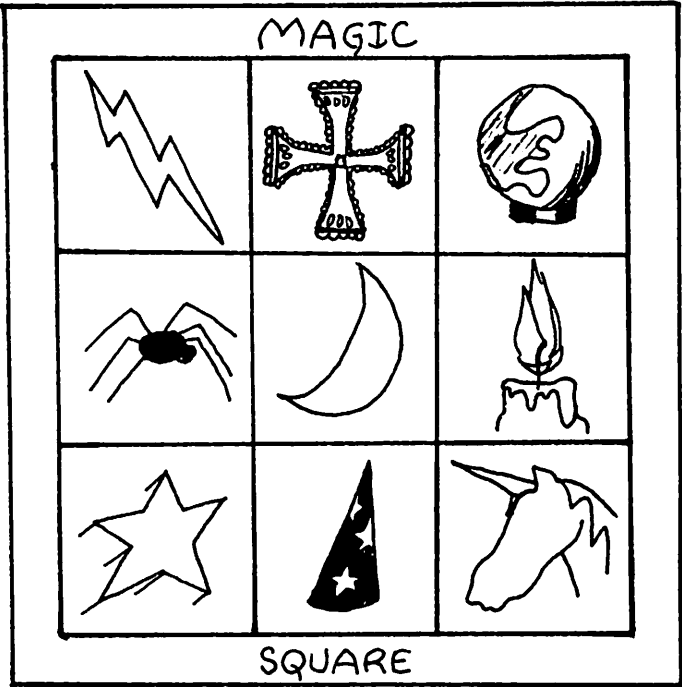
# Magic Square

A magic square of order  $N$  is an arrangement of integers in an  $N \times N$  matrix such that the sums of all the elements in every row and column and along the two main diagonals are equal. This program asks the user to input an odd integer  $N$ , then displays a magic square of order  $N$ .

## Program Listing

```
100 REM MAGIC SQUARE
110 CALL CLEAR
120 DIM A(11,11)
130 INPUT "SIZE=":N
140 N=INT(N)
150 IF INT(N/2)*2=N THEN 130
160 IF N<=0 THEN 130
170 I=1
180 J=(N+1)/2
190 FOR C=1 TO N*N
200 A(I,J)=C
210 IF C-INT(C/N)*N=0 THEN
220 ELSE 240
220 I=I+1
230 GOTO 320
240 IF I=1 THEN 250 ELSE 270
250 I=N
260 GOTO 280
270 I=I-1
280 IF J=N THEN 290 ELSE 310
290 J=1
300 GOTO 320
310 J=J+1
320 NEXT C
```

```
330 FOR I=1 TO N
340 FOR J=1 TO N
350 PRINT A(I,J);
360 NEXT J
370 PRINT
380 NEXT I
390 END
```



18

# Matrix Multiplication

This program multiplies two matrices A and B, and the result is in C matrix. The user specifies the sizes and elements of A and B matrices. The program does the rest.

## Sample Run

```
>RUN
NO. OF ROWS OF A=2
NO. OF COLUMNS OF A=3
NO. OF COLUMNS OF B=1
A(1,1)=
? 1
A(1,2)=
? 3
.
.
.
A MATRIX
1 3 1
2 6 1
B MATRIX
1
3
3
C MATRIX
13
23
** DONE **
```

