57 PRACTICAL PROGRAMS & GAMES IN BASIC

BY KEN TRACTON

57

Programs for Everything from Space War Games to Blackjack . . . from Craps to I Ching!

57 PRACTICAL PROGRAMS & GAMES IN BASIC

57 PRACTICAL PROGRAMS & GAMES IN BASIC

BY KEN TRACTON

Radio *Shack*®

FIRST EDITION

Copyright © 1978 by TAB BOOKS

Printed in the United States
of America

Preface

These programs are intended for the computer hobbyist or user who has access to a computer system with standard BASIC language available.

The various programs are written in such a manner that they will operate even with a simplified subset of full BASIC. The advanced programmer, with full BASIC available, can easily modify the programs to take advantage of the functions and capabilities of the full instruction set.

The programs were chosen not just for their intrinsic qualities, but also for their role as teaching aids.

I would like to take this opportunity to thank the following people who helped to make this book possible. Alec Grynspan, who supplied the Bubble-Sort program; Tom McRoberts, who was able to read my notes and typed the manuscript; Laura Semple, who drew the final drafts of the flow charts; Jane, who said programming is simple; and David, who named my Iguana.

I would like to extend a special thank you to Construction Data Systems, who allowed me access to their time-sharing facility on an IBM 370-158 computer systems.

Ken Tracton

Contents

BASIC STATEMENTS

BASIC (Beginners' All-purpose Symbolic Instruction Code) was invented and developed between 1963 and 1964 by John Kemeny and Thomas Kurtz of Dartmouth College. Since its first use in 1964, BASIC has steadily gained popularity as a high-level computer language which the user can easily master. The essential vocabulary is below:

Statement	Example	Definition
CHANGE	CHANGE N$ TO N	assigns to the elements of N the ASCII numeric value of the string N$
DATA	DATA 15, -8, 76,...	the DATA statement assigns appropriate values to the variables listed in the READ statement
DEF	DEF FNR (X, Y) = (X 2 + Y 8)	a single line function is defined by the DEF statement
DIM	DIM Z(3, 4)	dimensions the elements of X as a 3 by 4 matrix
END	END	ends program execution
FNEND	FNEND	a multiline DEF statement must end with a FNEND (function end) statement
FOR-TO	FOR X = 2 TO 66	defines the FOR, NEXT loop
GOTO	GOTO 100	transfers execution to line 100
GOSUB	GOSUB 100	transfers program control to a subroutine commencing at 100
IF-THEN	IF A = X THEN 100	transfers program execution to 100 if the relational test is true
INPUT	INPUT X, Y,...	assigns to the variable(s) the values presented by the user from a user defined device
LET	LET A = V	assigns the value of V to A
NEXT	NEXT X	returns control to the beginning of the FOR-TO loop
ON-GO TO	ON M GO TO 10, 20, 30	as M ranges in values from 1 up to 1st, 2nd,...line number is transferred control, as follows to GO TO statement
PRINT	PRINT "LESLIE"	prints the alphanumeric string within quotation marks
RANDOMIZE	RANDOMIZE	assures each call to the RND produces a different order of random numbers
READ	READ L, K,...	reads values from the DATA statement found in the same program
REM	REM AREA	remark is placed in the program to be used only during listing as a debugging aid

9

Statement	Example	Definition
RESTORE	RESTORE	restores the data pointer
RETURN	RETURN	returns program execution to the next instruction following the subroutine call
RND	RND	produced a random number
STOP	STOP	stops program execution

LIBRARY FUNCTIONS

ABS	absolute value
ATN	arctangent
ASC	converts a ASCII character to its numeric value and assigns it
CHR$	converts a numeric value to its ASCII character and assigns it
COS	cosine
COT	cotangent
DET	determinant
EXP	raise e to the x power
INT	truncates to an integer
LOC	determines the position of the pointer
LOF	determines the last storage location in a file
LOG	returns the log (base e) of the argument
RND	produces a random variable
SGN	determines the sign of a variable
SIN	sine
SQR	square root
TAB	positions printing head of a printer (CRT or LINE)
TAN	tangent

BASIC COMMAND FUNCTIONS

BYE	terminates time-sharing session
CATALOG	lists names of all files saved
GOOD-BYE	same as BYE
LIST	produced a listing of the current file
NEW	specifies that a new file is being formed
OLD	accesses an existing file
RENAME	allows the name of a current file to be changed
RUN	causes current program to be executed
SAVE	causes current program to be saved (stored)
SCRATCH	deletes the current file
SYSTEM	transfers control from BASIC to the system's monitor
UNSAVE	cancels storage of a file

exponential ↑
multiplication *

PROGRAMS

The programs appear in alphabetical order. Each one contains any applicable formulae, followed by an example of using the Program, the Flow Chart, and lastly, the Program itself.

ANNULAR SECTIONS

This program computes the various parameters; moment of inerta, polar moment of inertia and area; connected with an annular section.

FORMULAE
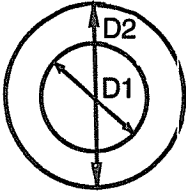
ANNULAR SECTION $\quad I = \dfrac{\pi \ (d_2^4 - d_1^4)}{64} \quad A = \dfrac{\pi \ (d_2^2 - d_1^2)}{4}$

$$J = \dfrac{\pi \ (d_2^4 - d_1^4)}{32}$$

where I and J is in (in.^{3}) and A is in (in.^{2}).

EXAMPLE

INSIDE DIAMETER (D1) =
?
3
OUTSIDE DIAMETER (D2) =
?
4.11
MOMENT OF INERTIA = 10
POLAR MOMENT OF INERTIA = 20
AREA OF SECTION = 6.18
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
* END

ANNULAR SECTIONS

START

PROMPT + INPUT
D1

PROMPT + INPUT
D2

CALCULATE
I, J AND A

OUTPUT
PARAMETERS

PROMPT + INPUT
CONTINUE
COMMAND

IS L = 1?   YES

NO

END

13
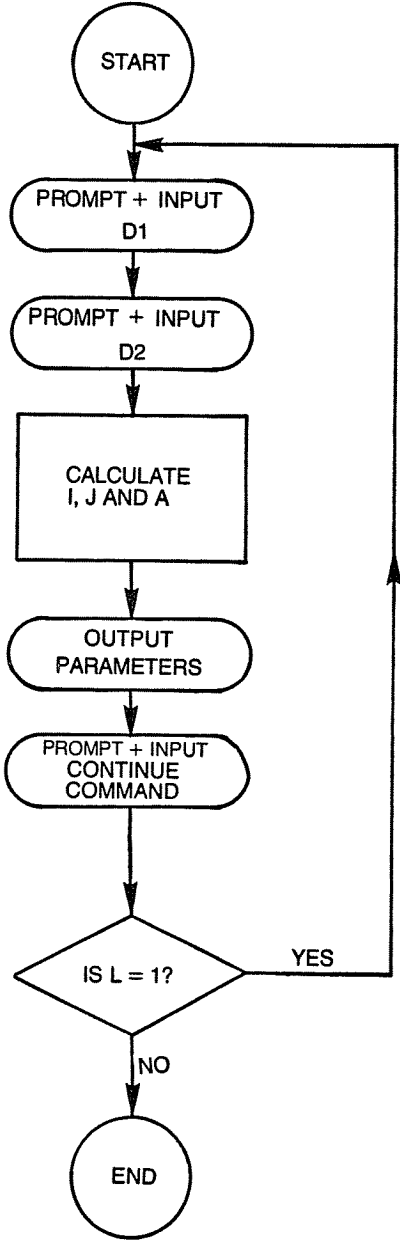
ANNULAR SECTIONS

```
10    REM THIS PROGRAM COMPUTES THE VARIOUS
20    REM PARAMETERS CONNECTED WITH AN ANNU-
      LAR
30    REM SECTION
40    PRINT "INSIDE DIAMETER (D1) = ";
50    INPUT D1
60    PRINT "OUTSIDE DIAMETER (D2) = ";
65    INPUT D2
70    LET P = 3.14159
80    LET I = (P*((D2↑4) − (D1↑4)))/64
90    LET J = I*2
100   LET A = (P*((D2↑2) − (D1↑2)))/4
110   PRINT "MOMENT OF INERTIA = ";I
120   PRINT "POLAR MOMENT OF INERTIA = ";J
130   PRINT "AREA OF SECTION = ";A
140   PRINT
150   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
160   INPUT L
170   IF L = 1 THEN 190
180   STOP
190   PRINT
200   GO TO 40
210   END
```

ARITHMETIC MEAN

After each sample is entered the computer responds with sample number and the current mean. The program will continue to run until the operator types an exit or a break command.

FORMULA

$$\overline{A} = \frac{1}{N} \sum_{i=1}^{N} a_i$$

EXAMPLE

ENTER SAMPLE
?
5
   N = 1   SAMPLE = 5   CURRENT MEAN = 5
ENTER SAMPLE
?
67
   N = 2   SAMPLE = 67   CURRENT MEAN = 36
ENTER SAMPLE
?
5

   N = 3   SAMPLE = 5   CURRENT MEAN = 25.6666

ENTER SAMPLE
?
45
   N = 4   SAMPLE = 45   CURRENT MEAN = 30.5
ENTER SAMPLE
?
12
   N = 5   SAMPLE = 12   CURRENT MEAN = 26.8000
ENTER SAMPLE
?
123
   N = 6   SAMPLE = 123   CURRENT MEAN = 42.8333
ENTER SAMPLE
?
0
   N = 7   SAMPLE = 0   CURRENT MEAN = 36.7142
ENTER SAMPLE
?
*END

ARITHMETIC MEAN
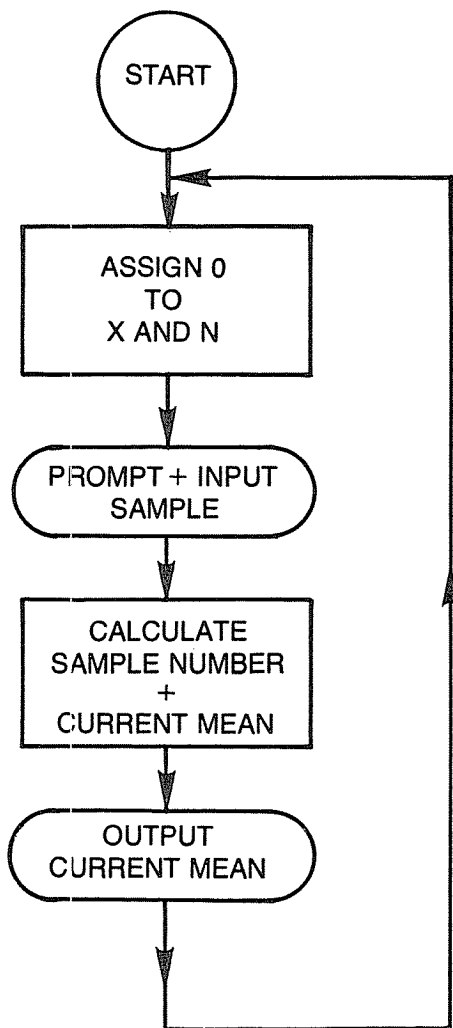
ARITHMETIC MEAN

```
10    REM THIS PROGRAM COMPUTES THE  ARITHME-
      TIC
20    REM MEAN
30    LET X = 0
40    LET N = 0
50    PRINT "ENTER SAMPLE";
60    INPUT W
70    LET N = N + 1
80    LET X = X + W
90    LET A = X/N
100   PRINT "N = "; N, "SAMPLE = "; W, "CURRENT MEAN
      = "; A
110   GOTO 50
120   END
```

ARITHMETIC PROGRESSION

From the following information; first term, common difference and number of terms, this program computes the arithmetic progression.

FORMULA

$$A, A + D, A + 2D, ... A + ((N - 1)D)$$

EXAMPLE

FIRST TERM =
?
10
COMMON DIFFERENCE =
?
2
NUMBER OF TERMS =
?
5
FOR TABLE TYPE 1, IF NOT TYPE 0
?
1
ARITHMETIC PROGRESSION

TERM NUMBER	TERM PROGRESSION
1	10
2	12
3	14
4	16
5	18

SUM = 70
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

ARITHMETIC PROGRESSION
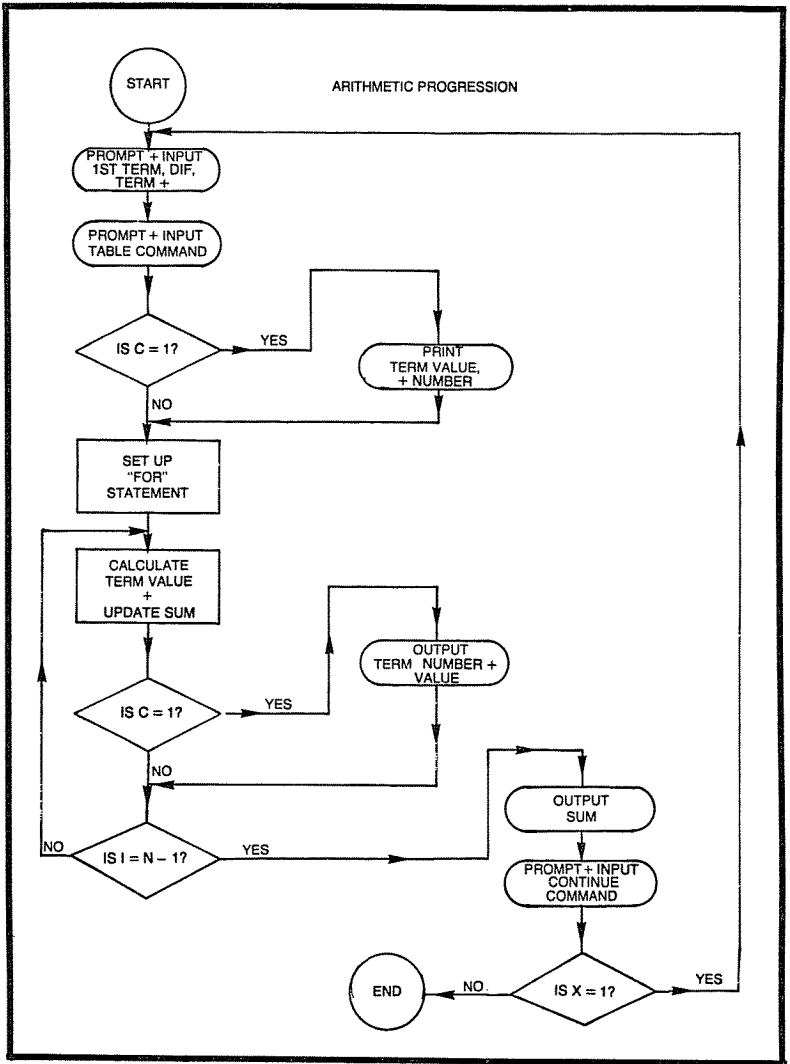
ARITHMETIC PROGRESSION

10   REM THIS PROGRAM COMPUTES ARITHMETIC PROGRESSION

20   PRINT "FIRST TERM = ";

30   INPUT A

40   PRINT "COMMON DIFFERENCE = ";

50   INPUT D

60   PRINT "NUMBER OF TERMS = ";

```
70    INPUT N
80    PRINT "FOR TABLE TYPE 1, IF NOT TYPE 0";
90    INPUT C
100   IF C = 1 THEN 120
110   GOTO 140
120   PRINT "ARITHMETIC PROGRESSION"
130   PRINT "TERM NUMBER", "TERM VALUE"
140   LET J = 0
150   FOR I = 0 TO N − 1
160   LET K = I + 1
170   LET L = A + (I*D)
180   LET J = J + L
190   IF C = 1 THEN 210
200   GOTO 220
210   PRINT K, L
220   NEXT I
230   PRINT "SUM = ";J
240   PRINT
250   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
260   INPUT X
270   IF X = 1 THEN 290
280   STOP
290   PRINT
300   GOTO 20
310   END
```

BLACKJACK

Blackjack, or the game of 21, is played against the computer, it being the dealer. Cards are dealt from a self-replenishing deck. Standard rules are as follows:

- Blackjack wins unless the dealer also gets blackjack, in which case there is no winning player.
- The highest score below 21 wins.
- The dealer must draw a card if he is below 17, but must stand if he has 17 or greater.
- Aces count as 11 unless it would force a hand over 21, in which case the ace counts as 1.

EXAMPLE

```
RUN
IF INSTRUCTIONS ARE REQUIRED TYPE YES
IF NOT TYPE NO
?
NO
***GOOD-LUCK-----MAY THE BEST ONE WIN***
THE DEALER HAS A 9 SHOWING
YOU HAVE A 5 AND A 11
YOUR TOTAL IS 16
DO YOU WANT A HIT, OR DO YOU STAND
?
HIT
YOUR CARD IS 4
DO YOU WANT A HIT, OR DO YOU STAND
?
STAND
THE DEALER HAS 12
THE DEALER DRAWS A 5
HIS TOTAL IS 17
YOU HAVE 20
YOU HAVE WON!!!
DO YOU WISH TO PLAY AGAIN
TYPE YES OR NO
?
NO
BLACKJACK SAYS GOOD-BYE
*END
```
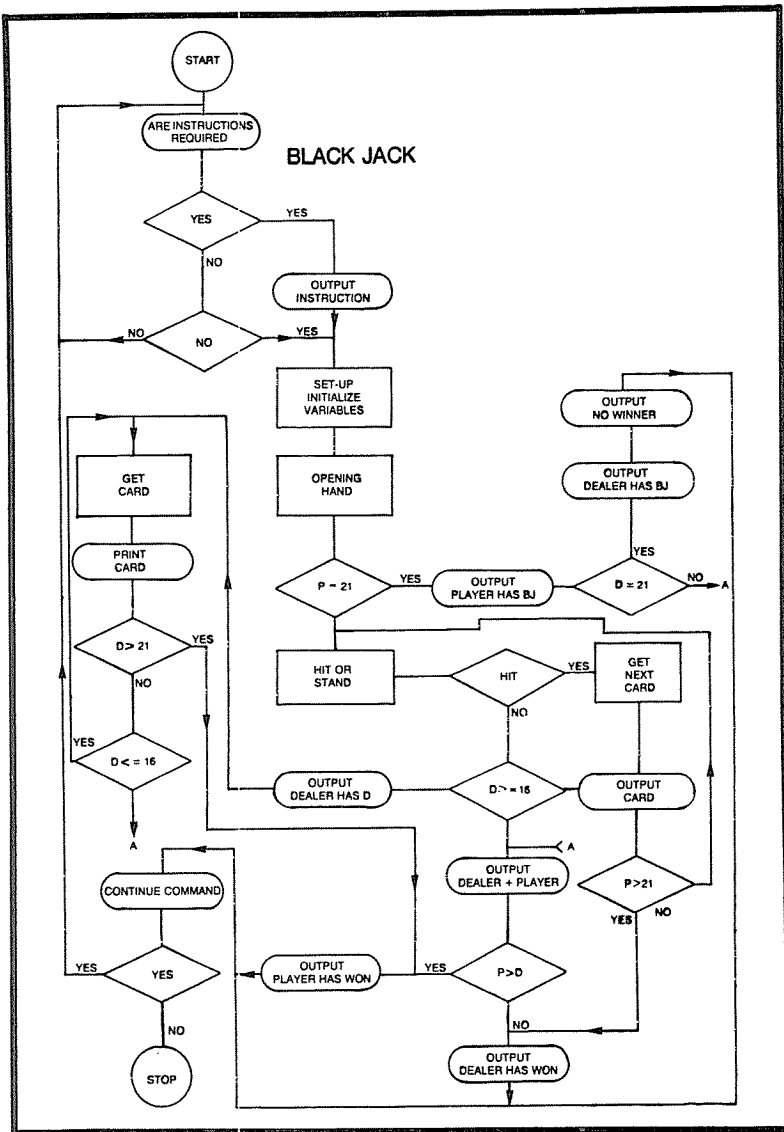
BLACK JACK

BLACKJACK

```
10   REM BLACKJACK
20   PRINT "IF INSTRUCTIONS ARE REQUIRED TYPE
     YES"
30   PRINT "IF NOT TYPE NO"
40   INPUT C$
50   IF C$ = "YES" THEN 90
```

```
60   IF C$ = "NO" THEN 340
70   PRINT "INVALID RESPONSE"
80   GOTO 20
90   PRINT
100  PRINT"**************BLACKJACK**************"
110  PRINT
120  PRINT "THE COMPUTER AS THE DEALER, DEALS
           TWO CARDS TO ITSELF"
130  PRINT "AND TWO CARDS TO THE PLAYER. THE
           PLAYER'S TWO CARDS"
140  PRINT "ARE SHOWN FACE UP, WHILE ONLY ONE OF
           THE DEALER'S"
150  PRINT "CARDS IS SHOWN. BOTH THE DEALER AND
           THE PLAYER"
160  PRINT "MAY DRAW ADDITIONAL CARDS."
170  PRINT "THE PLAYER'S GOAL IS TO REACH 21 OR
           LESS, BUT"
180  PRINT "BE CLOSER TO 21 THAN THE DEALER's
           HAND."
190  PRINT "IF THE PLAYER'S OR THE DEALER'S HAND
           TOTALS"
200  PRINT "GREATER THAN 21 HE IS BUSTED! THE
           KING"
210  PRINT "THE QUEEN AND THE JACK ALL COUNT AS
           10 POINTS."
220  PRINT  "ALL  OTHER  CARDS  EXCEPT  THE
           ACE COUNT AS THEIR FACE"
230  PRINT "VALUE SHOWS. THE ACE COUNTS AS 11
           UNLESS THIS"
240  PRINT "WOULD CAUSE THE HAND TO BE OVER 21,
           IN THAT"
250  PRINT "CASE THE ACE COUNTS AS 1."
260  PRINT "IF BOTH THE DEALER AND THE PLAYER
           GET BLACKJACK"
270  PRINT "WHICH IS A TWO CARD HAND TOTALING 21"
280  PRINT "NEITHER WINS, IT IS A PUSH"
290  PRINT "IF THE DEALER'S HAND IS BELOW OR
           EQUAL TO 16"
300  PRINT "HE MUST DRAW, AFTER 17 THE DEALER
           MUST STAND"
310  PRINT "TO RECEIVE A CARD YOU WANT A HIT-"
320  PRINT "TO STOP WHERE YOU ARE, YOU STAND-"
330  PRINT
```

```
340    PRINT"***GOOD-LUCK-----MAY THE BEST ONE
          WIN***"
350    REM 1ST HAND
355    RANDOMIZE
360    LET D = 0
370    LET P = D
380    GOSUB 820
390    LET D1 = C
400    GOSUB 820
410    LET D2 = C
420    GOSUB 890
430    LET P1 = C
440    GOSUB 890
450    LET P2 = 3
460    PRINT
470    PRINT "THE DEALER HAS A ";D1;" SHOWING"
480    PRINT "YOU HAVE A ";P1;" AND A ";P2
490    PRINT "YOUR TOTAL IS ";P1 + P2
500    LET D = D1 + D2
510    LET P = P1 + P2
520    IF P = 21 THEN 640
530    GOSUB 960
540    IF L = 1 THEN 690
550    IF D< = 16 THEN 740
560    PRINT "THE DEALER HAS ";D
570    PRINT "YOU HAVE ";P
580    IF P > D THEN 620
590    REM WIN OR LOSS STATEMENTS
600    PRINT "THE DEALER HAS WON!!!"
610    GOTO 1060
620    PRINT "YOU HAVE WON!!!"
630    GOTO 1060
640    PRINT "***YOU HAVE BLACKJACK***"
650    IF D = 21 THEN 670
660    GOTO 560
670    PRINT "THE DEALER ALSO HAS BLACKJACK,
          SORRY NO WINNER"
680    GOTO 1060
690    GOSUB 890
700    PRINT "YOUR CARD IS ";C
710    IF P > 21 THEN 600
730    GOTO 530
740    PRINT "THE DEALER HAS ";D
```

```
750    GOSUB 820
760    LET D = D + C
770    PRINT "THE DEALER DRAWS A ";C
780    PRINT "HIS TOTAL IS ";D
790    IF D > 21 THEN 620
800    IF D < = 16 THEN 750
810    GOTO 560
820    LET C = 1 + INT(11*RND)
830    IF C = 11 THEN 850
840    GOTO 880
850    IF D + C > 21 THEN 870
860    GOTO 880
870    LET C = 1
880    RETURN
890    LET C = 1 + INT(11*RND)
900    IF C = 11 THEN 920
910    GOTO 950
920    IF P + C> 21 THEN 940
930    GOTO 950
940    LET C = 1
950    RETURN
960    PRINT "DO YOU WANT A HIT, OR DO YOU STAND"
970    INPUT Q$
980    IF Q$ = "HIT" THEN 1020
990    IF Q$ = "STAND" THEN 1040
1000   PRINT "INVALID RESPONSE"
1010   GOTO 960
1020   LET L = 1
1030   GOTO 1050
1040   LET L = 0
1050   RETURN
1060   PRINT
1070   PRINT "DO YOU WISH TO PLAY AGAIN"
1080   PRINT "TYPE YES OR NO"
1090   INPUT L$
1100   IF L$ = "YES" THEN 1130
1110   PRINT "BLACKJACK SAYS GOOD-BYE"
1120   STOP
1130   PRINT
1140   GOTO 20
1150   END
```

BUBBLE SORT

This program is actually a subroutine rather than a stand-alone program. It is intended to be used in conjunction with larger programs which require that data be placed in an array in ascending sequence.

BUBBLE SORT

```
10    REM THIS PROGRAM IS A BUBBLE SORT, WHICH
      PLACES THE
20    REM VALUES IN AN ARRAY IN ASCENDING SE-
      QUENCE
30    REM IT IS INTENDED TO BE AN EXAMPLE AND
      NOT A
40    REM SPECIFIC CASE.
50    REM THIS PROGRAM MAY BE CONVERTED INTO A
      SUBROUTINE
60    REM FOR USE IN A LARGER PROGRAM
70    REM THE ARRAY IS Z OF LENGTH N
80    REM Z IS ASSUMED TO HAVE BEEN DECLARED IN A
      DIM
90    REM STATEMENT AND N SET AHEAD OF TIME
100   LET I = N - 1
110   FOR J = 1 TO I
120   LET K = J + 1
130   FOR L = N TO K STEP - 1
140   IF Z(L) > Z(J) THEN 210
150   REM SAVE FIRST VALUE
160   LET T = Z(L)
170   LET Z(L) = Z(J)
180   LET Z(J) = T
190   REM IF Z IS THE KEY FOR SORTING MULTIPLE
200   REM ARRAYS AND EXTRA CODE HERE
210   NEXT L
220   NEXT J
230   RETURN
```

CHI-SQUARE EVALUATION

This program computes the chi-square evaluation from the input-ed observed and expected frequencies. To terminate the evaluation the use simply inputs a 0 for the last expected frequency.

FORMULA

$$X^2 = \sum_{i=1}^{N} \frac{(O_1 - E_1)^2}{E_1}$$

EXAMPLE

OBSERVED FREQUENCIES =
?
10
EXPECTED FREQUENCIES =
?
10
OBSERVED FREQUENCIES =
?
0
OBSERVED FREQUENCIES =
?
0
CHI-SQUARE = 0
****
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

CHI-SQUARE EVALUATION

(Flowchart)
- START
- ASSIGN Ø TO J
- PROMPT + INPUT OBSERVE FREQUENCIES
- PROMPT + INPUT EXPECTED FREQ.
- IS E = O
  - YES → PRINT CHI-SQUARE → PROMPT + INPUT CONTINUE COMMAND → IS L = 1?
    - YES (loops back)
    - NO → STOP
  - NO → UPDATE CHI-SQUARE CALCULATE (loops back)
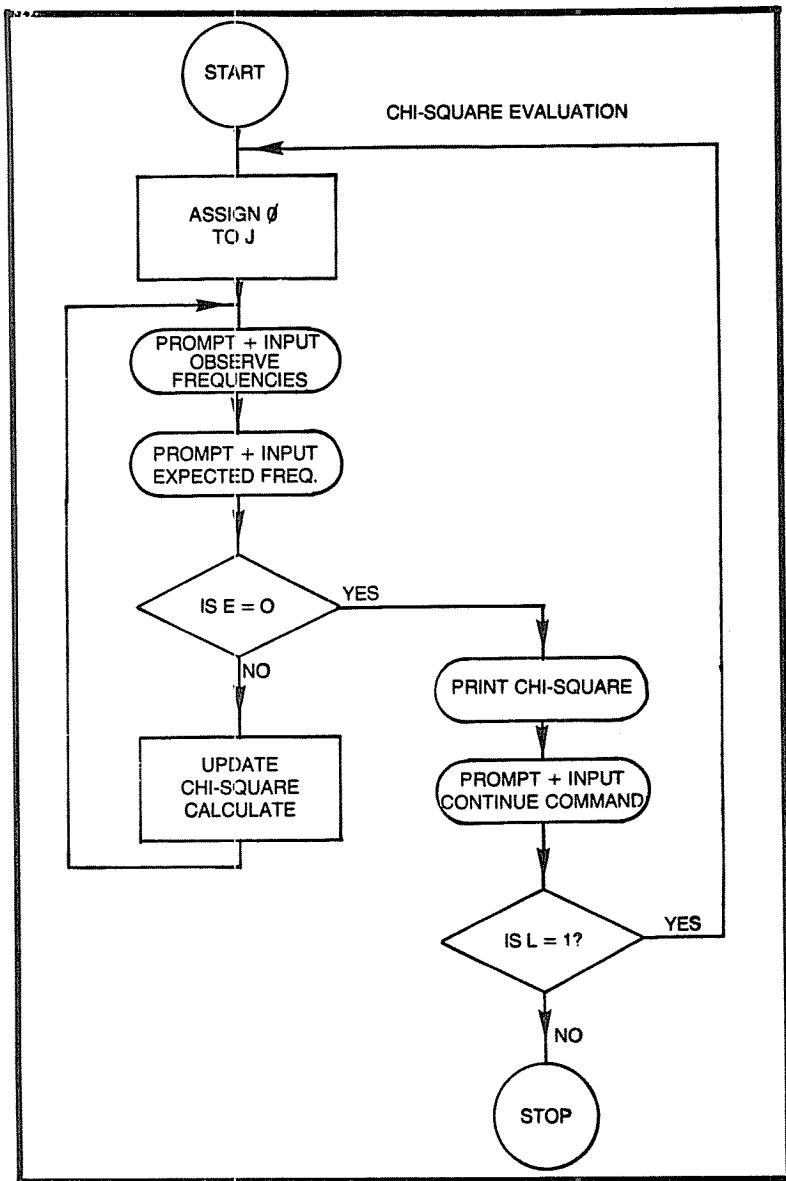
CHI-SQUARE EVALUATION

```
10    REM THIS PROGRAM COMPUTES CHI-SQUARE
20    REM EVALUATION ON THE OBSERVED TO
30    REM EXPECTED FREQUENCIES
40    LET J = 0
```

```
50    PRINT "OBSERVED FREQUENCIES = "
60    INPUT D
70    PRINT "EXPECTED FREQUENCIES = "
80    INPUT E
90    IF E = 0 THEN 130
100   LET K = ((D − E)↑2)/E
110   LET J = J + K
120   GOTO 50
130   PRINT "CHI-SQUARE = ";J
140   PRINT "********************"
150   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
160   INPUT L
170   IF L = 1 THEN 190
180   STOP
190   PRINT
200   GOTO 40
210   END
```

CIRCLE DETERMINED BY THREE POINTS

The user inputs three noncolinear points, and the program responds with the center and the radius of the circle thus generated.

FORMULAE

$$y_0 = \frac{K_2 - K_1}{N_2 - N_1} \qquad X_0 = K_2 - N_2 Y_0$$

$$r = (X_3 - X_0)^2 + (Y_3 - Y_0)^2$$

$$K_1 = \frac{(X_2 - X_1)(X_2 + X_1) + (Y_2 - Y_1)(Y_2 + Y_1)}{2(X_2 - X_1)}$$

$$K_2 = \frac{(X_3 - X_1)(X_3 + X_1) + (Y_3 - Y_1)(Y_3 + Y_1)}{2(X_3 - X_1)}$$

$$N_1 = \frac{Y_2 - Y_1}{X_2 - X_1} \qquad N_2 = \frac{Y_3 - Y_1}{X_3 - X_1}$$

EXAMPLES

INPUT X1,Y1
?
2,3
INPUT X2,Y2
?
5,4
INPUT X3,Y3
?
6,4.5
CENTER X0,Y0 = − 1.25,17.25
RADIUS = 15.10
TO CONTINUE TYPE 1, IF NOT TYPE 0
?
0
*END

30

CIRCLE DETERMINED BY THREE POINTS

```
                    ┌─────────┐
                    │  START  │
                    └─────────┘
                         │
                         │
              ┌──────────────────┐
              │  PROMPT AND      │
              │  INPUT           │
              │  POINTS          │
              └──────────────────┘
                         │
              ┌──────────────────┐
              │  COMPUTE         │
              │  Xo, Yo, R       │
              └──────────────────┘
                         │
              ┌──────────────────┐
              │  PRINT           │
              │  Xo, Yo, R       │
              └──────────────────┘
                         │
              ┌──────────────────┐
              │  PRINT AND       │
              │  INPUT CONTINUE  │
              │  COMMAND         │
              └──────────────────┘
                         │
                    ◇ IS L = 1? ◇ ──── YES
                         │
                         │ NO
                    ┌─────────┐
                    │  STOP   │
                    └─────────┘
```
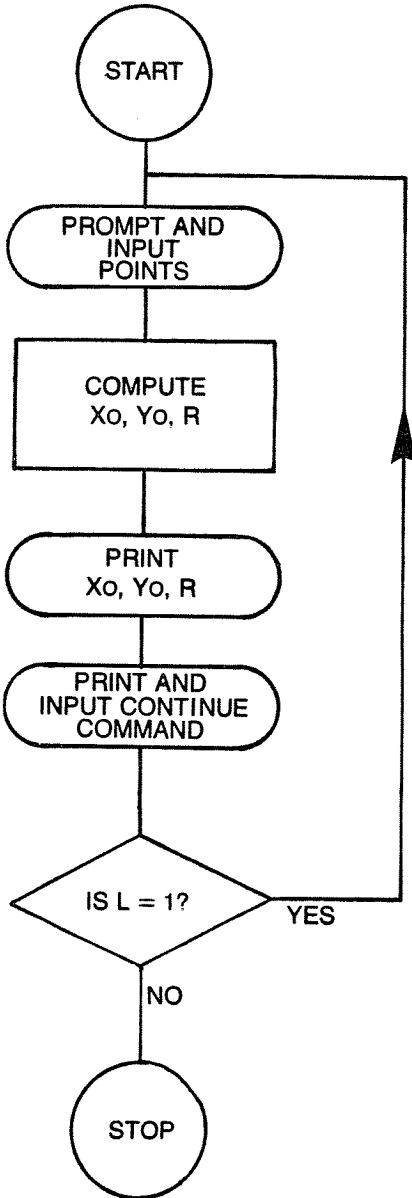
CIRCLE DETERMINED BY THREE POINTS

```
10    REM THIS PROGRAM COMPUTES THE CENTER
      AND RADIUS
20    REM OF A CIRCLE FROM 3 NON-COLINEAR POINTS
30    PRINT "INPUT X1,Y1";
40    INPUT X1, Y1
50    PRINT "INPUT X2,Y2";
60    INPUT X2,Y2
70    PRINT "INPUT X3,Y3";
80    INPUT X3,Y3
90    LET A = (Y2 − Y1)/(X2 − X1)
100   LET B = (Y3 − Y1)/(X3 − X1)
110   LET C = ((X2 − X1)*(X2 + X1))*
      + ((Y2 − Y1)*(Y2 + Y1))
120   LET D = C/(2*(X2 − X1))
130   LET E = ((X3 − X1)*(X3 + X1)) + ((Y3 − Y1)*
      (Y3 + Y1))
140   LET F = E/(2*(X3 − X1))
150   LET YO = (F − D)/(B − A)
160   LET XO = F − (B*YO)
170   LET R = SQR((X3 − X0)↑2 + (Y3 − Y0)↑2)
180   PRINT "CENTER X0,Y0 = ";X0;",";Y0
190   PRINT "RADIUS = ";R
200   PRINT
210   PRINT "TO CONTINUE TYPE 1, IF NOT TYPE 0"
220   INPUT L
230   IF L = 1 THEN 250
240   STOP
250   PRINT
260   GOTO 30
270   END
```

CIRCULAR SECTIONS

This program computes the various parameters: moment of inertia, polar moment of inertia and area connected within a circular section.

FORMULAE

CIRCULAR SECTION
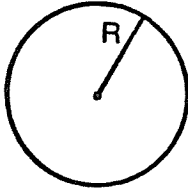
$$I = \frac{\pi D^4}{64} \qquad A = \frac{\pi D^2}{4}$$

$$J = \frac{\pi D^4}{32}$$

where I & J are in in⁴ and A = in²

EXAMPLE
RADIUS
?
5
MOMENT OF INERTIA = 490.873
POLAR MOMENT OF INTERTIA = 981.746
AREA OF SECTION = 78.5397
****
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

CIRCULAR SECTIONS

```
        ( START )
            │
            ▼
    ┌───────────────┐
    │   PROMPT +    │◄──────────┐
    │  INPUT RADIUS │           │
    └───────────────┘           │
            │                   │
            ▼                   │
    ┌───────────────┐           │
    │   CALCULATE   │           │
    │  PARAMETERS   │           │
    └───────────────┘           │
            │                   │
            ▼                   │
    ┌───────────────┐           │
    │    OUTPUT     │           │
    │  PARAMETERS   │           │
    └───────────────┘           │
            │                   │
    ┌───────────────┐           │
    │   CONTINUE    │           │
    │ MESSAGE + INPUT│          │
    └───────────────┘           │
            │                   │
            ▼          YES      │
         ◇ IS L = 1? ◇─────────┘
            │
           NO
            │
            ▼
        ( STOP )
```

34
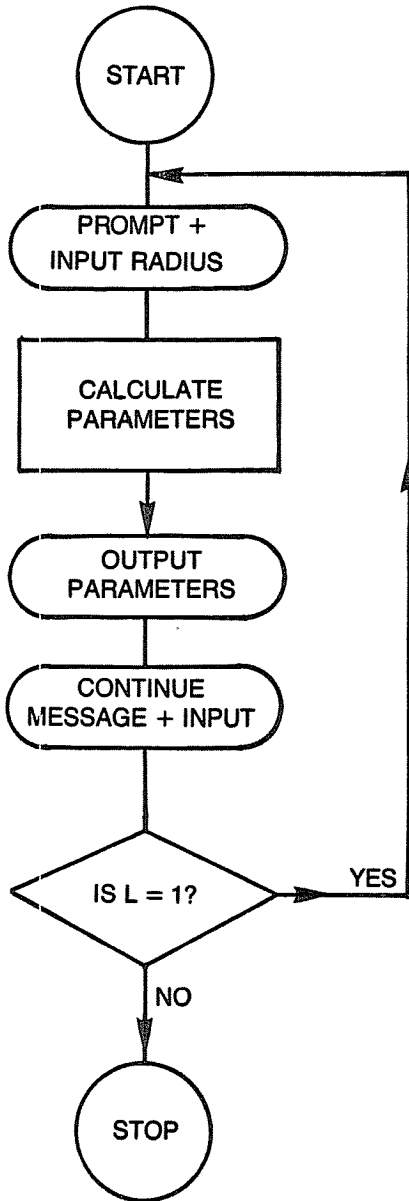
CIRCULAR SECTION

```
10    REM THIS PROGRAM COMPUTES THE PARA-
      METERS
20    REM CONNECTED WITH A CIRCULAR SECTION
30    PRINT "RADIUS"
40    INPUT R
50    LET P = 3.14159
60    LET D = 2*R
70    LET I = (P*(D↑4))/64
80    LET J = I*2
90    LET A = (P*(D↑2))/4
100   PRINT "MOMENT OF INERTIA = ";I
110   PRINT "POLAR MOMENT OF INERTIA = ";J
120   PRINT "AREA OF SECTION = ";A
130   PRINT "*************************"
140   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
150   INPUT L
160   IF L = 1 THEN 180
170   STOP
180   PRINT
190   GOTO 30
200   END
```

COMPOUNDED AMOUNTS

This program applies to an amount of principle placed into an account and compounded periodically, with no further deposits.

FORMULAE

$$X = Y(1 + I)^N \qquad\qquad N = \frac{ln(X/Y)}{ln(1 + I)}$$

$$Y = X(1 + I)^{-N} \qquad\qquad L = Y((1 + I)^N - 1)$$

where $N = $ *number of Time periods*
$\qquad I = $ *interest rate (decimal)* $\quad L = $ *interest*
$Y = $ *present value*
$X = $ *future value*

EXAMPLE

FUTURE VALUE(1)
PRESENT VALUE (2)
INTEREST (3)
PERIODIC INTEREST RATE (4)
NUMBER OF TIME PERIODS (5)
ENTER 1 TO 5
?
1
PRESENT VALUE =
?
100
INTEREST RATE =
?
10
NUMBER OF TIME PERIODS =
?
4
FUTURE VALUE = 146.41
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
ENTER 1 TO 5
?
2

```
FUTURE VALUE =
?
200
INTEREST RATE =
?
.1
NUMBER OF TIME PERIODS =
?
5
PRESENT VALUE =124.18
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
ENTER 1 TO 5
?
3
PRESENT VALUE =
?
300
INTEREST RATE =
?
.2
NUMBER OF TIME PERIODS =
?
6
INTEREST =595.80
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
ENTER 1 TO 5
?
4
PRESENT VALUE =
?
100
FUTURE VALUE =
?
200
NUMBER OF TIME PERIODS =
?
5
INTEREST RATE = .15
```

TYPE 1 TO CONTINUE, 0 TO STOP
?
1
ENTER 1 TO 5
?
5
FUTURE VALUE ==
?
500
PRESENT VALUE =
?
250
INTEREST RATE =
?
.2
NUMBER OF TIME PERIODS =3.8
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

COMPOUNDED AMOUNTS

START

PRINT OPTIONS

PROMPT AND
INPUT DESIRED
OPTION

CALCULATE
DESIRED
OPTION

PRINT AND
INPUT OPTION
VARIABLES

PRINT
OPTION
RESULT

PROMPT AND
INPUT CONTINUE
COMMAND

S L = 1   YES

NO

STOP
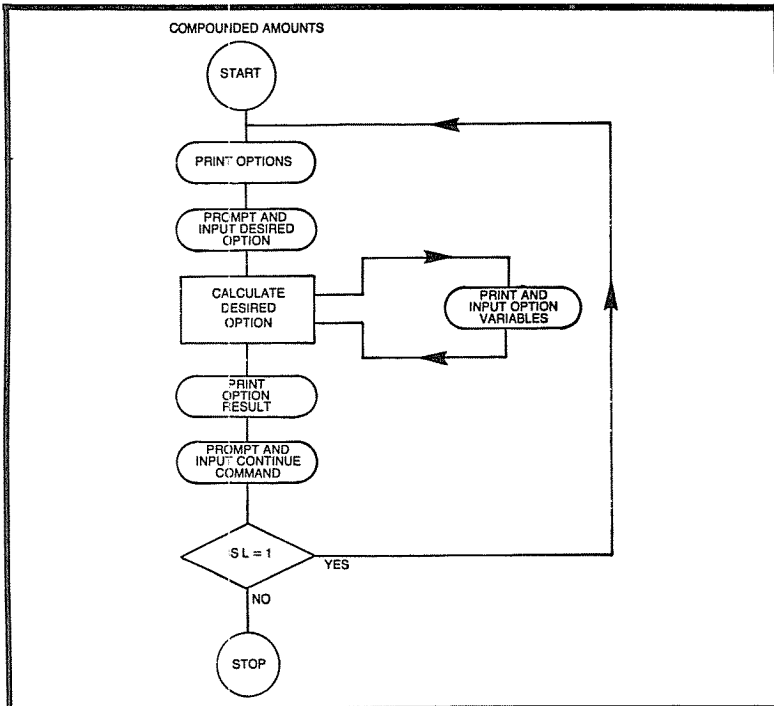
COMPOUNDED AMOUNTS

```
10    REM COMPOUNDED AMOUNTS
20    PRINT "FUTURE VALUE (1)"
30    PRINT "PRESENT VALUE (2)"
40    PRINT "INTEREST (3)"
50    PRINT "PERIODIC INTEREST RATE (4)"
60    PRINT "NUMBER OF TIME PERIODS (5)"
70    PRINT "ENTER 1 TO 5";
80    INPUT A
90    ON A GOTO 100,160,220,280,340
100   GOSUB 410
110   GOSUB 440
120   GOSUB 470
130   LET X = Y*((1 +I)↑N)
140   PRINT "FUTURE VALUE = ";X
150   GOTO 390
160   GOSUB 500
170   GOSUB 440
180   GOSUB 470
190   LET Y = X*((1 +I)↑N)
200   PRINT "PRESENT VALUE = ";Y
210   GOTO 390
220   GOSUB 410
230   GOSUB 440
240   GOSUB 470
250   LET L = Y*((1 +I↑N) - 1)
260   PRINT "INTEREST = ";L
270   GOTO 390
280   GOSUB 410
290   GOSUB 500
300   GOSUB 470
310   LET I = ((X/Y)↑(1/N)) - 1
320   PRINT "INTEREST RATE = ";I
330   GOTO 390
340   GOSUB 500
350   GOSUB 410
360   GOSUB 440
370   LET N = (LOG(X/Y))/(LOG(1 + I))
380   PRINT "NUMBER OF TIME PERIODS = ";N
390   PRINT
400   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
401   INPUT L
```

```
402    IF L = 1 THEN 404
403    STOP
404    PRINT
405    GOTO 70
410    PRINT "PRESENT VALUE = ";
420    INPUT Y
430    RETURN
440    PRINT "INTEREST RATE = ";
450    INPUT I
460    RETURN
470    PRINT "NUMBER OF TIME PERIODS = ";
480    INPUT N
490    RETURN
500    PRINT "FUTURE VALUE = ";
510    INPUT X
520    RETURN
530    END
```

COORDINATE TRANSLATION AND/OR ROTATION

This program will compute rectangular-coordinate translation and/or rotation. The origin is translated from (0,0) to a new point (X,Y) and the X,Y axes are rotated to an angle A to give new axes X^1, Y^1.