## Program Listing

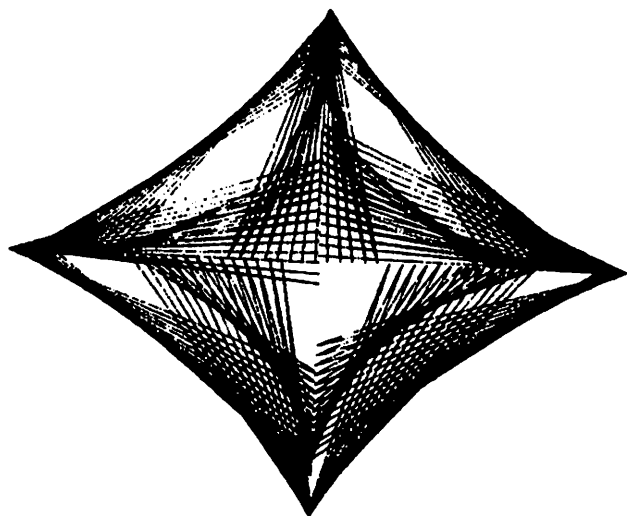


```

100 REM MATRIX MULTIPLICATION
110 CALL CLEAR
120 DIM A(20,20),B(20,20)
130 DIM C(20,20)
140 INPUT "NO. OF ROWS OF A=":L
150 INPUT "NO. OF COLUMNS OF
    A=":M
160 INPUT "NO. OF COLUMNS OF
    B=":N
170 FOR I=1 TO L
180 FOR J=1 TO M
190 PRINT "A(";I;",";J;")="
200 INPUT A(I,J)
210 NEXT J
220 NEXT I
230 FOR J=1 TO M
240 FOR K=1 TO N
250 PRINT "B(";J;",";K;")="
260 INPUT B(J,K)
270 NEXT K
280 NEXT J
290 PRINT "A MATRIX"
300 FOR I=1 TO L
310 FOR J=1 TO M
320 PRINT A(I,J);
330 NEXT J
340 PRINT
350 NEXT I
360 PRINT "B MATRIX"
370 FOR J=1 TO M
380 FOR K=1 TO N
390 PRINT B(J,K);
400 NEXT K
410 PRINT
420 NEXT J
430 FOR I=1 TO L
440 FOR J=1 TO N
450 C(I,J)=0

```

```
460 FOR K=1 TO M
470 C(I,J)=C(I,J)+A(I,K)*B(K,J)
480 NEXT K
490 NEXT J
500 NEXT I
510 PRINT "C MATRIX"
520 FOR I=1 TO L
530 FOR J=1 TO N
540 PRINT C(I,J);
550 NEXT J
560 PRINT
570 NEXT I
580 STOP
590 END
```



# Monte Carlo

In this program, the value of  $\pi$  is estimated by taking random samples  $(X,Y)$  in a unit square and counting the number of samples that lie inside a unit circle centered at the origin. The probability that a sample is in the unit circle is  $\pi/4$ . As the following run shows, after 500 trials, the result is quite close to  $\pi$ .

## Sample Run

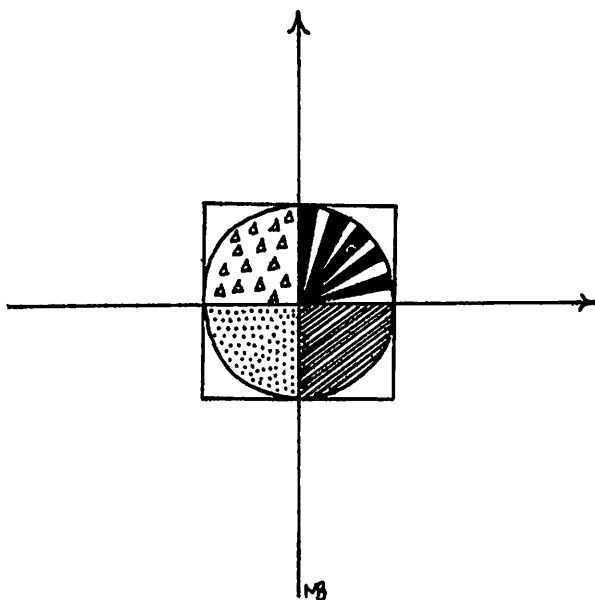
>RUN

1	4
2	2
3	2.66666
.	.
.	.
100	3.2
.	.
.	.
200	3.1
.	.
.	.
500	3.176

## Program Listing

```
100 REM MONTE CARLO
110 RANDOMIZE
120 C=0
130 I=0
140 X=RND
```

```
150 Y=RND
160 IF (X*X+Y*Y)<1 THEN 170
    ELSE 180
170 C=C+1
180 I=I+1
190 R=C/I*4
200 PRINT I,R
210 GOTO 140
220 END
```



# Pascal Triangle

From the coefficients of the following binomial expansions,

$$(a+b)^0 = 1$$

$$(a+b)^1 = a+b$$

$$(a+b)^2 = a^2 + 2ab + b^2$$

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

·  
·  
·

one can form a Pascal triangle

```
      1
     1 1
    1 2 1
   1 3 3 1
```

· · · · ·

This program will print out a Pascal triangle up to the coefficients of a ninth-order binomial expansion.