FORMULAE

$$X^1 = (X - X_o)Cos\ \alpha + (Y - Y_o)Sin\ \alpha$$
$$Y^1 = -(X - X_o)Sin\ \alpha + (Y - Y_o)Cos\ \alpha$$
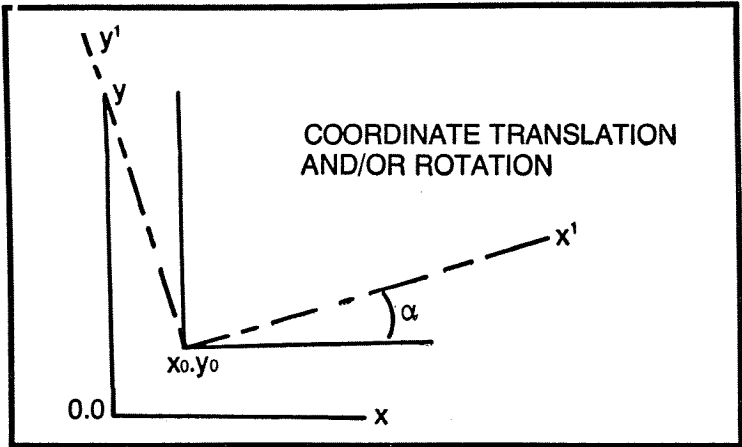
EXAMPLES

ANGLE OF ROTATION IN DEGREES=
?
45
COORDINATES OF NEW ORIGIN (X0,Y0)=
?
5,6
OLD COORDINATES OF DATA POINT (X1, Y1)=
?
8,9
SPECIFIED ROTATION = 45
NEW COORDINATES OF DATA POINT
X2 =4.24
Y2 = 0
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

COORDINATE TRANSLATION AND/OR ROTATION

```
                    ( START )
                        |
              +---------+---------+
              |                   |
       +-------------+            |
       | PROMPT AND  |            |
       | INPUT ANGLE |            |
       | OF ROTATION |            |
       +-------------+            |
              |                   |
       +-------------+            |
       | PROMPT AND  |            |
       | INPUT NEW   |            |
       |   ORIGIN    |            |
       +-------------+            |
              |                   |
       +-------------+            |
       | PROMPT AND  |            |
       | INPUT OLD   |            |
       | DATA POINTS |            ^
       +-------------+            |
              |                   |
       +-------------+            |
       | CALCULATE   |            |
       |    NEW      |            |
       | DATA POINTS |            |
       +-------------+            |
              |                   |
       +--------------+           |
       |PRINT SPECIFIED|          |
       |  ROTATION    |           |
       +--------------+           |
              |                   |
       +-------------+            |
       | PRINT NEW   |            |
       | COORDINATES |            |
       +-------------+            |
              |                   |
       +-------------+            |
       | PROMPT AND  |            |
       |INPUT CONTINUE|           |
       |  COMMAND    |            |
       +-------------+            |
              |                   |
            / IS L = 1? \ --------+
            \           /   YES
              |
              | NO
              |
          ( STOP )
```

42
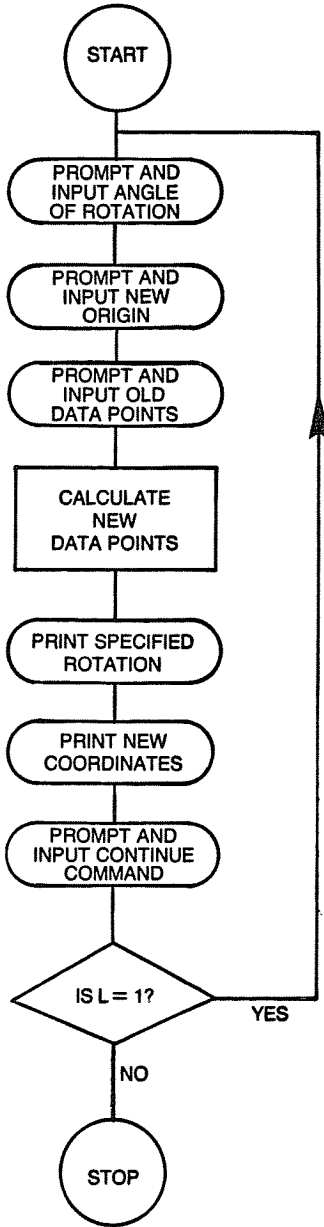
COORDINATE TRANSLATION AND/OR ROTATION

```
10    REM THIS PROGRAM COMPUTES RECTANGULAR
      COORDINATE
20    REM TRANSLATION AND/OR ROTATION
30    PRINT "ANGLE OF ROTATION IN DEGREES = ";
40    INPUT A
50    LET B = (A*3.14159)/180
60    PRINT "COORDINATES OF NEW ORIGIN (X0,Y0) =';
70    INPUT X,Y
80    PRINT "OLD COORDINATES OF DATA POINT (X1,Y1)
      =";
90    INPUT F,C
100   LET Z = F - X
110   LET W = C - Y
120   LET D = (Z*COS(B)) + (W*SIN(B))
130   LET E = - (Z*SIN(B)) + (W*COS(B))
140   PRINT
150   PRINT "SPECIFIED ROTATION =";A
160   PRINT "NEW COORDINATES OF DATA POINT"
170   PRINT "X2 = ";D
180   PRINT "Y2 = ";E
190   PRINT
200   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
210   INPUT L
220   IF L = 1 THEN 240
230   STOP
240   PRINT
250   GOTO 30
260   END
```

CRAPS

The following game-simulation is that of craps. Craps is a game that is played with two dice. The object of the game is to either win by throwing a 7 or an 11 on the first throw, or by matching your throw on the following throws. If on the first throw a 2, 3, or a 12 comes up you lose automatically; also, if you throw a 7 when looking for a match you lose too.

EXAMPLE

```
RUN
FOR RULES, TYPE RULES, OTHERWISE TYPE GO
?
GO
6
8
4
YOU THROW A 6 YOU WIN BY MATCHING
TO CONTINUE TYPE Y, IF NOT TYPE N
?
Y
FOR RULES, TYPE RULES, OTHERWISE TYPE GO
?
GO
YOU THROW A 7 YOU WIN
TO CONTINUE TYPE Y, IF NOT TYPE N
N
*END
```

CRAPS

START

PROMPT + INPUT RULES OR GO

IS R$ = "GO" — YES

NO

IS R$ = "RULES" — NO

YES

OUTPUT RULES

GENERATE D₁, D₂

IF 1st THROW CHECK FOR 7, 11, 2, 3 OR 12 — YES → 7/11 — NO → 2, 3, 12 — NO → OUTPUT THROW

NO

YES → WIN MESSAGE

YES → LOSE MESSAGE

MATCH? — YES → WIN MESSAGE

NO

7? — YES → WIN MESSAGE → PROMPT + INPUT CONTINUE COMMAND

NO

OUTPUT THROW

IS L$ = "Y" — NO → STOP
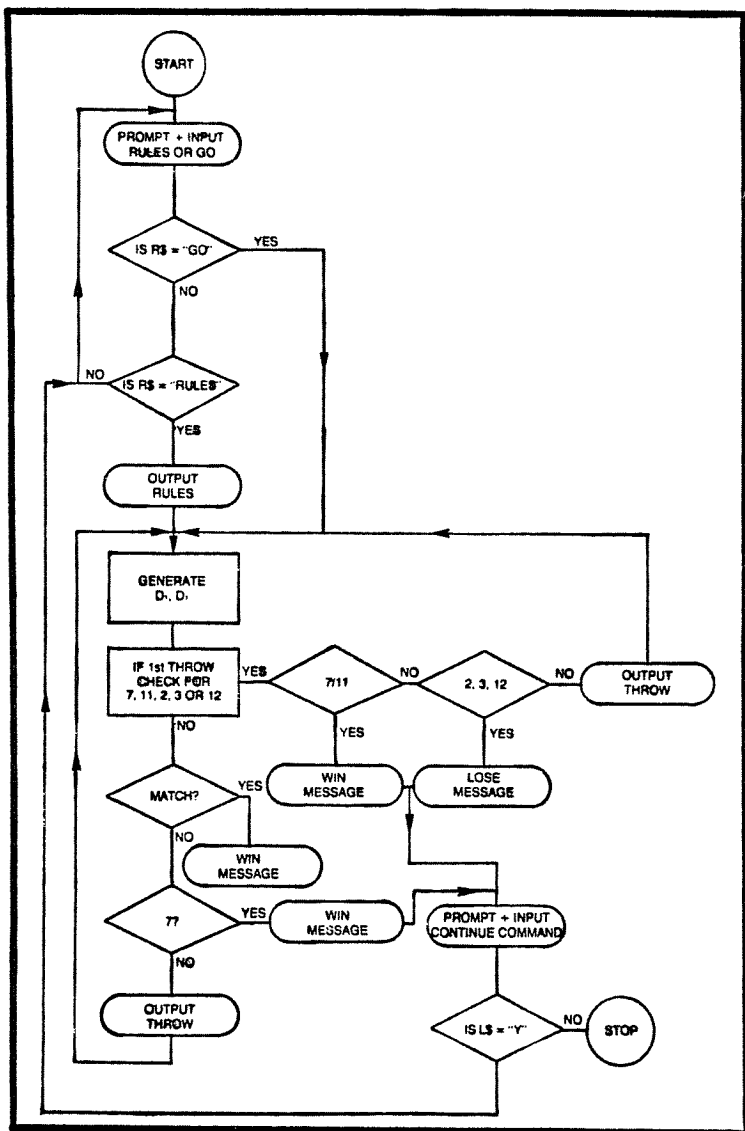
CRAPS

10	REM THIS PROGRAM SIMULATES THE GAME OF CRAPS
20	RANDOMIZE
30	PRINT " FOR RULES, TYPE RULES, OTHERWISE TYPE GO"
40	INPUT R$

```
50    IF R$ = "RULES" THEN 90
60    IF R$ = "GO" THEN 140
70    PRINT "INVALID COMMAND"
80    GOTO 30
90    PRINT "A 7 OR 11 ON THE FIRST THROW WINS"
100   PRINT "YOU CAN ALSO WIN BY THROWING A
      4,5,6,8,9,10"
110   PRINT " AND MATCHING IT BEFORE THROWING A
      7. IF ON"
120   PRINT "THE FIRST THROW A 2,3 OR A 12 COMES UP"
130   PRINT "YOU LOSE AUTOMATICALLY"
140   LET J = 0
150   GOSUB 460
160   LET D1 = N
170   GOSUB 460
180   LET D2 = N
190   LET D3 = D1 + D2
200   LET J = J + 1
210   IF J = 1 THEN 260
220   IF D3 = D4 THEN 420
230   IF D3 = 7 THEN 440
240   PRINT "YOU THROW A ";D3
250   GOTO 150
260   IF D3 = 7 THEN 320
270   IF D3 = 2 THEN 340
280   IF D3 = 3 THEN 340
290   IF D3 = 12 THEN 340
300   IF D3 = 11 THEN 320
310   LET D4 = D3
315   GOTO 240
320   PRINT "YOU THROW A ";D3;" YOU WIN"
330   GOTO 350
340   PRINT "YOU THROW A ";D3;" YOU LOSE"
350   PRINT
360   PRINT "TO CONTINUE TYPE Y, IF NOT TYPE N"
370   INPUT L$
380   IF L$ = "Y" THEN 400
390   STOP
400   PRINT
410   GOTO 30
420   PRINT "YOU THROW A ";D3;"YOU WIN BY MATCH-
      ING"
430   GOTO 350
```

```
440    PRINT "YOU THROW A ";D3;" SORRY YOU LOSE"
450    GOTO 350
460    LET N = 1 + INT(6*RND)
470    RETURN
480    END
```

CURVE TABLES (PLOTTING)

This program generates a series of data points, with which the user can plot curves. The data points generated lie between the values given for the starting value and the end point of X. Step size may be changed by using a step statement in conjunction with the FOR statement. The user must supply the equation of the curve in line 140.

EXAMPLE

(ASSUMING THE EQUATION IS (2*X) + 6)
STARTING VALUE OF X =
?
5
END VALUE OF X =
?
10
TABLE

X	Y
5	16
6	18
7	20
8	22
9	24
10	26

TYPE 1 TO CONTINUE,0 TO STOP
?
0
*END

CURVE TABLES

```
                    START

              PROMPT AND
                INPUT
              END VALUES

              PRINT TABLE

              PRINT "X", "Y"

              CALCULATE
                 X, Y

                PRINT
                 I, Y

              IS I = W        YES

                 NO

                          PRINT AND
                          INPUT CONTINUE
                          COMMAND

                          IS L = 1?     YES

                             NO

                            STOP
```

49
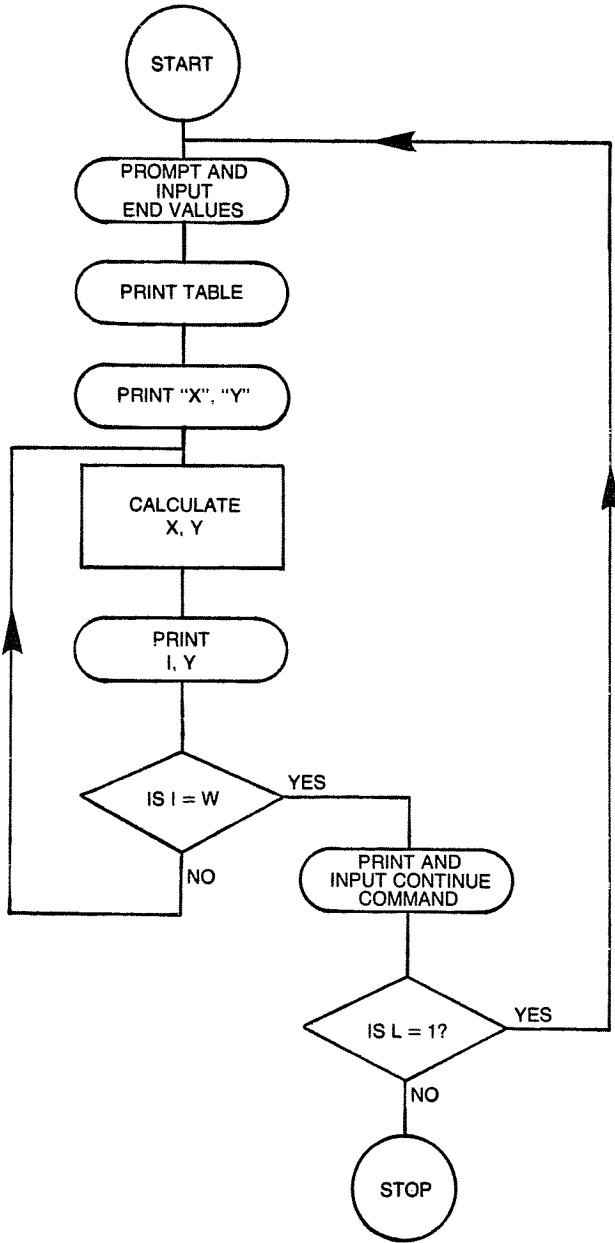
CURVE TABLES

```
10    REM THIS PROGRAM GENERATES A SERIES OF
20    REM DATA POINTS, WITH WHICH THE USER CAN
30    REM PLOT CURVES. THE DATA POINTS GENE-
      RATED
40    REM LIE BETWEEN THE VALUES GIVEN FOR X
50    REM IF A STEP SIZE OTHER THAN 1 IS DERIVED
60    REM USE A STEP STATEMENT WITH THE FOR
      STATEMENTS
70    PRINT "STARTING VALUE OF X = ";
80    INPUT X
90    PRINT "END VALUE OF X = ";
100   INPUT W
110   PRINT "TABLE"
120   PRINT "X","Y"
130   FOR I = X TO W
140   LET Y =
150   PRINT I,Y
160   NEXT I
170   PRINT
180   PRINT "TYPE 1 TO CONTINUE,0 TO STOP"
190   INPUT L
200   IF L = 1 THEN 220
210   STOP
220   PRINT
230   GOTO 70
240   END
```

DAY OF THE WEEK

This computer program computes the day of the week (e.g., Monday) from the date entered. The date entered must not be prior 1753, this is due to changes involving the switch-over from the Julian to the Gregorian Calandar.

EXAMPLE

RUN
ENTER DAY(D), MONTH(M) AND YEAR(Y)
?
16,02,1977
THE DAY OF THE WEEK IS WEDNESDAY
FOR NEXT DATE IN YES, IF NOT
TYPE NO
?
NO
DAY OF THE WEEK SAYS GOOD-BYE
*END

DAYS OF THE WEEK

START

INITIALIZE
VARIABLES

PROMPT + INPUT
D, M, Y

IS Y>1752   NO   ERROR
                     MESSAGE

YES

CALCULATE
DAY OF   OUTPUT
WEEK   RESULTS

PROMPT + INPUT
CONTINUE COMMAND

YES   IS L$ = "YES"

NO

IS L$ = "NO"   NO

YES

STOP
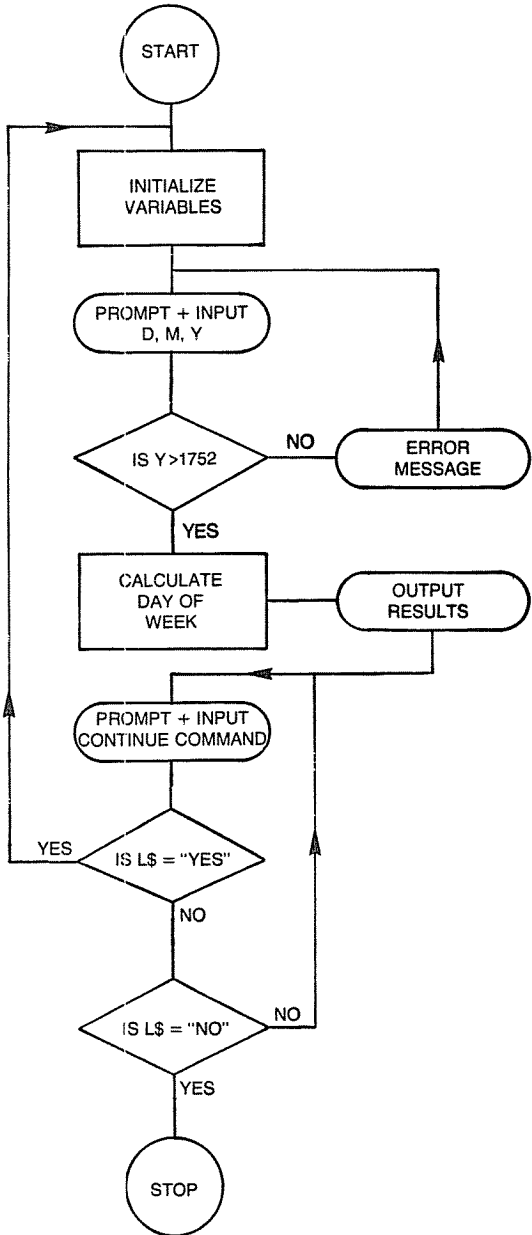
DAY OF THE WEEK

```
10    REM THIS PROGRAM COMPUTES THE DAY OF THE
      WEEK
20    REM RESTRICTION: THE DATE MUST BE AFTER
      1752
30    LET J$(1) = "SUNDAY"
40    LET J$(2) = "MONDAY"
50    LET J$(3) = "TUESDAY"
60    LET J$(4) = "WEDNESDAY"
70    LET J$(5) = "THURSDAY"
80    LET J$(6) = "FRIDAY"
90    LET J$(7) = "SATURDAY"
100   PRINT "ENTER DAY(D), MONTH(M) and YEAR(Y)"
110   INPUT D, M, Y
120   IF Y > 1752 THEN 150
130   PRINT "YEAR MUST NOT BE PRIOR TO 1753"
140   GOTO 100
150   LET K = INT(0.6 + (1/M))
160   LET L = Y - K
170   LET O = M + 12*K
180   LET P = L/100
190   LET Z1 = INT(P/4)
200   LET Z2 = INT(P)
210   LET Z3 = INT((5*L)/4)
220   LET Z4 = INT (13*(O + 1)/5)
230   LET Z = Z4 + Z3 - Z2 + Z1 + D - 1
240   LET Z=Z - (7*INT(Z/7)) + 1
250   PRINT "THE DAY OF THE WEEK IS "; J$(Z)
260   PRINT
270   PRINT "FOR NEXT DATE TYPE IN YES, IF NOT"
280   PRINT "TYPE NO"
290   INPUT L$
300   IF L$ = "YES" THEN 340
310   IF L$ = "NO" THEN 360
320   PRINT "INVALID COMMAND"
330   GOTO 270
340   PRINT
350   GOTO 30
360   PRINT "DAY OF THE WEEK SAYS GOOD-BYE"
370   END
```

DETERMINANT AND INVERSE OF A 2 × 2 MATRIX

The computer computes the inverse and determinant of a 2×2 matrix supplied by the user.

EXAMPLE

```
ENTER A11,A12
?
5,3
ENTER A21,A22
?
2,1
ORIGINAL MATRIX
5    3
2    1
INVERSE OF MATRIX
- 1    4
  2  - 5
DETERMINANT = - 1
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
```

START

PROMPT AND
INPUT MATRIX

COMPUTE
DETERMINANT
AND INVERSE

DETERMINANT AND INVERSE
OF 2 × 2 MATRIX

PRINT
ORIGINAL
MATRIX

PRINT
INVERSE OF
MATRIX

PRINT
DETERMINANT

PROMPT AND
INPUT CONTINUE
COMMAND

IS L = 1?   YES

NO

STOP
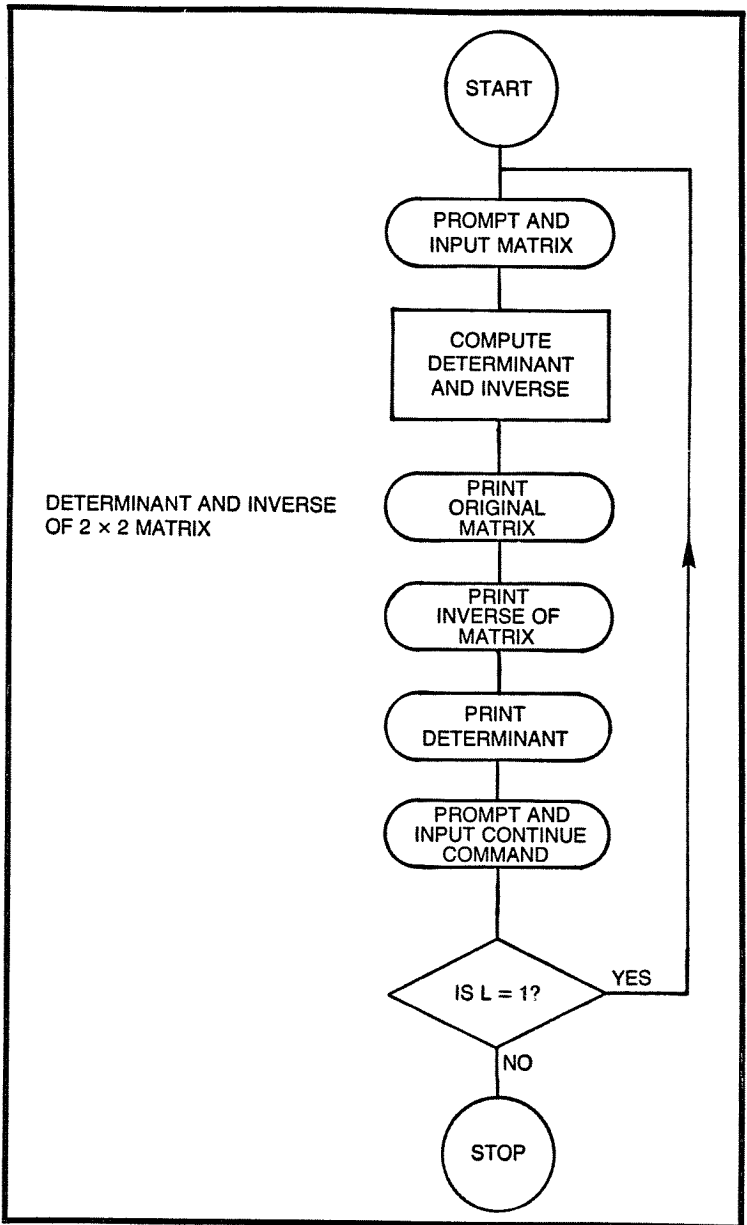
DETERMINANT AND INVERSE OF 2 × 2 MATRIX

10 REM THIS PROGRAM COMPUTES THE DETER-
    MINANT AND
20 REM INVERSE OF A 2 × 2 MATRIX

55

```
30      PRINT "ENTER A11, A12";
40      INPUT A,B
50      PRINT "A21, A22";
60      INPUT C,D
70      LET E = (D*A) - (B*C)
80      LET F = D/E
90      LET G = - B/E
100     LET H = - C/E
110     LET I = A/E
120     PRINT "ORIGINAL MATRIX"
130     PRINT A,B
140     PRINT C,D
150     PRINT
160     PRINT "INVERSE OF MATRIX"
170     PRINT F,G
180     PRINT H,I
190     PRINT
200     PRINT "DETERMINANT = ";E
210     PRINT
220     PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
230     INPUT L
240     IF L = 1 THEN 30
250     STOP
260     PRINT
270     GOTO 30
280     END
```

DETERMINATE INVERSE OF A 3 × 3 MATRIX

The user inputs his original matrix, and the computer responds with the determinate and the inverse.

FORMULAE

$$\text{MATRIX} = \begin{pmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{pmatrix}$$

$$\text{DET A} = A_1B_2C_3 + B_1C_2A_3 + C_1B_3A_2 - C_1B_2A_3 - C_2B_3A_1 - C_3A_2B_1$$

$$\text{INVERSE OF MATRIX A} = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix}$$

$\alpha_1 = (B_2C_3 - B_3C_2)/\text{DET A}$
$\alpha_2 = (A_3C_2 - A_2C_3)/\text{DET A}$
$\alpha_3 = (A_2B_3 - A_3B_2)/\text{DET A}$
$\beta_1 = (B_3C_1 - B_1C_3)/\text{DET A}$
$\beta_2 = (A_1C_3 - A_3C_1)/\text{DET A}$
$\beta_3 = (A_3B_1 - A_1B_3)/\text{DET A}$
$\gamma_1 = (B_1C_2 - B_2C_1)/\text{DET A}$
$\gamma_2 = (A_2C_1 - A_1C_2)/\text{DET A}$
$\gamma_3 = (A_1B_2 - A_2B_1)/\text{DET A}$

EXAMPLE

ENTER 3X3 MATRIX
?
1,4,2,2,4,2,3,5,1
ORIGINAL MATRIX
1 4 2
2 4 2
3 5 1
DETERMINANT = 6
INVERSE OF MATRIX
− 1 1 0
.67 − .83 .33
− .33 1.17 − .67
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

DETERMINANT AND INVERSE OF 3 × 3 M

```
                    ╭─────────╮
                    │  START  │
                    ╰─────────╯
                         │
              ╭──────────────────────╮
              │  PROMPT AND           │
              │  INPUT 3 × 3          │
              │  MATRIX               │
              ╰──────────────────────╯
                         │
              ┌──────────────────────┐
              │  ·CALCULATE           │
              │  DETERMINANT          │
              │  AND INVERSE          │
              └──────────────────────┘
                         │
              ╭──────────────────────╮
              │  PRINT                │
              │  ORIGINAL             │
              │  MATRIX               │
              ╰──────────────────────╯
                         │
              ╭──────────────────────╮
              │  PRINT                │
              │  DETERMINANT          │
              ╰──────────────────────╯
                         │
              ╭──────────────────────╮
              │  PRINT INVERSE        │
              │  OF                   │
              │  MATRIX               │
              ╰──────────────────────╯
                         │
              ╭──────────────────────╮
              │  PROMPT AND           │
              │  INPUT CONTINUE       │
              │  COMMAND              │
              ╰──────────────────────╯
                         │
                    ╱─────────╲
                   ╱           ╲    YES
                  ◇   IS L = 1? ◇────────▶
                   ╲           ╱
                    ╲─────────╱
                         │ NO
                    ╭─────────╮
                    │  STOP   │
                    ╰─────────╯
```
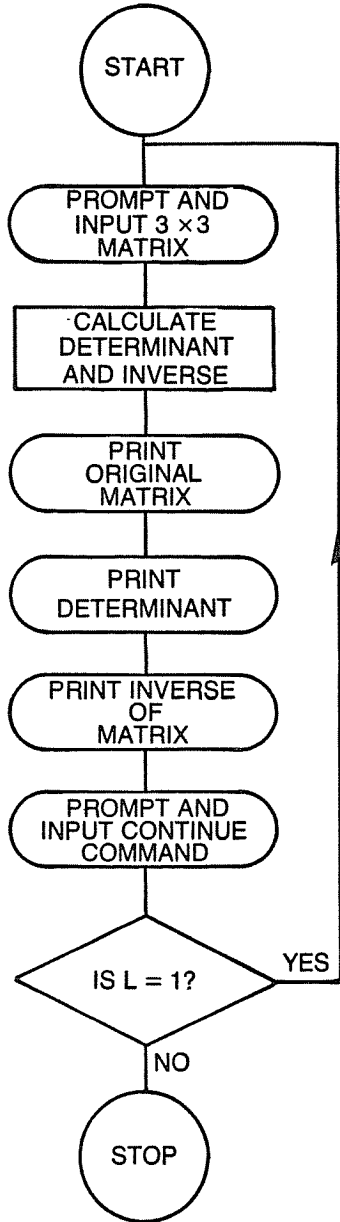
DETERMINANT AND INVERSE OF 3 × 3 MATRIX

```
10    REM THIS PROGRAM COMPUTES 3 × 3 MATRIX
      OPERATIONS
20    PRINT "ENTER 3 × 3 MATRIX"
30    INPUT A,B,C,D,E,F,G,H,I
40    REM COMPUTE DETERMINANT OF 3*3
50    LET M = (A*E*I) + (B*F*G) + (C*H*D)
60    LET N = M − (C*E*G) − (F*H*A) − (I*D*B)
70    REM COMPUTE INVERSE
80    LET O = ((E*I) − (H*F))/N
90    LET P = ((G*F) − (D*I))/N
100   LET Q = ((D*H) − (G*E))/N
110   LET R = ((H*C) − (B*I))/N
120   LET S = ((A*I) − (G*C))/N
130   LET T = ((G*B) − (A*H))/N
140   LET U = ((B*F) − (E*C))/N
150   LET V = ((D*C) − (A*F))/N
160   LET W = ((A*E) − (D*B))/N
170   PRINT "ORIGINAL MATRIX"
180   PRINT A,B,C
190   PRINT D,E,F
200   PRINT G,H,I
210   PRINT
220   PRINT "DETERMINANT = ";N
230   PRINT
240   PRINT "INVERSE OF MATRIX"
250   PRINT O,R,U
260   PRINT P,S,V
270   PRINT Q,T,W
280   PRINT
290   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
300   INPUT L
310   IF L = 1 THEN 20
320   STOP
330   END
```

FACTORIALS

This program computes by iterative multiplication of the factorial of X.

FORMULA

$$(X)(X - 1)(X - 2)... (X - X + 1)$$

EXAMPLE

X =
?
5
X = 5 X! = 120
TYPE 1 TO CONTINUE,0 TO STOP
?
1
X =
?
7
X = 7 X! = 5040
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
X =
?
18
X = 18 X! = 6.40237E15
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

FACTORIALS

START

PROMPT AND INPUT X

CALCULATE $(X_i) \times (X_{i-1})$

IS I = X?

YES

NO

PRINT X AND X!

PROMPT AND INPUT CONTINUE COMMAND

IS L = 1?

STOP

IS L = 1?
YES NO STOP
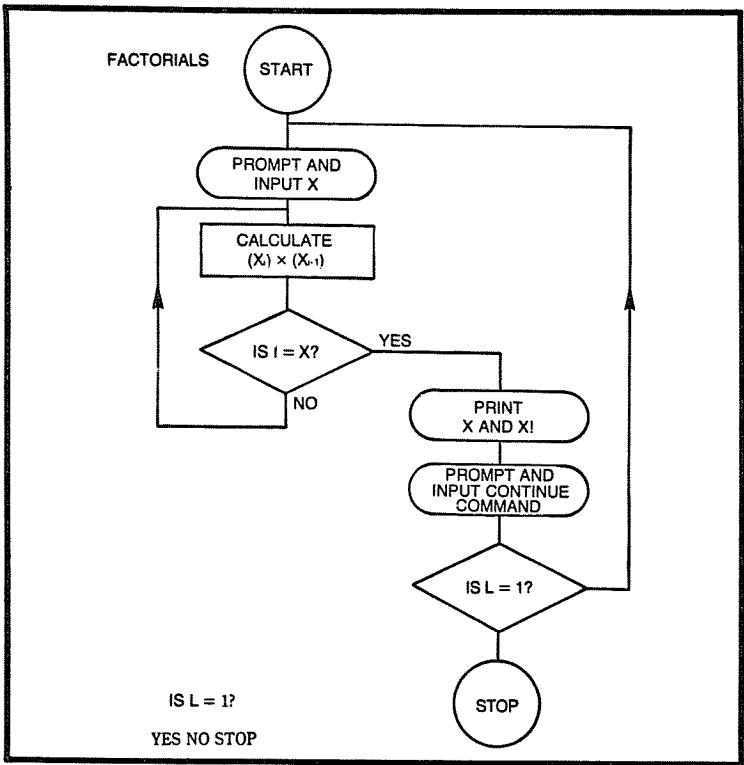
FACTORIALS

```
10   REM THIS PROGRAM COMPUTES THE FACTORIALS
        OF X
20   REM BY ITERATIVE MULTIPLICATIONS
30   PRINT "X = ";
40   INPUT X
50   LET Z = 1
60   FOR I = 1 TO X
70   LET Z = Z*I
80   NEXT I
90   PRINT "X = ";X, "X! = ";Z
100  PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
110  INPUT L
120  IF L = 1 THEN 140
130  STOP
140  PRINT
150  GOTO 30
160  END
```

FIBONACCI NUMBERS

This program computes a table of Fibonacci numbers from the first two terms entered by the user, who may also specify the maximum number of terms.

FORMULA

$F_1 = t^{th}$ *term in the sequence;*   *For any two terms the*

first term $= f_i - 2,$ *second term* $= f_i - 1$

$$f_i = f_{i-1} + f_{i-2}$$

EXAMPLE

ENTER 1ST TERM
?
1
ENTER 2ND TERM
?
1
MAXIMUM NUMBER OF TERMS =
?
10
TABLE OF FIBONACCI NUMBERS

TERM NO.	FIBONACCI NUMBERS
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34
10	55

MAXIMUM NUMBER OF TERMS REACHED
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
ENTER FIRST TERM
?
27

ENTER SECOND TERM
?
963
MAXIMUM NUMBER OF TERMS =
?
5
TABLE OF FIBONACCI NUMBERS

TERM NO.	FIBONACCI NUMBERS
1	27
2	963
3	990
4	1953
5	2943

MAXIMUM NUMBER OF TERMS REACHED
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

FIBONACCI NUMBERS

```
                    ( START )
                        |
         ( PROMPT AND INPUT VARIABLES )
                        |
         ( PRINT TABLE OF NUMBERS )
                        |
         ( PRINT "TERM NO." AND "FIBONACCI NO." )
                        |
              ( PRINT K, A )
                        |
              ( PRINT K, B )
                        |
            [ CALCULATE NEXT NUMBER ]
                        |
              ( PRINT K, Q. )
                        |
               < IS K ≥ N? >  --YES-->  ( PRINT "MAX TERMS REACHED" )
                   |                               |
                  NO                    ( PROMPT AND INPUT CONTINUE COMMAND )
                                                   |
                                          < IS L = 1? > --YES-->
                                                   |
                                                  NO
                                                   |
                                               ( STOP )
```
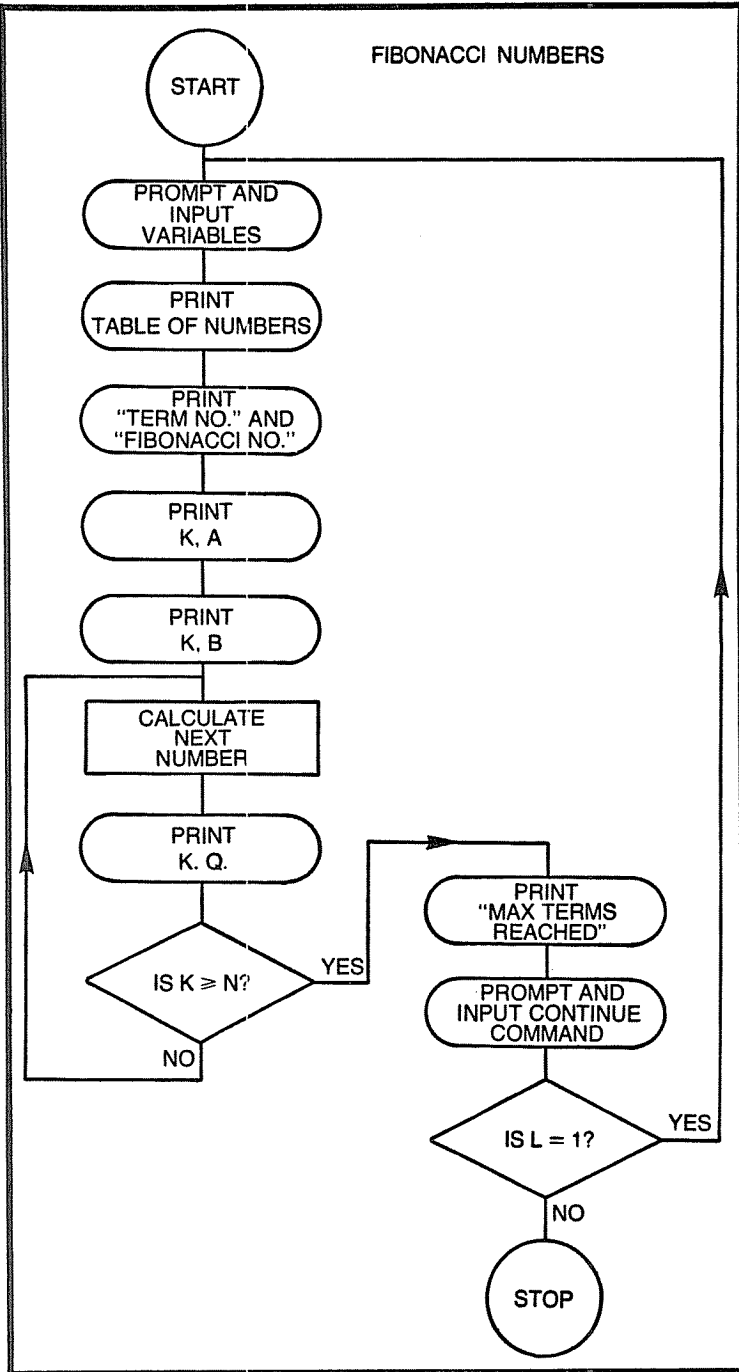
FIBONACCI NUMBERS

```
10   REM THIS PROGRAM COMPUTES A TABLE OF
     FIBONACCI NUMBERS
20   PRINT "ENTER FIRST TERM"
30   INPUT A
40   PRINT "ENTER SECOND TERM"
50   INPUT B
60   PRINT "MAXIMUM NUMBER OF TERMS = "
70   INPUT N
80   PRINT
90   PRINT "TABLE OF FIBONACCI NUMBERS"
100  PRINT "TERM NO.","FIBONACCI NUMBER"
110  LET K = 1
120  PRINT K,A
130  LET K = 2
140  PRINT K,B
150  LET K = K + 1
160  LET Q = A + B
170  PRINT K,Q
180  LET A = B
190  LET B = Q
200  IF K > = N THEN 220
210  GOTO 150
220  PRINT "MAXIMUM NUMBER OF TERMS REACHED"
230  PRINT
240  PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
250  INPUT L
260  IF L = 1 THEN 280
270  STOP
280  PRINT
290  GOTO 20
300  END
```

FIRST DERIVATIVE

The user must supply the expression to complete the assignment statement in line 280. The computer will then derive the first derivative of the supplied equation.

FORMULA

$$x > 0$$

$$f^1(x) = \frac{f(x + \Delta x/2) - f(x - \Delta x/2)}{\Delta x}$$

EXAMPLE

(ASSUMING THE EQUATION IS X↑2)
VALUE OF X =
?
10
IF X = 10 THEN F(X) = 100
AND F'(X) = 20
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
VALUE OF X =
?
20
IF X = 20 THEN F(X) = 400
AND F'(X) = 40
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
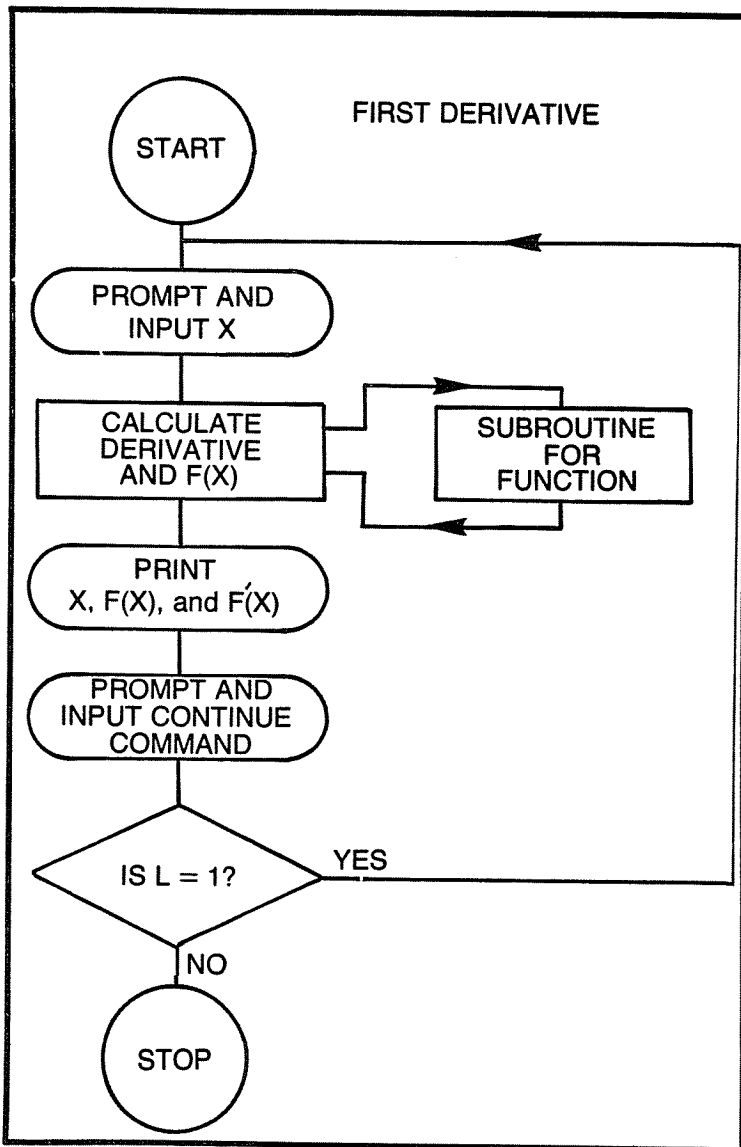
FIRST DERIVATIVE

FIRST DERIVATIVE

```
10   REM THIS PROGRAM COMPUTES THE FIRST DE-
     RIVATIVE
20   REM OF A FUNCTION ENTERED BY THE USER
30   PRINT "VALUE OF X = ";
40   INPUT Y
```

```
50    LET Y = X
60    LET Z = (X*(1E − 04))/2
70    LET W = X + Z
80    LET V = X − Z
90    LET X = W
100   GOSUB 280
110   LET A = P
120   LET X =V
130   GOSUB 280
140   LET B =P
150   LET X = Y
160   GOSUB 280
170   LET C = P
180   LET F = (A − B)/2*Z
190   PRINT "IF X = ";Y,"THEN F(X) = ";C
200   PRINT "AND F' (X) = ";F
210   PRINT
220   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
230   INPUT L
240   IF L = 1 THEN 260
250   STOP
260   PRINT
270   GOTO 40
280   LET P =
290   RETURN
300   END
```

GAMMA FUNCTION AND GENERALIZED FACTORIAL

This program computes both the gamma function and the generalized factorial via polynomial approximation.