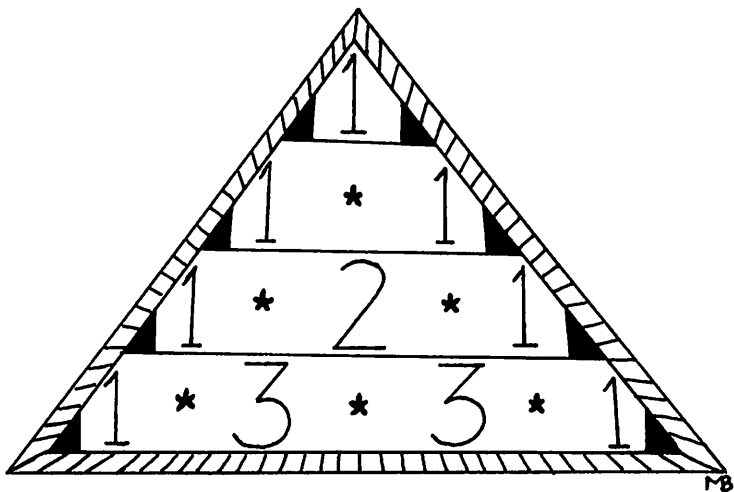
## Program Listing

```
100 REM PASCAL TRIANGLE
110 CALL CLEAR
120 DIM A(11)
130 FOR I=1 TO 11
140 A(I)=0
150 NEXT I
160 A(1)=1
170 FOR I=1 TO 10
```

```

180 B$=""
190 FOR J=1 TO I
200 NB=A(J)
210 B$=B$&" "&STR$(NB)
220 NEXT J
230 PRINT TAB(16-I-INT(I/6)*(I-5));B$
240 PRINT " "
250 FOR K=I TO 1 STEP -1
260 A(K+1)=A(K+1)+A(K)
270 NEXT K
280 NEXT I
290 END

```



# Prime Factors

An integer, which can be evenly divided only by 1 and itself, is called a prime number. For example, 7 is a prime number; whereas 6 is not a prime number. This program prints out all prime factors of an integer.

## Sample Run

```
>RUN
ENTER A NUMBER 1179
ALL PRIME FACTORS OF 1179
3
3
131
```

## Program Listing

```
100 REM PRIME FACTORS
110 INPUT "ENTER A NUMBER ":N
120 PRINT "ALL PRIME FACTORS OF";N
130 N=ABS(INT(N))
140 PF=2
150 IF PF>N THEN 220
160 IF INT(N/PF)*PF<>N THEN 200
170 N=N/PF
180 PRINT PF
190 GOTO 160
200 PF=PF+1
210 GOTO 150
220 END
```

# Prime Numbers

An integer number is said to be a prime number if it is divisible by 1 and itself only. For any positive integer N specified by the user, this program will print out all prime numbers less than or equal to N.

## Program Listing

```
100 REM PRIME NUMBERS
110 CALL CLEAR
120 INPUT "ENTER POSITIVE INTEGER N  ":N
130 DIM A(1000)
140 FOR I=2 TO N
150 A(I)=I
160 NEXT I
170 NL=SQR(N)
180 NL=INT(NL)
190 FOR K=2 TO NL STEP 1
200 LM=2*K
210 IF LM>N THEN 250
220 FOR L=LM TO N STEP K
230 A(L)=0
240 NEXT L
250 NEXT K
260 PRINT "ALL PRIME NUMBERS
    LESS THAN ";N,,,,
270 FOR L=2 TO N
280 IF A(L)=0 THEN 300
290 PRINT A(L);
300 NEXT L
310 END
```



# Zeros of Functions

This program will find a real zero of a function  $F(x)$  defined in statement 20. In our case,  $F(x)=x^7-5$ . User has to provide the values for LOW and HIGH, which define an interval for  $x$ . This program is based on the bisection method. If the function does not have a zero in the interval (LOW,HIGH), then the computer will ask you to change LOW and HIGH.

## Program Listing

```
10 REM ZEROS OF FUNCTIONS
20 DEF F(X)=x^7-5
30 INPUT "LOW=":L
40 INPUT "HIGH=":H
50 IF (F(L)*F(H)<=0.0)THEN 80
60 PRINT "CHANGE LOW OR HIGH"
70 GOTO 30
80 IF (ABS(H-L)<=1.0E-10)THEN 160
90 X=(H+L)/2
100 PRINT H
110 IF (F(X)*F(H)>=0.0)THEN 140
120 L=X
130 GOTO 150
140 H=X
150 GOTO 80
160 PRINT "ZERO OF F(X) IS ";H
170 END
```

ISBN 0-914729-00-4  
**\$6.95**