FORMULAE

$$\Gamma(X) = \int_{0}^{\infty} t^{x-1} e^{-t} dt$$

$$\Gamma(X) \cong (1 + A_1(Y) + A_2(Y)^2 \ldots \ldots A_8(Y)^8)Z$$

where Y = fractional part of X
where $Z = \_ (X - 1)(X - 2)\ldots\ldots(X - N)$
and $X - N = 1 + Y$

EXAMPLE

RUN
TYPE G FOR THE GAMMA FUNCTION OR
TYPE F FOR THE GENERALIZED FACTORIAL
?
G
ENTER VALUE OF X
?
5
GAMMA (5) = 24
TO CONTINUE TYPE Y, IF NOT TYPE N
?
Y
TYPE G FOR THE GAMMA FUNCTION OR
TYPE F FOR THE GENERALIZED FACTORIAL
?
F
ENTER VALUE OF X
?
5
5! = 120
TO CONTINUE TYPE Y, IF NOT TYPE N
?
N
*END

GAMMA FUNCTION AND GENERALIZED FACTORIAL

START

INITIALIZE
VARIABLES

PROMPT INPUT
G OR F

IS A$ = "G"

CALCULATE
Γ(X)

FACTORIAL
SUBROUTINE

POLYNOMIAL
SUBROUTINE

A

IS A$ = "F"

CALCULATE
X!

A   B   B

OUTPUT
Γ(X)

OUTPUT
X!

PROMPT + INPUT
CONTINUE COMMAND

IS L$ = "Y"

STOP
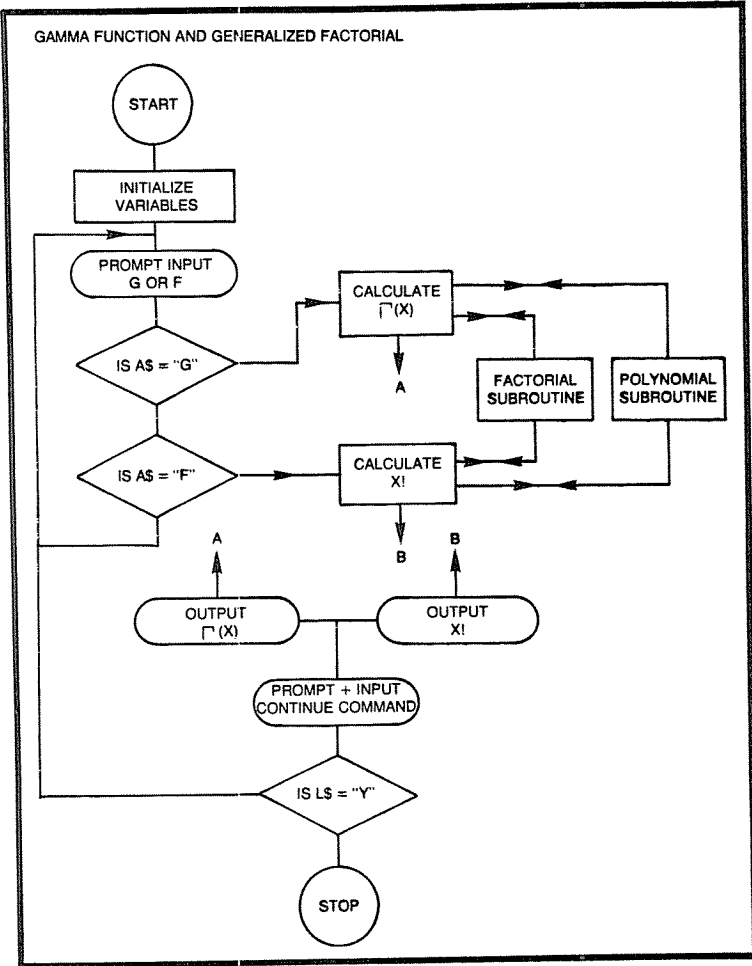
GAMMA FUNCTION AND GENERALIZED FACTORIAL

```
10    REM THIS PROGRAM GENERATES VIA POLY-
      NOMIAL
20    REM APPROXIMATION THE GAMMA FUNCTION
30    REM AND THE GENERALIZED FACTORIALS
40    LET A = 0.57717
50    LET B = 0.98821
60    LET C = 0.89706
70    LET D = 0.91821
80    LET E = 0.7567
90    LET F = 0.4822
100   LET G = 0.19353
```

```
110    LET H = 0.03587
120    PRINT "TYPE G FOR THE GAMMA FUNCTION OR"
130    PRINT "TYPE F FOR THE GENERALIZED FAC-
       TORIAL"
140    INPUT A$
150    IF A$ = "G" THEN 190
160    IF A$ = "F" THEN 300
170    PRINT "INVALID RESPONSE"
180    GOTO 120
190    PRINT "ENTER VALUE OF X"
200    INPUT X
210    LET K = X
220    LET K = K - 1
230    IF K > = 0 THEN 260
240    PRINT "X MUST BE EQUAL TO OR GREATER
       THAN 1"
250    GOTO 190
260    GOSUB 490
270    IF (X - 1) = INT(X - 1) THEN 410
280    GOSUB 570
290    GOTO 410
300    PRINT "ENTER VALUE OF X"
310    INPUT X
320    LET K = X
330    IF K > = 0 THEN 360
340    PRINT "X MUST BE GREATER THAN OR EQUAL
       TO 0"
350    GOTO 300
360    GOSUB 490
370    IF X = INT(X) THEN 390
380    GOSUB 570
390    PRINT X;"! = ";K
400    GOTO 420
410    PRINT "GAMMA (";X;") = ";K
420    PRINT
430    PRINT "TO CONTINUE TYPE Y, IF NOT TYPE N"
440    INPUT L$
450    IF L$ = "Y" THEN 470
460    STOP
470    PRINT
480    GOTO 120
490    LET J = 1
500    LET J = J*K
```

```
510    LET K = K − 1
520    IF K < 1 THEN 540
530    GOTO 500
540    LET L = K
550    LET K = J
560    RETURN
570    LET A1 = 1 + (A*L) + (B*(L↑2)) + (C*(L↑3))
580    LET A1 = A1 + (D*(L↑4)) + (E*(L↑5)) + (F*(L↑6))
590    LET A1 = A1 + (G*(L↑7)) + (H*(L↑8))
600    LET K = A1*K
610    RETURN
620    END
```

GAUSSIAN PROBABILITY FUNCTION

This program computes the Gaussian probability function of X.

FORMULA

$$f(X) = \frac{1}{\sqrt{2\pi}} \; e- \frac{X^2}{2}$$

EXAMPLES

X =
?
3.2
F(X) = .00238
TO CONTINUE TYPE 1, 0 TO STOP
?
1
X =
?
4
F(X) = .000133
TO CONTINUE TYPE 1, 0 TO STOP
?
1
X =
?
1.2
F(X) = .19418
TO CONTINUE TYPE 1, 0 TO STOP
?
0
*END

GAUSSIAN PROBABILITY FUNCTION

Flowchart:
- START
- GAUSSIAN PROBABILITY FUNCTION
- PROMPT + INPUT X
- CALCULATE f (X)
- OUTPUT f (X)
- PROMPT + INPUT CONTINUE COMMAND
- IS L = 1? — YES / NO
- STOP
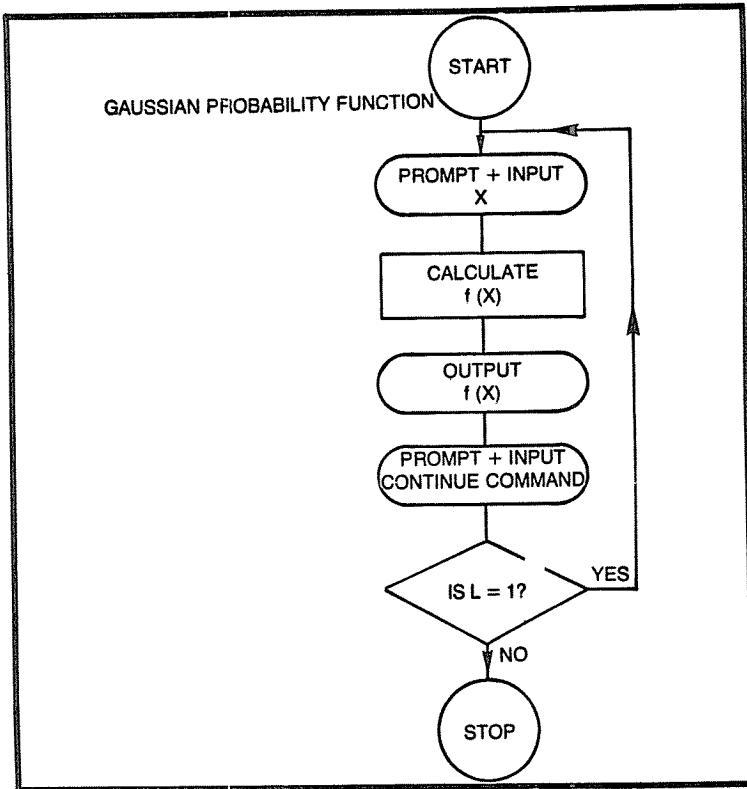
GAUSSIAN PROBABILITY FUNCTION

```
10    REM THIS PROGRAM COMPUTES THE GAUSSIAN
          PROBABILITY
20    REM FUNCTION OF X
30    PRINT "X = ";
40    INPUT X
50    LET A = EXP( – (X↑2)/2)
60    LET B = .398942
70    LET C = B*A
80    PRINT "F(X) = ";C
90    PRINT
100   PRINT "TO CONTINUE TYPE 1, 0 TO STOP"
110   INPUT L
120   IF L = 1 THEN 140
130   STOP
140   PRINT
150   GOTO 30
160   END
```

GAUSSIAN QUADRATURE

This program computes the integral f(x)dx for a finite A by the 6-point Gaussian-Legendre quadrature formula. It should be noted that f(x) must be a single-valued function.

FORMULA

$$\int_A^\infty f(x)dx = \frac{1}{2} \sum_{i=}^{6} \frac{4D_i}{(1+C_i)^2} f\left(\frac{2}{1+C_i}+A-1\right)$$

EXAMPLE

ENDPOINT A =
?
0
INTEGRAL = 0.92
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

The LET Q statement in line 470 must be completed by the user. Any single-valued expression may be used for f(x).

75
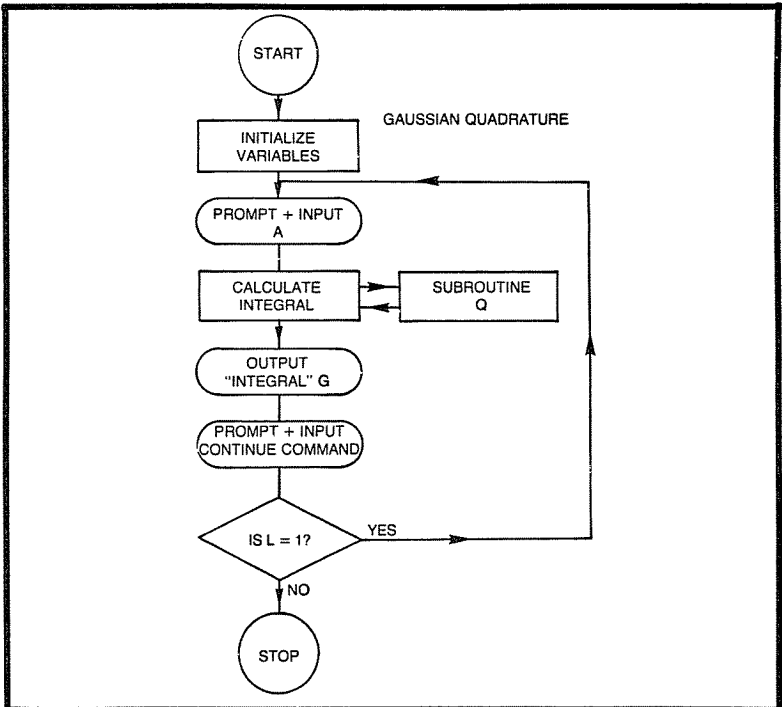
GAUSSIAN QUADRATURE

```
10    REM THIS PROGRAM COMPUTES THE INTEGRAL
      BETWEEN A
20    REM AND INFINITY BY GAUSSIAN QUADRATURE
30    LET C1 = .238619
40    LET C2 = - C1
50    LET C3 = .661209
60    LET C4 = - C3
70    LET C5 = .932470
80    LET C6 = - C5
90    LET D1 = .467914
100   LET D2 = .360762
110   LET D3 = .171324
120   LET J = 0
130   PRINT "ENDPOINT A = ";
140   INPUT A
150   LET X = (2/(1 + C1)) + (A - 1)
160   GOSUB 470
170   LET N = ((4*D1)/(1 + C1)↑2)*Q
180   LET J = J + N
190   LET X = (2/(1 + C2)) + (A - 1)
200   GOSUB 470
210   LET N = ((4*D1)/(1 + C2)↑2)*Q
220   LET J = J + N
230   LET X = (2/(1 + C3)) + (A - 1)
240   GOSUB 470
250   LET N = ((4*D2)/(1 + C3)↑2)*Q
260   LET J = J + N
270   LET X = (2/(1 + C4)) + (A - 1)
280   GOSUB 470
290   LET N = ((4*D2)/(1 + C4)↑2)*Q
300   LET J = J + N
310   LET X = (2/(1 + C5)) + (A - 1)
320   GOSUB 470
330   LET N = ((4*D3)/(1 + C5)↑2)*Q
340   LET J = J + N
350   LET X = (2/(1 + C6)) + (A - 1)
360   GOSUB 470
370   LET N = ((4*D3)/(1 + C6)↑2)*2
380   LET J = J + N
390   LET G = J/2
400   PRINT "INTEGRAL = ";G
```

```
410    PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
420    INPUT L
430    IF L = 1 THEN 450
440    STOP
450    PRINT
460    GOTO 120
470    LET Q =
480    RETURN
490    END
```

GENERALIZED MEAN

This program computes the generalized mean, which becomes equal to the arithmetic mean if the T entered by the user is 1, and equal to the harmonic mean if the user enters − 1.

FORMULA

$$M(T) = \left(\frac{1}{N} \sum_{K=1}^{N} x_K^T \right)^{\frac{1}{T}} \quad \text{where } X > 0$$

EXAMPLE

T =
?
4
INPUT SAMPLE
?
5
GENERALIZED MEAN
M(T) = 5 WHERE T = 4
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

GENERALIZED MEAN

START

INITIALIZE
VARIABLES

PROMPT + INPUT
T

PROMPT + INPUT
SAMPLES

IS X = 0?

YES

NO

UPDATE J
AND N

COMPUTE
MEAN

OUTPUT
M(T) AND T

PROMPT + INPUT
CONTINUE COMMAND

IS L = 1

YES

NO

STOP

79
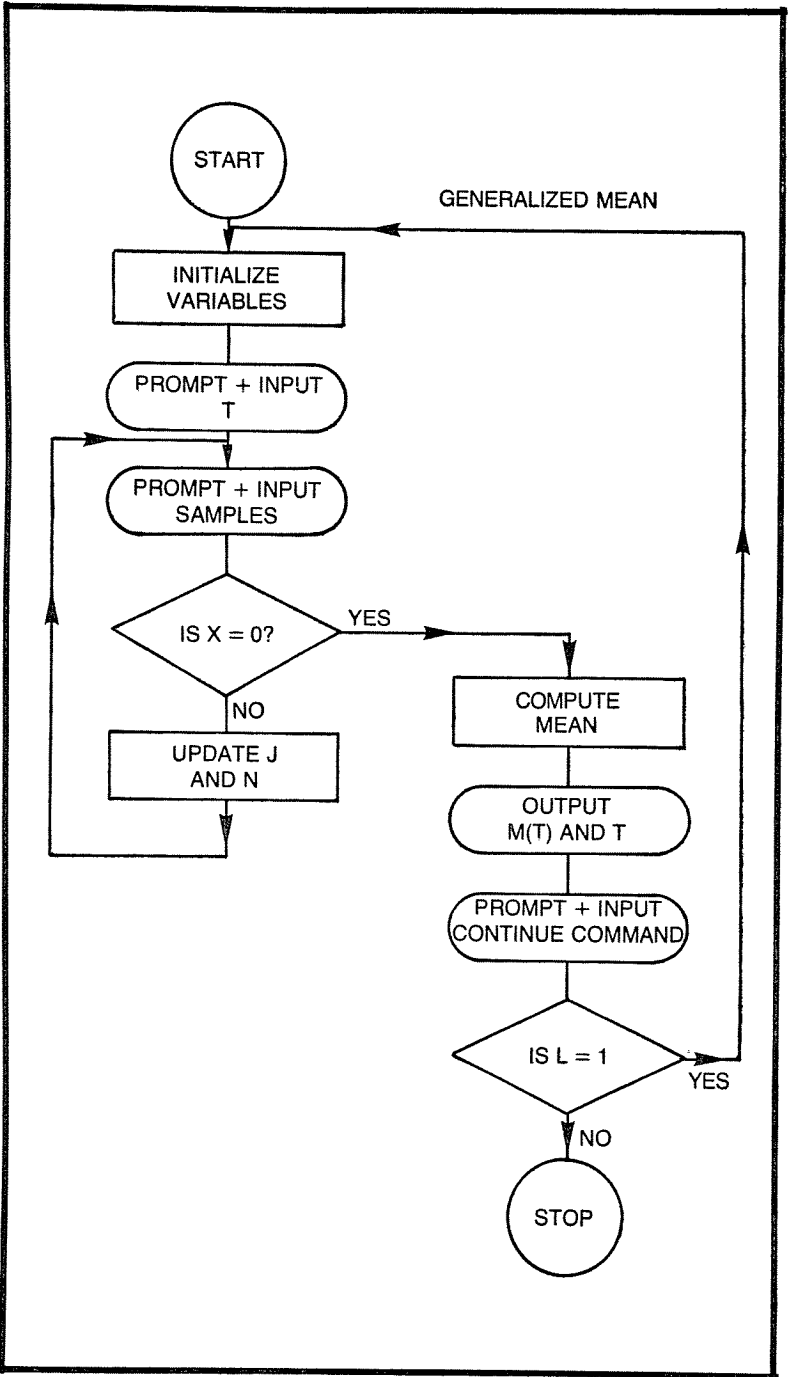
GENERALIZED MEAN

```
10    REM THIS PROGRAM COMPUTES THE GENER-
      ALIZED MEAN
20    REM IF T = 1, THEN THE GENERALIZED MEAN
      M(T)
30    REM IS EQUAL TO THE ARITHMETIC MEAN. IF
      T = - 1
40    REM THEN M(T) IS EQUAL TO THE HARMONIC
      MEAN
50    LET J = 0
60    LET N = 0
70    PRINT "T = ";
80    INPUT T
90    PRINT "INPUT SAMPLE";
100   INPUT X
110   IF X = 0 THEN 150
120   LET J = J + (X↑T)
130   LET N = N + 1
140   GOTO 90
150   LET G = (J/N)↑(1/T)
160   PRINT "GENERALIZED MEAN"
170   PRINT "M(T) = ";G, "WHERE T = ";T
180   PRINT
190   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
200   INPUT L
210   IF L = 1 THEN 230
220   STOP
230   PRINT
240   GOTO 50
250   END
```

GEOMETRIC MEAN

This program computes the geometric mean ot the sample entered by the user until a 0 is entered for the sample.

FORMULA

$$G = \sqrt[n]{(a_1)(a_2)\ldots\ldots(a_n)}$$

EXAMPLE

SAMPLE =
?
2
NUMBER OF SAMPLES = 1 CURRENT SAMPLE = 2
CURRENT MEAN = 2
SAMPLE =
?
5
NUMBER OF SAMPLES = 2 CURRENT SAMPLE = 5
CURRENT MEAN = 3.1623
SAMPLE =
?
0
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
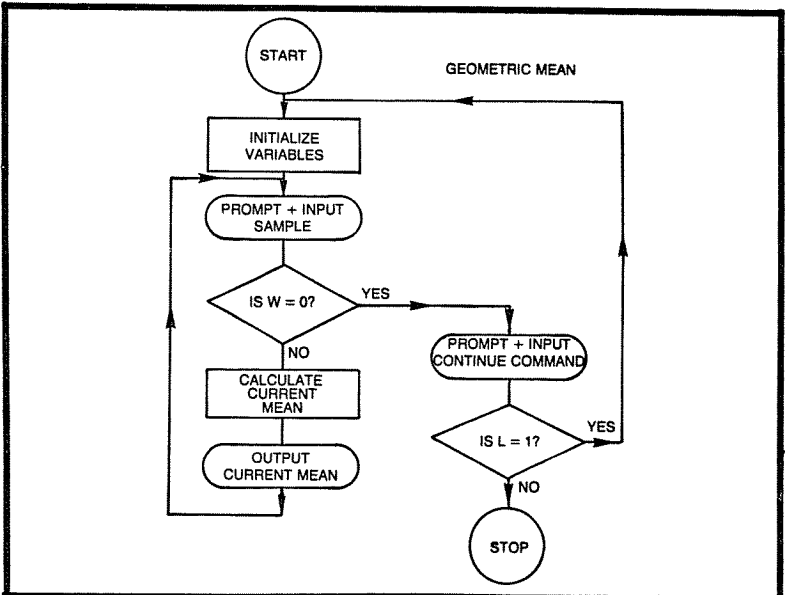
GEOMETRIC MEAN

GEOMETRIC MEAN

```
10    REM THIS PROGRAM COMPUTES THE GEOMETRIC
      MEAN
20    REM AFTER EACH SAMPLE IS ENTERED, THE
      NUMBER OF
30    REM SAMPLES, THE CURRENT SAMPLE AND CUR-
      RENT MEAN
40    REM IS PRINTED.
50    LET Y = 1
60    LET N = 0
70    PRINT "SAMPLE = ";
80    INPUT W
90    IF W = 0 THEN 160
100   LET N = N + 1
110   LET Y = Y*W
120   LET G = Y^(1/N)
130   PRINT "NUMBER OF SAMPLES = ";N,"CURRENT
      SAMPLE = ";W
140   PRINT "CURRENT MEAN = ";G
150   GOTO 70
160   PRINT
170   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
180   INPUT L
190   IF L = 1 THEN 210
200   STOP
210   PRINT
220   GOTO 50
230   END
```

GEOMETRIC PROGRESSION

From the following information: first term, ratio of terms, anu number of terms, this program computes the geometric progression.

FORMULA

$$A, AR, AR^2..., AR^{N-1}$$

EXAMPLE

FIRST TERM =
?
10
RATIO OF TERMS =
?
2
NUMBER OF TERMS =
?
5
FOR TABLE, TYPE 1, IF NOT TYPE 0
?
1
GEOMETRIC PROGRESSION

TERM NUMBER	TERM VALUE
1	10
2	20
3	40
4	80
5	160

SUM = 310
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

GEOMETRIC PROGRESSION

START

PROMPT + INPUT
VARIABLES

PROMPT + INPUT
TABLE COMMAND

IS C = 1?
YES → PRINT
TABLE HEADER
NO

COMPUTER
PROGRESSION

IS C = 1
YES → OUTPUT
K, L
NO

IS I = N – 1?
YES → OUTPUT
SUM
NO

PROMPT + INPUT
CONTINUE COMMAND

IS X = 1?

STOP
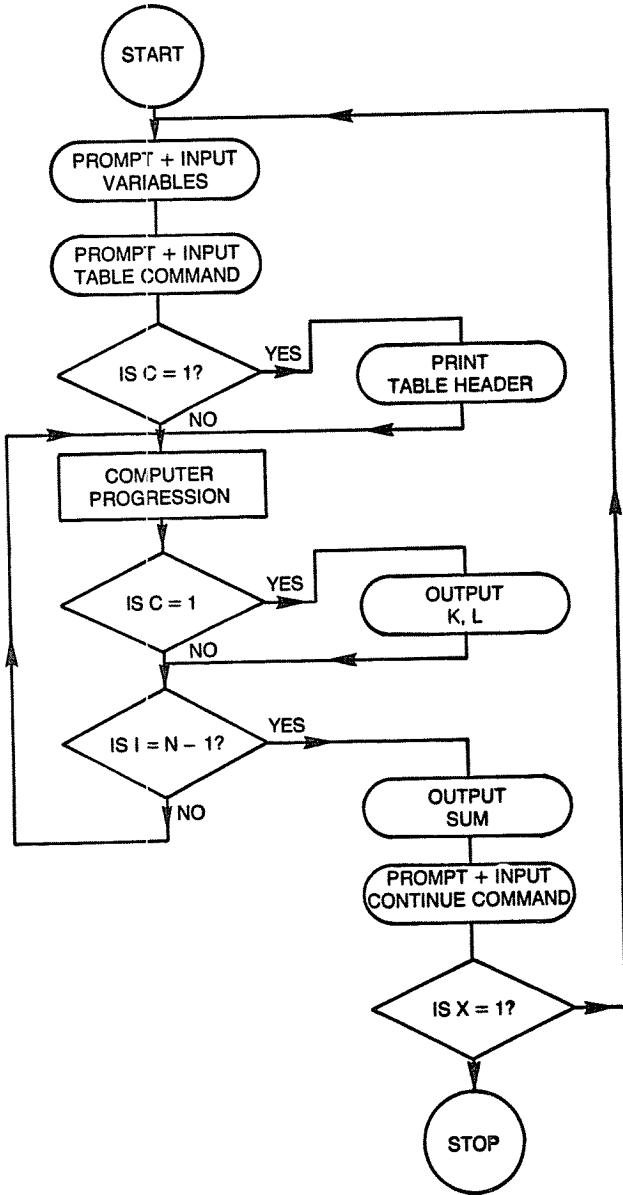
GEOMETRIC PROGRESSION

```
10    REM THIS PROGRAM COMPUTES THE VALUES AND
          THEIR SUM
20    REM OF A GEOMETRIC PROGRESSION
30    PRINT "FIRST TERM = ";
40    INPUT A
50    PRINT "RATIO OF TERMS = ";
60    INPUT R
70    PRINT "NUMBER OF TERMS = ";
80    INPUT N
90    PRINT "FOR TABLE TYPE 1, IF NOT TYPE 0"
100   INPUT C
110   IF C = 1 THEN 130
120   GOTO 160
130   PRINT
140   PRINT "GEOMETRIC PROGRESSION"
150   PRINT "TERM NUMBER";,"TERM VALUE";
160   LET J = 0
170   FOR I = 0 TO N − 1
180   LET K = I + 1
190   LET L = A*(R↑I)
200   LET J = J + L
210   IF C = 1 THEN 230
220   GOTO 240
230   PRINT K,L
240   NEXT I
250   PRINT "SUM = ";J
260   PRINT
270   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
280   INPUT X
290   IF X = 1 THEN 310
300   STOP
310   PRINT
320   GOTO 30
330   END
```

HARMONIC MEAN

This program computes the harmonic mean of the samples entered
by the user, until a 0 is entered for the sample.

FORMULA

$$H = \frac{N}{\sum\limits_{i=1}^{N} \dfrac{1}{a_i}}$$

EXAMPLE

SAMPLE =
?
2
N = 1 SAMPLE = 2 CURRENT MEAN = 2
SAMPLE =
?
5
N = 2 SAMPLE = 5 CURRENT MEAN = 2.86
SAMPLE =
?
7
N = 3 SAMPLE = 7 CURRENT MEAN = 3.56
SAMPLE =
?
0
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
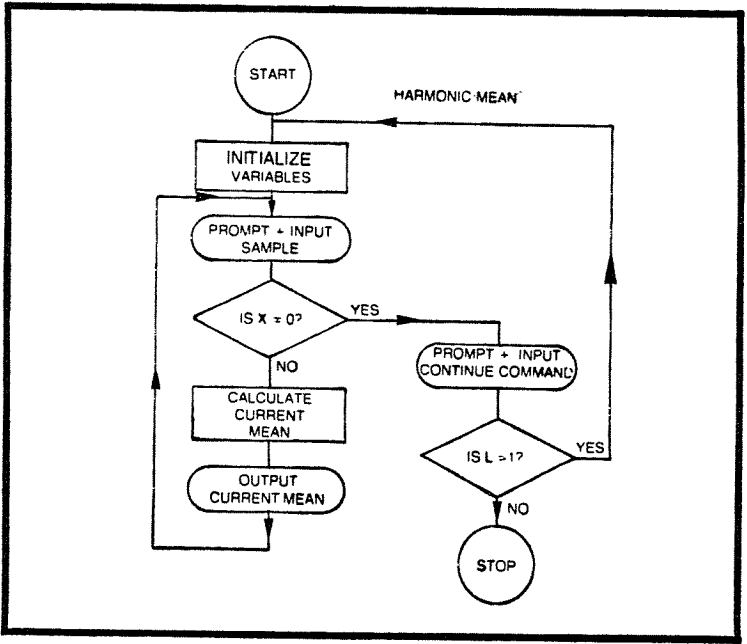
HARMONIC MEAN

```
10    REM THIS PROGRAM COMPUTES THE HARMONIC
      MEAN
20    REM OF THE SAMPLES ENTERED BY THE USER
30    LET Z = 0
40    LET N = 0
50    PRINT "SAMPLE = ";
60    INPUT X
70    IF X = 0 THEN 130
80    LET N = N + 1
90    LET Z = Z + (1/X)
100   LET H = N/Z
110   PRINT "N = ";N,"SAMPLE = ";X,"CURRENT MEAN
      = ";H
120   GOTO 50
130   PRINT
140   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
150   INPUT L
160   IF L = 1 THEN 180
170   STOP
180   PRINT
190   GOTO 30
200   END
```

HARMONIC NUMBERS

This program computes the first N harmonic numbers where N is entered by the user.

FORMULA

1, 1 + 1/2, 1 + 1/2 + 1/3, 1 + 1/2 + 1/3 + 1/4, ...

EXAMPLE

MAXIMUM TERM NUMBER
?
35

TERM NUMBER	TERM VALUE
1	1
2	1.5
3	1.83333
4	2.08333
5	2.28333
6	2.45
7	2.59285
8	2.71785
9	2.82896
10	2.92896
11	3.01987
12	3.10321
13	3.18013
14	3.25156
15	3.31822
16	3.38072
17	3.43955
18	3.49510
19	3.54774
20	3.59773
21	3.64535
22	3.69081
23	3.73429
24	3.77595
25	3.81595
26	3.85442
27	3.89145
28	3.92717
29	3.96165
30	3.99498

TERM NUMBER	TERM VALUE
31	4.02724
32	4.05849
33	4.08879
34	4.11821
35	4.14678
36	4.17456

*END

```
                    ┌─────────┐
                    │  START  │
                    └─────────┘
                         │
              ┌──────────────────────┐
              │   PROMPT + INPUT     │
              │     MAX TERM#        │
              └──────────────────────┘
                         │
              ┌──────────────────────┐
              │     INITIALIZE       │
              │     VARIABLES        │
              └──────────────────────┘
                         │
              ┌──────────────────────┐
              │  PRINT TERM # +      │     HARMONIC NUMBERS
              │   TERM VALUE         │
              └──────────────────────┘
                         │
              ┌──────────────────────┐
              │      COMPUTE         │
              │     HARMONIC         │
              │      NUMBERS         │
              └──────────────────────┘
                         │
              ┌──────────────────────┐
              │       OUTPUT         │
              │        K, D          │
              └──────────────────────┘
                         │
              NO      ◇ IS I = N? ◇
                         │
                        YES
                    ┌─────────┐
                    │  STOP   │
                    └─────────┘
```

89
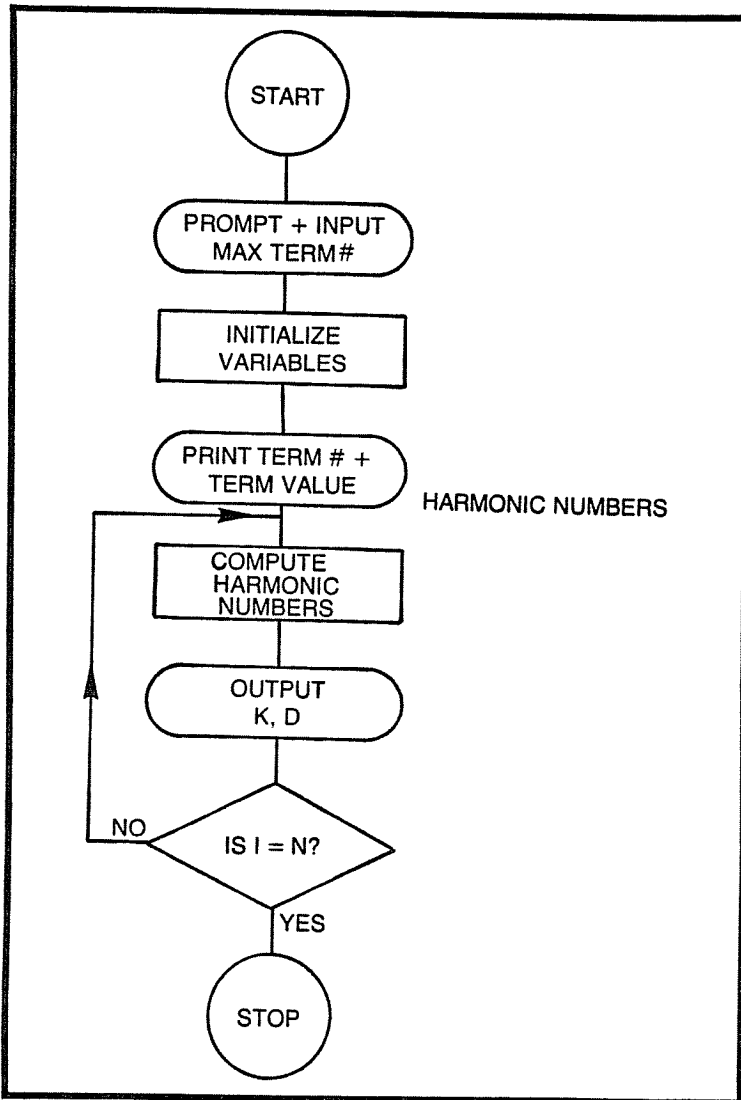
HARMONIC NUMBERS

```
10    REM THIS PROGRAM COMPUTES HARMONIC
          NUMBERS
20    PRINT "MAXIMUM TERM NUMBER"
30    INPUT N
40    LET K = 0
50    LET D = 0
60    PRINT "TERM NUMBER", "TERM VALUE"
70    FOR I = 0 TO N
80    LET K = I + 1
90    LET C = 1/K
100   LET D = D + C
110   PRINT K,D
120   NEXT I
130   PRINT
140   END
```

HARMONIC PROGRESSIONS

This program computes from the following information the values of
A and B, the difference and the number of terms desired. At the
user's option a table of progressions may be generated. In either
case the sum of the number of terms is produced.

FORMULA

$$\frac{A}{B} , \frac{A}{B + D} , \frac{A}{B + 2D}, \ldots \frac{A}{B + (N - 1)D}$$

EXAMPLE

VALUE OF A =
?
10
VALUE OF B =
?
2
DIFFERENCE =
?
3
NUMBER OF TERMS =
?
20
FOR TABLE TYPE 1, IF NOT TYPE 0
?
1
HARMONIC PROGRESSION
TERM NUMBER TERM VALUE

1	5
2	2
3	1.25
4	.90901
5	.714286
6	.588235
7	.5
8	.434783
9	.384615
10	.344828
11	.3125
12	.285714

TERM NUMBER	TERM VALUE
13	.263158
14	.243802
15	.227273
16	.212766
17	.2
18	.188679
19	.178571
20	.169492

SUM = 14.4078

****

TYPE 1 TO CONTINUE, 0 TO STOP

?

0

*END
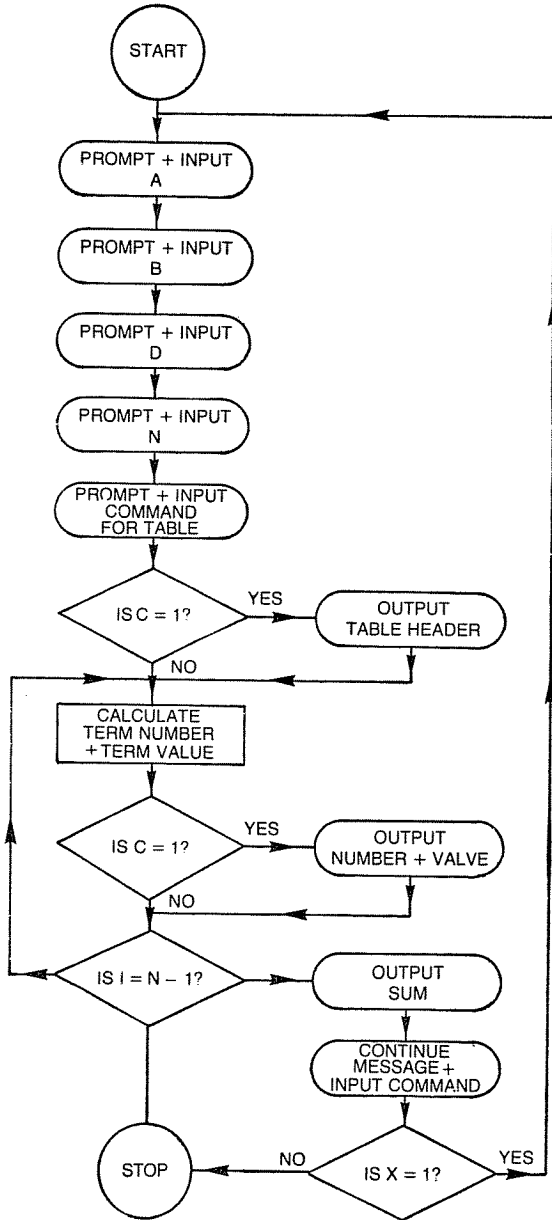
HARMONIC PROGRESSIONS

```
10    REM THIS PROGRAM COMPUTES A SERIES OF
      HARMONIC
20    REM PROGRESSIONS
30    PRINT "VALUE OF A = "
40    INPUT A
50    PRINT "VALUE OF B = "
60    INPUT B
70    PRINT "DIFFERENCE = "
80    INPUT D
90    PRINT "NUMBER OF TERMS = "
100   INPUT N
110   PRINT "FOR TABLE TYPE 1, IF NOT TYPE 0"
120   INPUT C
130   IF C = 1 THEN 150
140   GOTO 170
150   PRINT "HARMONIC PROGRESSION"
160   PRINT "TERM NUMBER","TERM VALUE"
170   LET J = 0
180   FOR I = 0 to N − 1
190   LET K = I + 1
200   LET L = A/(B + (I*D))
210   LET J = J + L
220   IF C = 1 THEN 240
230   GOTO 250
240   PRINT K,L
250   NEXT I
260   PRINT "SUM = ";J
270   PRINT "********************"
280   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
290   INPUT X
300   IF X = 0 THEN 320
310   STOP
320   PRINT
330   GOTO 30
340   END
```

HYDROCARBON COMBUSTION

This program simulates the burning of a hydrocarbon compound; complete combustion is assumed, and the option of excess air is available.

FORMULAE

AIR = 1 + % EXCESS AIR/100

$O_2 = C + S + H/4 - O/2$

AF(MOLES) = O_2(4.762)AIR

AF(MASS) = 1.8094(AF MOLES)/.7507C + 0.063H + 2.004S + 0.875N + O

TOTAL MOLES = O_2(4.762 AIR) + H/4 + O/2 + N/2

VOLUME % CO_2 = 100C/M

VOLUME % SO_2 = 100S/M

VOLUME % H_2O = 100H/2M

VOLUME % O_2 = 100(AIR − 1)O_2/M

VOLUME % N_2 = (100((3.762)AIR(02) + N/2)/M

EXAMPLE

RUN

FOR INSTRUCTIONS TYPE YES, IF NOT TYPE NO
?
NO
ENTER CARBON(C), HYDROGEN(H), OXYGEN(O), SULPHUR(S), NITROGEN(N) IN THAT ORDER
?
1,4,0,0,0
ENTER PERCENTAGE EXCESS AIR, IF ZERO
ENTER 0, EXAMPLE: − 34% ENTER AS 34
?
0
AIR-FUEL RATIO WITH RESPECT TO MOLES = 9.52
AIR-FUEL RATIO WITH RESPECT TO MASS = 17.19
TOTAL MOLES OF PRODUCTION = 10.52
******PERCENTAGE OF VOLUME OF PRODUCETS******
CARBON DIOXIDE = 9.50%
SULPHUR DIOXIDE = 0.0%
WATER = 19.0%
OXYGEN = 0.0%
NITROGEN = 71.49%
******COMPLETE COMBUSTION ASSUMED******
TO TRY NEXT COMPOUND TYPE YES
TO STOP TYPE NO
?
NO
COMBUSTION SAYS GOOD-BYE
*END

HYDROCARBON COMBUSTION
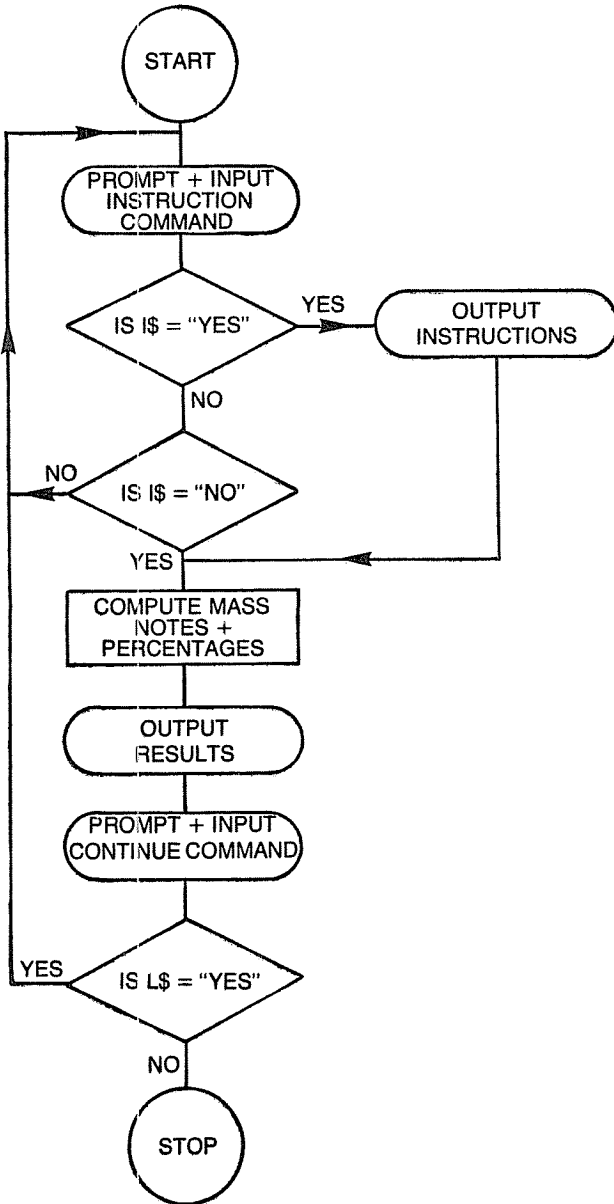
HYDROCARBON COMBUSTION

```
10      REM THIS PROGRAM COMPUTES THE PERCENT-
        AGES OF THE
20      REM PRODUCTS PRODUCED BY HYDROCARBON
        COMBUSTION
30      PRINT "FOR INSTRUCTIONS TYPE YES, IF NOT
        TYPE NO"
40      INPUT I$
50      IF I$ = "YES" THEN 90
60      IF I$ = "NO" THEN 130
70      PRINT "INVALID COMMAND"
80      GOTO 30
90      PRINT "THE AMOUNTS OF EACH ELEMENT MUST
        BE"
100     PRINT "ENTERED, EVEN IF THE AMOUNT IS ZERO"
110     PRINT "EXAMPLE: −  METHANE (CH4) MUST BE
        ENTERED AS"
120     PRINT "C;1, H;4, O;0, S;0, N;0"
130     PRINT
140     PRINT "ENTER CARBON(C), HYDROGEN(H), OXY-
        GEN(O)"
150     PRINT "SULPHUR(S), NITROGEN(N) IN THAT
        ORDER"
160     INPUT C,H,O,S,N
170     PRINT "ENTER PERCENTAGE EXCESS AIR, IF
        ZERO"
180     PRINT "ENTER 0, EXAMPLE: −  34%  ENTER AS 34"
190     INPUT E
200     LET E = 1 + (E/100)
210     LET O2 = C + S + (H/4) − (O/2)
220     LET A = O2*E*4.762
230     LET A1 = 1.8094*A
240     LET F = (0.7507*C) + (0.063*H) + (2.004*S)
250     LET F = (0.875*N) + O + F
260     LET A1 = A1/F
270     LET M = A + (HP4) + (O/2) + (N/2)
280     LET C2 = (C*100)/M
290     LET S2 = (S*100)/M
300     LET H2 = (H*100)/(2*M)
310     LET O3 = (100*(E− 1)*O2)/M
320     LET N2 = (100*((3.762*E*O2) +(N/2)))/M
330     PRINT
```

```
340    PRINT "AIR-FUEL RATIO WITH RESPECT TO
       MOLES = ";A
350    PRINT "AIR-FUEL RATIO WITH RESPECT TO
       MASS = ";A1
360    PRINT "TOTAL MOLES OF PRODUCT = ";M
370    PRINT "******PERCENTAGE VOLUME OF PRO-
       DUCTS******"
380    PRINT "CARBON DIOXIDE = ";C2;" % "
390    PRINT "SULPHUR DIOXIDE = ";S2;" % "
400    PRINT "WATER = ";H2;" % "
410    PRINT "OXYGEN = ";O3;" % "
420    PRINT "NITROGEN = ";N2;" % "
430    PRINT "******COMPLETE COMBUSTION AS-
       SUMED******"
440    PRINT
450    PRINT "TO TRY NEXT COMPOUND TYPE YES"
460    PRINT "TO STOP TYPE NO"
470    INPUT L$
480    IF L$ = "YES" THEN 510
490    PRINT "COMBUSTION SAYS GOOD-BYE"
500    STOP
510    PRINT
520    GOTO 30
530    END
```

HYPERBOLIC FUNCTIONS

This program computes the following hyperbolic trigonometric functions: sinH, cosH, tanH, cscH, secH, cotH.

FORMULAE

$$\sinh x = \frac{e^x - e^{-x}}{2} \qquad \operatorname{csch} x = \frac{1}{\sinh x}$$

$$\cosh x = \frac{e^x + e^{-x}}{2} \qquad \operatorname{sech} x = \frac{1}{\cosh x}$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad \coth x = \frac{1}{\tanh x}$$

EXAMPLE

TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
1
X =
?
4
SINH 4 = 27.2899
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
2
X =
?
5
COSH 5 = 74.2099
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
3
X =
?
.02
TANH .02 = .019997

```
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
4
X =
?
3.5
CSCH 3.5 = .060449
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
5
X =
12
SECH 12 = .000012
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
6
X =
?
.3
COTH .3 = 3.43273
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
```

```
          START

          PRINT
          OPTIONS

      PROMPT + INPUT
      DESIRED OPTION

        CALCULATE           OPTION          PROMPT +
         DESIRED           SUBROUTINE        INPUT X
         OPTION

         OUTPUT
      RESULT OF OPTION

      PROMPT + INPUT
     CONTINUE COMMAND

         IS L = 1?

          STOP
```
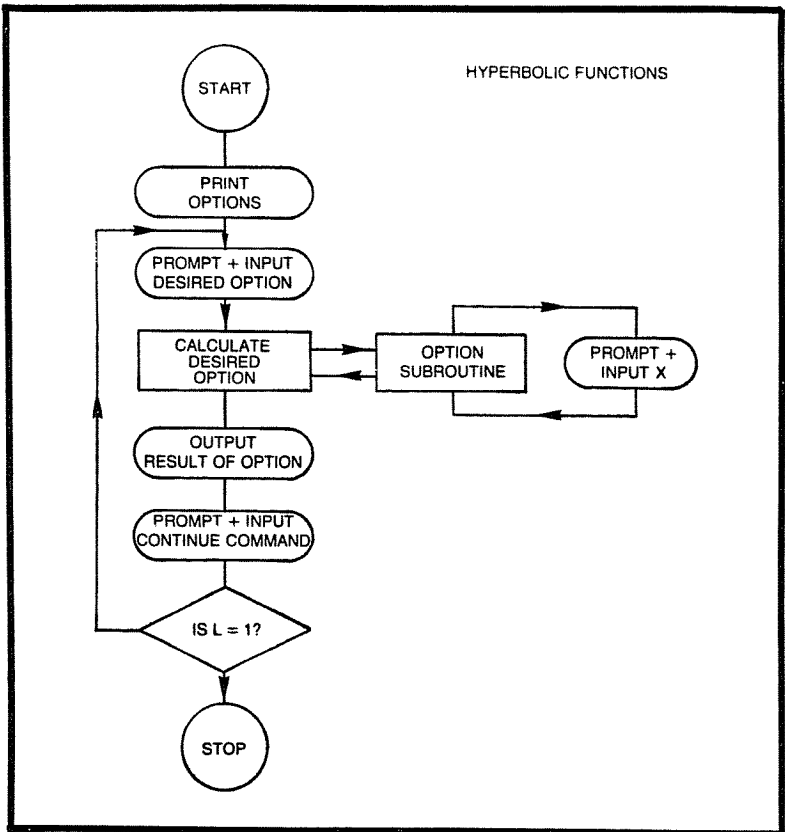
HYPERBOLIC FUNCTIONS

10   REM THIS PROGRAM COMPUTES HYPERBOLIC FUNCTIONS

20   PRINT "SINH (1)"

30   PRINT "COSH (2)"

40   PRINT "TANH (3)"

50   PRINT "CSCH (4)"

60   PRINT "SECH (5)"

70   PRINT "COTH (6)"

80   PRINT "TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED"

90   INPUT C

100   ON C GOTO 110,160,200,270,320,370

110   GOSUB 490

120   GOSUB 520

130   LET Z = Y/2

```
140      PRINT "SINH";X; =" ;Z
150      GOTO 420
160      GOSUB 490
170      GOSUB 540
180      LET Z = B/2
190      PRINT "COSH";X;" = ";Z
200      GOTO 420
210      GOSUB 490
220      GOSUB 520
230      GOSUB 540
240      LET Z = Y/B
250      PRINT "TANH";X;" = ";Z
260      GOTO 420
270      GOSUB 490
280      GOSUB 520
290      LET Z = 1/(Y/2)
300      PRINT "CSCH";X;" = ";Z
310      GOTO 420
320      GOSUB 490
330      GOSUB 540
340      LET Z = 1/(B/2)
350      PRINT "SECH";X;" = ";Z
360      GOTO 420
370      GOSUB 490
380      GOSUB 520
390      GOSUB 540
400      LET Z = 1/(Y/B)
410      PRINT "COTH";X;" = ";Z
420      PRINT
430      PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
440      INPUT L
450      IF L = 1 THEN 470
460      STOP
470      PRINT
480      GOTO 80
490      PRINT "X = "
500      INPUT X
510      RETURN
520      LET Y = EXP(X) – EXP( – X)
530      RETURN
540      LET B = EXP(X) + EXP( – X)
550      RETURN
560      END
```

I CHING (THE CHINESE BOOK OF CHANGES)

The I Ching is a collection of 64 hexagrams used to determine possible future events. It was inspired by the ancient Chinese. Each hexagram consists of six lines which may be broken (- -) or unbroken(---). The user of the I Ching may throw coins or sticks to produce the six lines. This program computes the six lines randomly on an algorithm using the date and time. The meanings of each of the possible 64 hexagrams may be looked up in the Chinese Book of Changes (The I Ching).

EXAMPLE

```
RUN
ENTER DATE AS M,D,Y
?
8,28,1977
ENTER TIME AS H,M(24 HOUR CLOCK)
?
9,56
---
---
- -
- -
---
- -

TO CONTINUE TYPE Y, IF NOT TYPE N
?
N
THE I CHING SAYS GOOD-BYE
*END
```

I CHING

START

PROMPT + INPUT
DATE + TIME

COMPUTE N

COMPUTE
C(I)

IS I = 6

PRINT
HEXAGRAM
C(I) = 1 OR 2

IS I = 6

PROMPT + INPUT
CONTINUE COMMAND

IS L$ = "4"

STOP
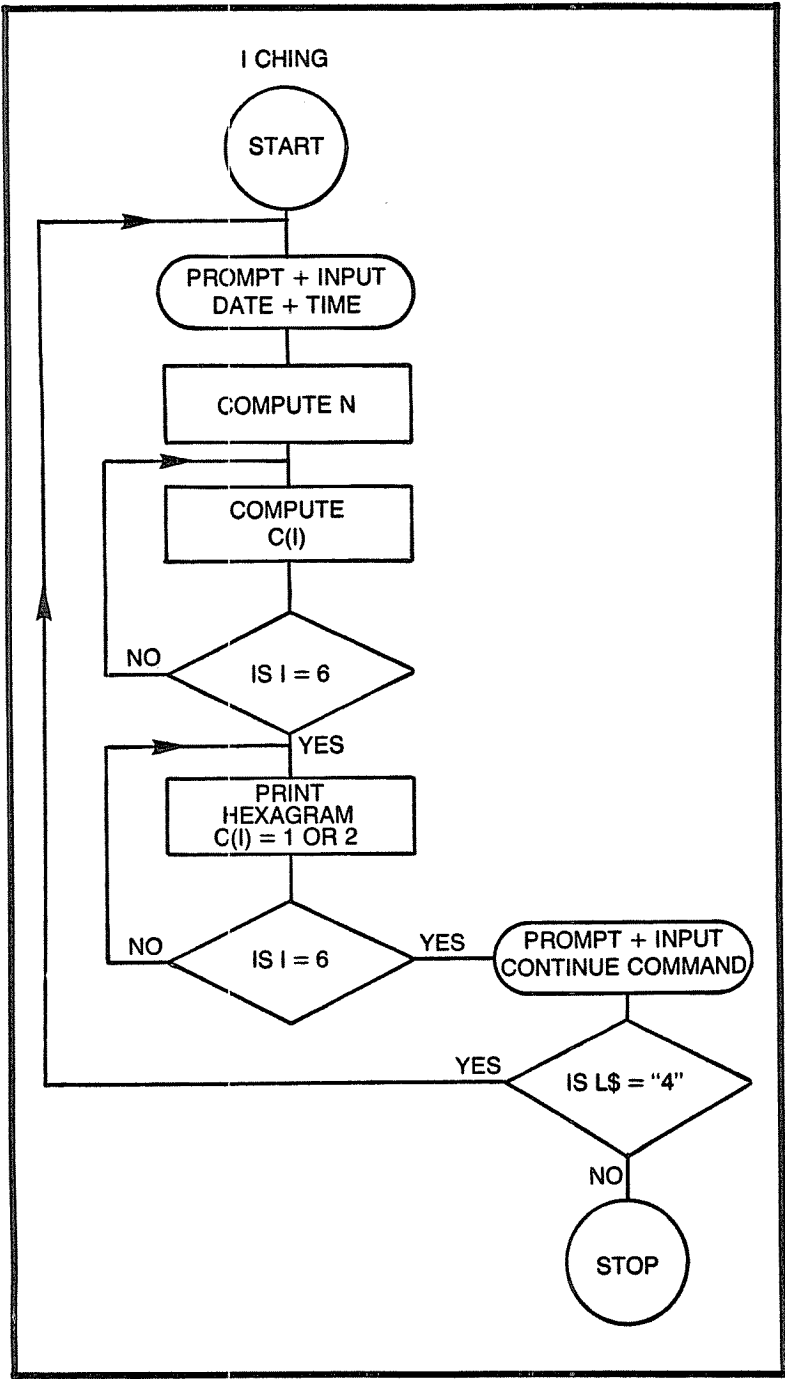
I CHING

```
10    REM THIS PROGRAM SIMULATES THE I CHING
20    RANDOMIZE
30    PRINT "ENTER DATE AS M,D,Y"
40    INPUT M,D,Y
50    PRINT "ENTER TIME AS H,M(24 HOUR CLOCK)"
60    INPUT H,M1
70    LET N = M + (D/30) + (Y/100) + H + (M1/60)
80    FOR I = 1 TO 6
90    LET K = N*RND(0)
100   LET K = K - INT(K)
110   LET C(I) = 1 + INT(2*K)
120   NEXT I
130   FOR I = 1 TO 6
140   IF C(I) = 1 THEN 170
150   PRINT "---"
160   GOTO 180
170   PRINT "- -"
180   NEXT I
190   PRINT
200   PRINT "TO CONTINUE TYPE Y, IF NOT TYPE N"
210   INPUT L$
220   IF L$ = "Y" THEN 250
230   PRINT "THE I CHING SAYS GOOD-BYE"
240   STOP
250   PRINT
260   GOTO 30
270   END
```

INTEGRAL BETWEEN TWO LIMITS

This program computes the integral between the limits of finite points A and B for single-valued function $f(x)$ by the six-point Gauss-Legendre quadrature formula.

FORMULA

$$\int_a^b f(x)dx = \frac{b-a}{2} \sum_{i=1}^b D_i f \frac{(C_i(b-a)+b+a)}{2}$$

EXAMPLE

Where $f(x) = 13*X\uparrow2 - 6*X\uparrow2 + SIN(X) + 1/X$
ENTER ENDPOINTS A,B
?
− 1,1
INTEGRAL = 4.28786
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

INTEGRAL BETWEEN TWO LIMITS
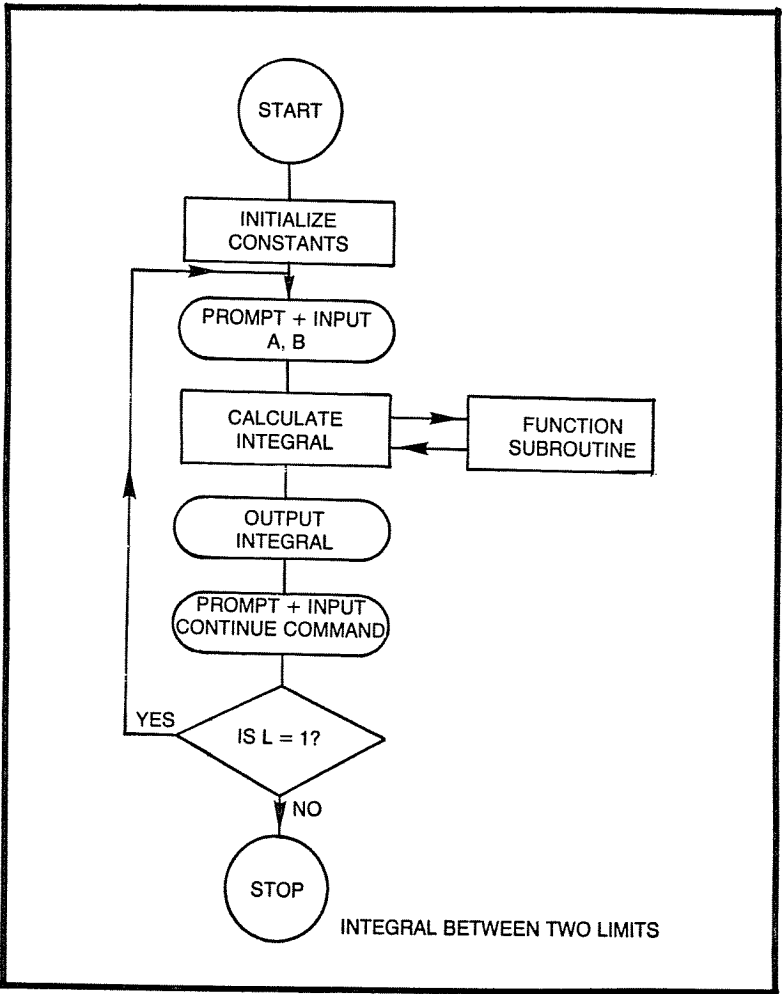
INTEGRAL BETWEEN TWO LIMITS

10	REM THIS PROGRAM COMPUTES THE INTEGRAL BETWEEN
20	REM THE LIMITS A AND B OF F(X)
30	LET C1 = .238619
40	LET C2 = C1
50	LET C3 = .661209
60	LET C4 = C3
70	LET C5 = .932470
80	LET C6 = C5
90	LET D1 = .467914

```
100   LET D2 = .360762
110   LET D3 = .171324
120   LET J = 0
130   PRINT "ENTER ENDPOINTS A,B"
140   INPUT A,B
150   LET Y = B – A
160   LET Z = B + A
170   LET X = ((C1*Y) + Z)/2
180   GOSUB 500
190   LET N = D1*Q
200   LET J = J + N
210   LET X = ((C2*Y) + Z)/2
220   GOSUB 500
230   LET N = D1*Q
240   LET J = J + N
250   LET X = ((C3*Y) + Z)/2
260   GOSUB 500
270   LET N = D2*Q
280   LET J = J + N
290   LET X = ((C4*Y) + Z)/2
300   GOSUB 500
310   LET N = D2*Q
320   LET J = J + N
330   LET X = ((C5*Y) + Z)/2
340   GOSUB 500
350   LET N = D3*Q
360   LET J = J + N
370   LET X = ((C6*Y) + Z)/2
380   GOSUB 500
390   LET N = D3*Y
400   LET J = J + N
410   LET G = (Y/2)*J
420   PRINT "INTEGRAL = ";G
430   PRINT
440   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
450   INPUT L
460   IF L = 1 THEN 480
470   STOP
480   PRINT
490   GOTO 120
500   LET Q = 13*X↑2 – 6*X↑2 + SIN(X) + 1/X
510   RETURN
520   END
```

INTERACTIVE GROWTH PATTERN

This program computes a growth pattern between two quantities called X and Y, where the existence of Y depends on the destruction of an X, but to be just, X can propagate.

EXAMPLE
NUMBER OF DESTROYERS (Y) =
?
201
NUMBER OF CREATORS (X) =
?
347
PROPAGATION RATE OF X =
?
6
CHANCE OF MEETING BETWEEN X AND Y =
?
.01
TIME BETWEEN SAMPLE DISPLAYED =
?
.5
MAXIMUM GENERATIONS =
?
10
INTERACTIVE GROWTH PATTERN

CREATOR	DESTROYER
1039	449
1822	2558
0	24600
0	12300
0	6150
0	3075
0	1537
0	768
0	384
0	192

MAXIMUM NUMBER OF GENERATIONS REACHED
FOR FURTHER GENERATIONS, TYPE 1, IF NOT 0
?
1
MAXIMUM GENERATIONS =
?

```
10
0          96
0          48
0          24
0          12
0          6
0          3
0          1
0          0
```
ALL GROWTH ENDED
NUMBER OF GENERATIONS = 18
FOR NEXT PATTERN TYPE 1, 0 TO STOP
?
0
*END

INTERACTIVE GROWTH PATTERN
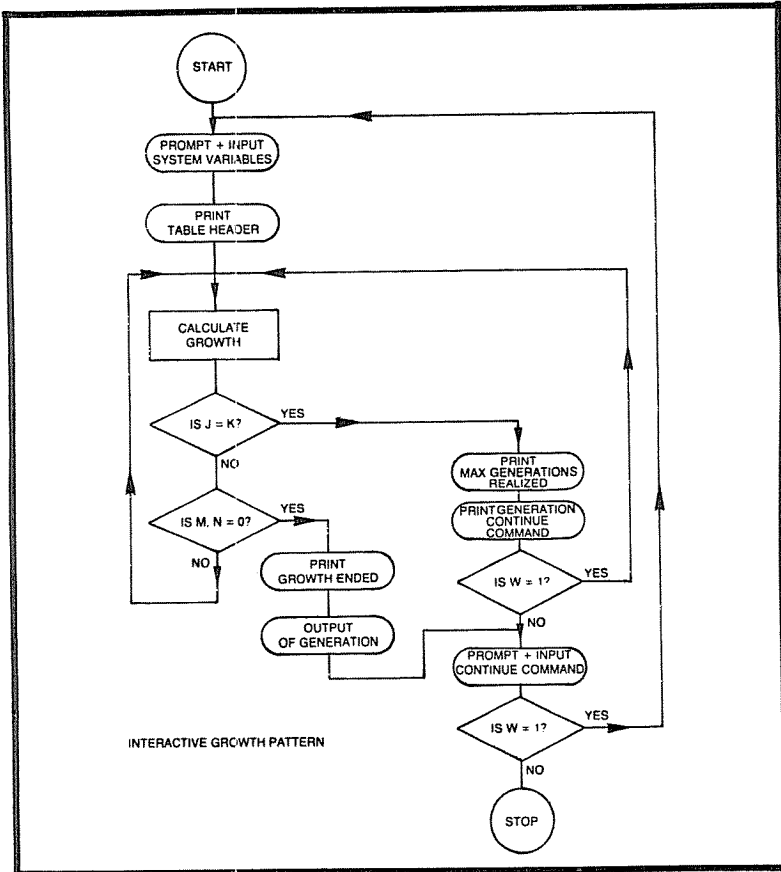
INTERACTIVE GROWTH PATTERN

```
10    REM THIS PROGRAM COMPUTES THE GROWTH
      PATTERN BETWEEN
20    REM A QUANTITY Y THE DESTROYER AND A
      QUANTITY X
30    REM THE CREATOR. X CAN PROPAGATE, AND ON A
40    REM CHANCE METTING BETWEEN X AND Y, X IS
      DESTROYED,
50    REM THUS INCREASING THE NUMBER OF YS.
60    PRINT "NUMBER OF DESTROYERS (Y) = ";
70    INPUT Y
80    PRINT "NUMBER OF CREATORS (X) = ";
90    INPUT X
100   PRINT "PROPAGATION RATE OF X = ";
110   INPUT Z
120   PRINT "CHANCE OF METTING BETWEEN X AND Y
      = ";
130   INPUT A
140   PRINT "TIME BETWEEN SAMPLES = ";
150   INPUT H
160   PRINT "MAXIMUM GENERATIONS = ";
170   INPUT K
180   LET J = 0
190   PRINT
200   PRINT "INTERACTIVE GROWTH PATTERN"
210   PRINT
220   PRINT "CREATOR","DESTROYER"
230   IF J = K THEN 470
240   LET J = J + 1
250   LET B = A*X*Y
260   LET C = ((Y − B)*H) + Y
270   IF C < 0 THEN 390
280   LET Y = C
290   LET D = (((X*Z) − B)*H) + X
300   IF D < 0 THEN 140
310   LET X = D
320   LET M = INT(X)
330   LET N = INT(Y)
340   PRINT M,N
350   IF M = 0 THEN 370
360   GOTO 230
370   IF N = 0 THEN 430
380   GOTO 230
```

```
390    LET Y = 0
400    GOTO 290
410    LET X = 0
420    GOTO 320
430    PRINT "ALL GROWTH ENDED"
440    PRINT "NUMBER OF GENERATIONS = ";J
450    PRINT
460    GOTO 560
470    PRINT "MAXIMUM NUMBER OF GENERATIONS
       REACHED"
480    PRINT
490    PRINT "FOR FURTHER GENERATIONS TYPE 1, IF
         NOT 0"
500    INPUT W
510    IF W = 1 THEN 530
520    GOTO 560
530    PRINT "MAXIMUM GENERATIONS = ";
540    INPUT K
550    GOTO 230
560    PRINT
570    PRINT "FOR NEXT PATTERN TYPE 1, 0 TO STOP"
580    INPUT W
590    IF W = 1 THEN 610
600    STOP
610    PRINT
620    GOTO 60
630    END
```

INVERSE HYPERBOLIC FUNCTIONS

This program computes the following hyperbolic trigonometric functions: $sinH^{-1}$, $cosH^{-1}$, $tanH^{-1}$, $cscH^{-1}$, $secH^{-1}$ $cotH^{-1}$

FORMULAE

$$sin \; H^{-1}x = ln(X + (x^2 + 1)^{.5})$$

$$cos \; H^{-1}x = ln(x + (x^2 - 1)^{.5})$$

$$tan \; H^{-1}x = \frac{1}{2} \; ln\left(\frac{1+x}{1-x}\right)$$

$$csc \; H^{-1}x = sin \; H^{-1}\left(\frac{1}{x}\right)$$

$$sec \; H^{-1}x = cos \; H^{-1}\left(\frac{1}{x}\right)$$

$$cot \; H^{-1}x = tan \; H^{-1}\left(\frac{1}{x}\right)$$

EXAMPLE

TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
1
X =
?
12
SINH − 1 12 = 3.17979
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
2
X =
?
45
COSH − 1 45 = 4.49969
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
3

```
X =
?
.00055
TANH − 1 .00055 == .00055
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
4
X =
?
23
CSCH − 1 23 = .04346
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
5
X =
?
.125
SECH − 1 .125 = 2.76866
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
TYPE A NUMBER 1 TO 6 FOR FUNCTION DESIRED
?
6
X =
?
8
COTH − 1 8 = .12566
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
```

INVERSE HYPERBOLIC FUNCTIONS
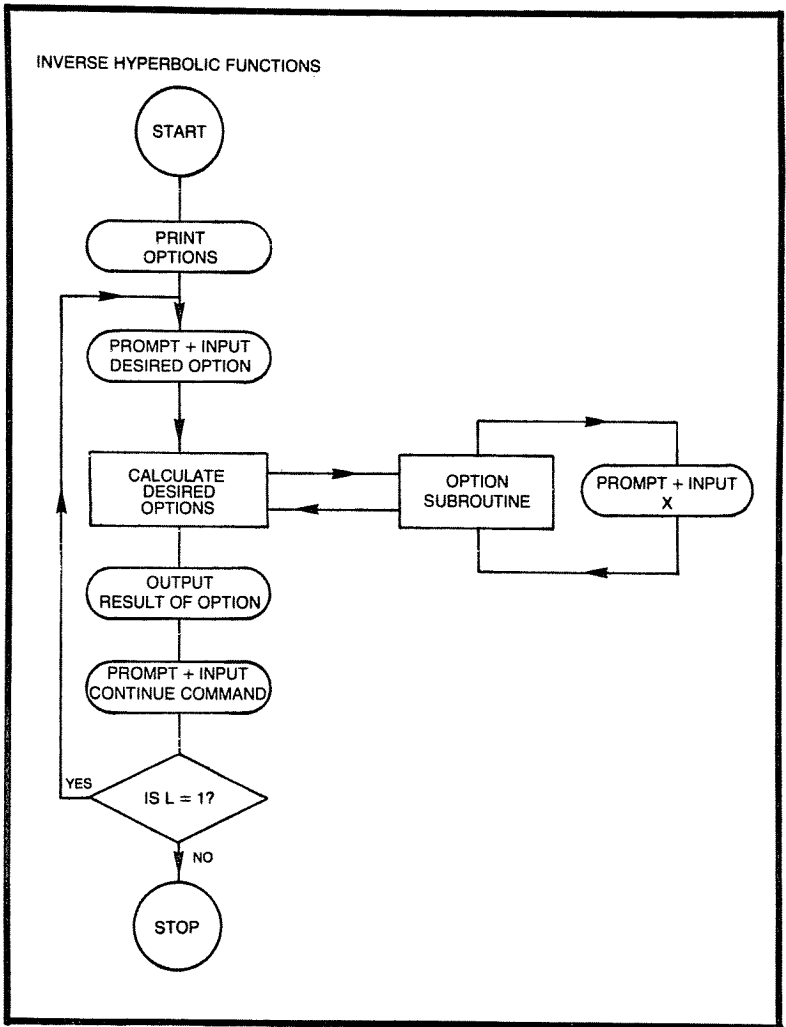
INVERSE HYPERBOLIC FUNCTIONS

10   REM THIS PROGRAM COMPUTES INVERSE HYPER-
        BOLIC FUNCTIONS
20   PRINT "SINH – 1 (1)"
30   PRINT "COSH – 1 (2)"
40   PRINT "TANH – 1 (3)"
50   PRINT "CSCH – 1 (4)"
60   PRINT "SECH – 1 (5)"
70   PRINT "COTH – 1 (6)"

```
80   PRINT "TYPE A NUMBER 1 TO 6 FOR FUNCTION
     DESIRED"
90   INPUT C
100  ON C GOTO 110,150,190,230,280,330
110  GOSUB 440
120  GOSUB 470
130  PRINT "SINH − 1";X;" = ";Z
140  GOTO 370
150  GOSUB 440
160  GOSUB 490
170  PRINT "COSH − 1";X;" = ";Z
180  GOTO 370
190  GOSUB 440
200  GOSUB 510
210  PRINT "TANH − 1";X;" = ";Z
220  GOTO 370
230  GOSUB 530
240  GOSUB 470
250  LET X = A
260  PRINT "CSCH − 1";X;" = ";Z
270  GOTO 370
280  GOSUB 530
290  GOSUB 490
300  LET X = A
310  PRINT "SECH − 1";X;" = ";Z
320  GOTO 370
330  GOSUB 530
340  GOSUB 510
350  LET X = A
360  PRINT "COTH − 1";X;" = ";Z
370  PRINT
380  PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
390  INPUT L
400  IF L = 1 THEN 420
410  STOP
420  PRINT
430  GOTO 80
440  PRINT "X = ";
450  INPUT X
460  RETURN
470  LET Z = LOG(X + SQR((X↑2) + 1))
480  RETURN
490  LET Z = LOG(X + SQR((X↑2) − 1))
```

```
500     RETURN
510     LET Z = (LOG((1 + X/(1 − X)))/2
520     RETURN
530     PRINT "X = ";
540     INPUT X
550     LET A = X
560     LET X = 1/X
570     RETURN
580     END
```

L-PAD MINIMUM LOSS SYSTEM

In systems where two resistive stages must be coupled, a minimum loss L-pad can be used for matching. A typical application for this pad would be to couple inputs and outputs of audio circuits. The user inputs the two impedances Z_1 and Z_2; the program responds with the L-pad resistors R_1 and R_2, and also the system loss in decibels.

FORMULAE

L PAD MIN LOSS
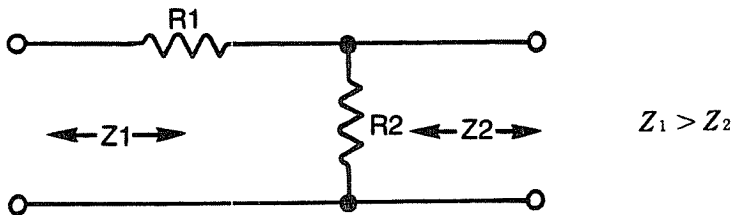
$$R_1 = Z_1 \sqrt{1 - \frac{Z_2}{Z_1}}$$

$$R_2 = \sqrt{\frac{Z_2}{1 - \dfrac{Z_2}{Z_1}}}$$

$$LOSS = 20 \log \left(\sqrt{\frac{Z_1}{Z_2}} + \sqrt{\frac{Z_1}{Z_2} - 1} \right)$$

EXAMPLE

1ST IMPEDANCE
?
300
2ND IMPEDANCE
?
75
COMPONENTS OF THE L-PAD
RESISTOR 1 = 259.807
RESISTOR 2 = 86.6025
LOSS IN DECIBELS = 11.4389
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

118

L-PAD MINIMUM LOSS SYSTEM

```
                    ╭─────────╮
                    │  START  │
                    ╰────┬────╯
                         │         ◄────────────┐
                         ▼                       │
                ╭─────────────────╮              │
                │  PROMPT + INPUT │              │
                │  "IMPEDANCES"   │              │
                ╰────────┬────────╯              │
                         ▼                       │
                ┌─────────────────┐              │
                │   CALCULATE     │              │
                │   COMPONENTS    │              │
                │      OF         │              │
                │   JT AND T SYS. │              │
                └────────┬────────┘              │
                         ▼                       │
                ╭─────────────────╮              │
                │     PRINT       │              │
                │  "COMPONENTS"   │              │
                ╰────────┬────────╯              │
                         ▼                       │
                ╭─────────────────╮              │
                │    OUTPUT       │              │
                │ RESISTOR VALUES │              │
                ╰────────┬────────╯              │
                         ▼                       │
                ╭─────────────────╮              │
                │    OUTPUT       │              │
                │ LOSS IN DECIBELS│              │
                ╰────────┬────────╯              │
                         ▼                       │
                ╭─────────────────╮              │
                │  PROMPT + INPUT │              │
                │ CONTINUE COMMAND│              │
                ╰────────┬────────╯              │
                         ▼                       │
                      ╱─────╲          YES       │
                     ╱IS X = 1?╲─────────────────┘
                      ╲─────╱
                         │ NO
                         ▼
                    ╭─────────╮
                    │  STOP   │
                    ╰─────────╯
```
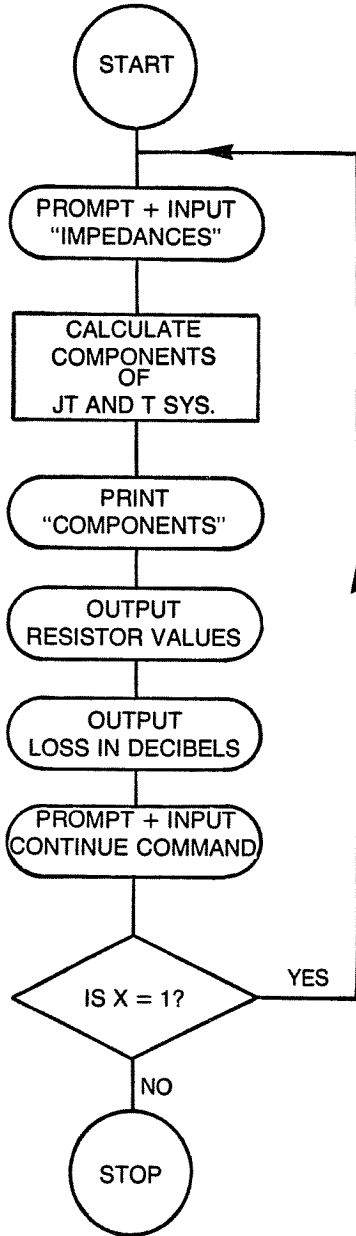
L-PAD MINUMUM LOSS SYSTEM

```
10    REM THIS PROGRAM COMPUTES THE VALUES OF
20    REM THE TWO RESISTANCES REQUIRED TO CON-
         STRUCT A
30    REM L-PAD OF MINIMUM LOSS
40    PRINT "1ST IMPEDANCE"
50    INPUT Z1
60    PRINT "2ND IMPEDANCE"
70    INPUT Z2
80    LET R1 = Z1*SQR(1 − (Z2/Z1))
90    LET R2 = Z2/SQR(1 − (Z2/Z1))
100   LET M = SQR(Z1/Z2) + SQR((Z1/Z2) − 1)
110   LET L = 20*(LOG(M)/LOG(10))
120   PRINT "COMPONENTS OF THE L-PAD"
130   PRINT "RESISTOR 1 = ";R1
140   PRINT "RESISTOR 2 = ";R2
150   PRINT "LOSS IN DECIBELS = ";L
160   PRINT
170   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
180   INPUT X
190   IF X = 1 THEN 210
200   STOP
210   PRINT
220   GOTO 40
230   END
```

LINEAR INTERPOLATION

If Y is a function of X, and $Y_1 - Y_2$ are the values of the function at $X_1 - X_2$, respectively, Y may be computed for any value of X.

FORMULA

LINEAR INTEPOLATION
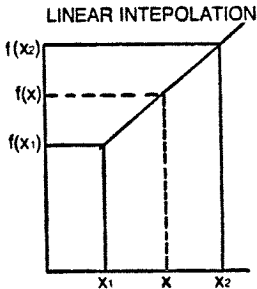
$$f(x) \cong \frac{(x_2 - x)(f(x_1)) + (x - x_1)(f(x_2))}{x_2 - x_1}$$

EXAMPLE

```
KNOWN VALUE OF X (X1,X2) =
?
10,50
KNOWN VALUES OF Y (F(X1),F(X2)) =
?
15,55
INTERPOLATE F(X) AT X =
?
60
INTERPOLATED F(X) = 65
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
KNOWN VALUE OF X (X1,X2) =
?
43,98
KNOWN VALUES OF Y (F(X1), F(X2)) =
?
86,196
INTERPOLATE F(X) AT X =
?
1234
INTERPOLATED F(X) = 2468
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
```

LINEAR INTERPOLATION

```
        ( START )
            │
            ▼ ◄──────────────┐
  ( PROMPT + INPUT           │
    KNOWN VALUES )           │
            │                │
  ( PROMPT + INPUT           │
        X )                  │
            │                │
  [  CALCULATE               │
        F(x)  ]              │
            │                │
  ( OUTPUT                   │
     F (x) )                 │
            │                │
  ( PROMPT + INPUT           │
    CONTINUE COMMAND )       │
            │                │
         ◇ IS L = 1? ◇ ─ YES ┘
            │
           NO
            ▼
        ( STOP )
```
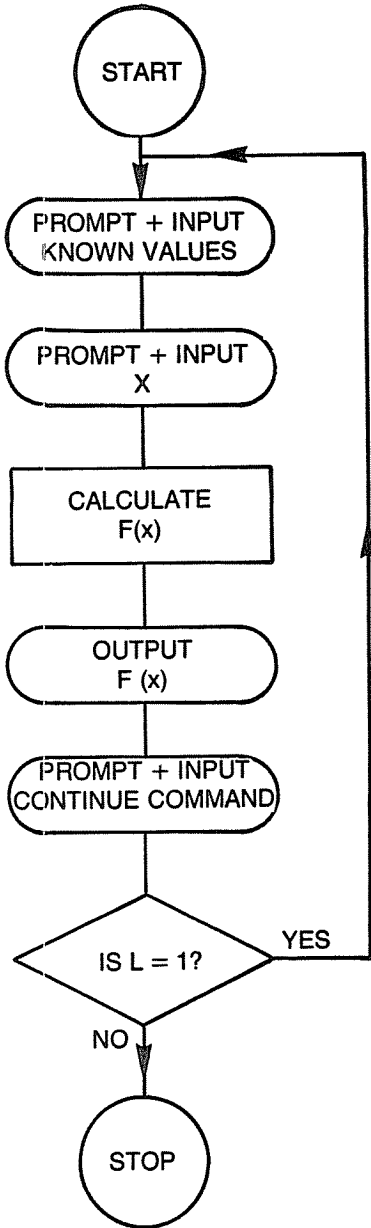
LINEAR INTERPOLATION

```
10   REM THIS PROGRAM COMPUTES LINEAR INTER-
     POLATION
20   PRINT "KNOWN VALUE OF X (X1,X2) = ";
30   INPUT X1,X2
40   PRINT "KNOWN VALUES OF Y (F(X1),F(X2)) = ";
50   INPUT Y1,Y2
60   PRINT "INTERPOLATE F(X) AT X = ";
70   INPUT X
80   LET G = ((X2 − X)*Y1) + ((X − X1)*Y2)
90   LET F =G/(X2 − X1)
100  PRINT "INTERPOLATED F(X) = ";F
110  PRINT
120  PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
130  INPUT L
140  IF L = 1 THEN 160
150  STOP
160  PRINT
170  GOTO 20
180  END
```

123

LOGARITHMS OF ANY BASE

This simple program allows the user to compute the value of a logarithm to any base. The number X and the base Y must both be positive if machine error is not to occur.

FORMULA

$$log_y x = \frac{\ln x}{\ln y}$$

EXAMPLES

BASE =
?
16
X =
?
54
LOG 54 = 1.43872
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
BASE =
?
567
X =
?
23
LOG 23 = .494529
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

LOGARITHMS OF ANY BASE

START

PROMPT + INPUT
BASE

PROMPT + INPUT
X

CALCULATE
LOG Y^x

OUTPUT
LOG X

PROMPT + INPUT
CONTINUE COMMAND

IS L = 1?

YES

NO
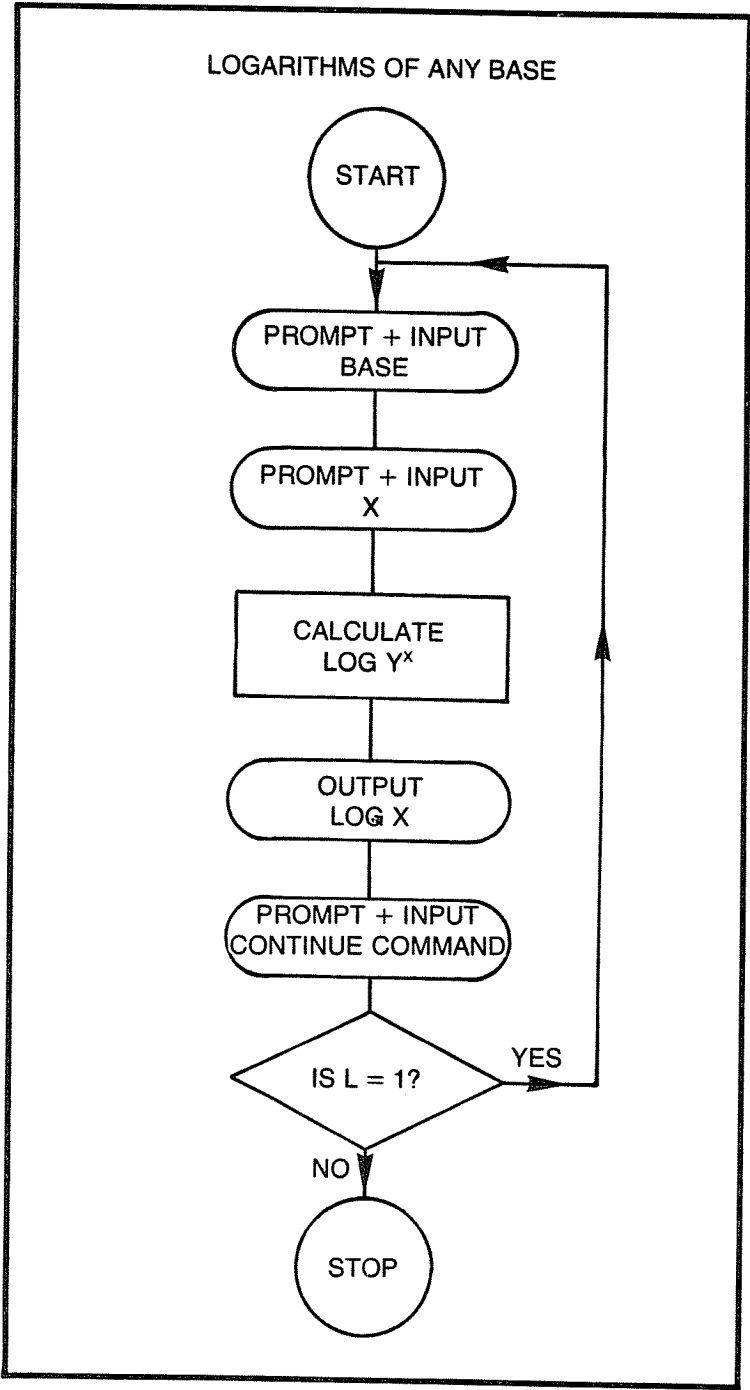
STOP

LOGARITHMS OF ANY BASE

```
10    REM THIS PROGRAM WILL COMPUTE THE LOG
20    REM OF ANY POSITIVE NUMBER X, TO ANY POSI-
      TIVE
30    REM BASE Y
40    PRINT "BASE = "
50    INPUT Y
60    PRINT "X = "
70    INPUT X
80    LET J = LOG(X)/LOG(Y)
90    PRINT "LOG";X;" = ";J
100   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
110   INPUT L
120   IF L = 1 THEN 140
130   STOP
140   PRINT
150   GOTO 40
160   END
```

MEAN, STANDARD DEVIATION, STANDARD ERROR FOR GROUPED DATA

This program generates the mean, standard deviation and standard error for grouped data until the user enters a 0 for the value of the frequency.

FORMULAE

$$Mean \; \overline{x} = \frac{\Sigma f_i x_i}{\Sigma f_i}$$

$$Standard \; error \; S\overline{x} = \frac{S}{\sqrt{\Sigma f_i}}$$

$$Standard \; deviation \; S = \frac{\sqrt{\Sigma f_i \; x_i^2 - (\Sigma f_i) \overline{x}^2}}{\Sigma f_i - 1}$$

EXAMPLE

ENTER SAMPLE VALUE AND FREQUENCY
?
5,2
ENTER SAMPLE VALUE AND FREQUENCY
?
10,3
ENTER SAMPLE VALUE AND FREQUENCY
?
6,1
ENTER SAMPLE VALUE AND FREQUENCY
?
0,0
NUMBER OF SAMPLES ENTERED = 3
MEAN = 7.67
STANDARD DEVIATION = 5.77
STANDARD ERROR = 2.36
TYPE 1 TO CONTINUF, 0 TO STOP
?
0
*END

MEAN, STANDARD DEVIATION, AND STANDARD ERROR FOR GROUPED DATA

```
                        ┌─────────┐
                        │  START  │
                        └────┬────┘
                             │        ┌──────────────────◄──────────┐
                        ┌────▼─────┐                                 │
                        │ INITIALIZE │                               │
                        │ VARIABLES  │                               │
                        └────┬─────┘                                 │
                             │                                       │
                    ┌────────▼────────┐                             │
                    │ PROMPT = INPUT  │                             │
                    │ SAMPLE = FREQ   │                             │
                    └────────┬────────┘                             │
                             │                                       │
                        ◇────▼────◇    YES                          │
                       │ IS F <= 0? │──────►──────┐                 │
                        ◇────┬────◇               │                 │
                          NO │                     │                 │
                    ┌────────▼────────┐   ┌────────▼────────┐       │
                    │     UPDATE      │   │    COMPUTE      │       │
                    │  CALCULATIONS   │   │    OUTPUT       │       │
                    └─────────────────┘   └────────┬────────┘       │
                                                   │                 │
                                          ┌────────▼────────┐       │
                                          │    OUTPUT       │       │
                                          │   "STAT" DATA   │       │
                                          └────────┬────────┘       │
                                                   │                 │
                                          ┌────────▼────────┐       │
                                          │ PROMPT + INPUT  │       │
                                          │ CONTINUE COMMAND│       │
                                          └────────┬────────┘       │
                                                   │                 │
                                              ◇────▼────◇   YES      │
                                             │ IS L = 1? │───────────┘
                                              ◇────┬────◇
                                                NO │
                                          ┌────────▼────────┐
                                          │      STOP       │
                                          └─────────────────┘
```
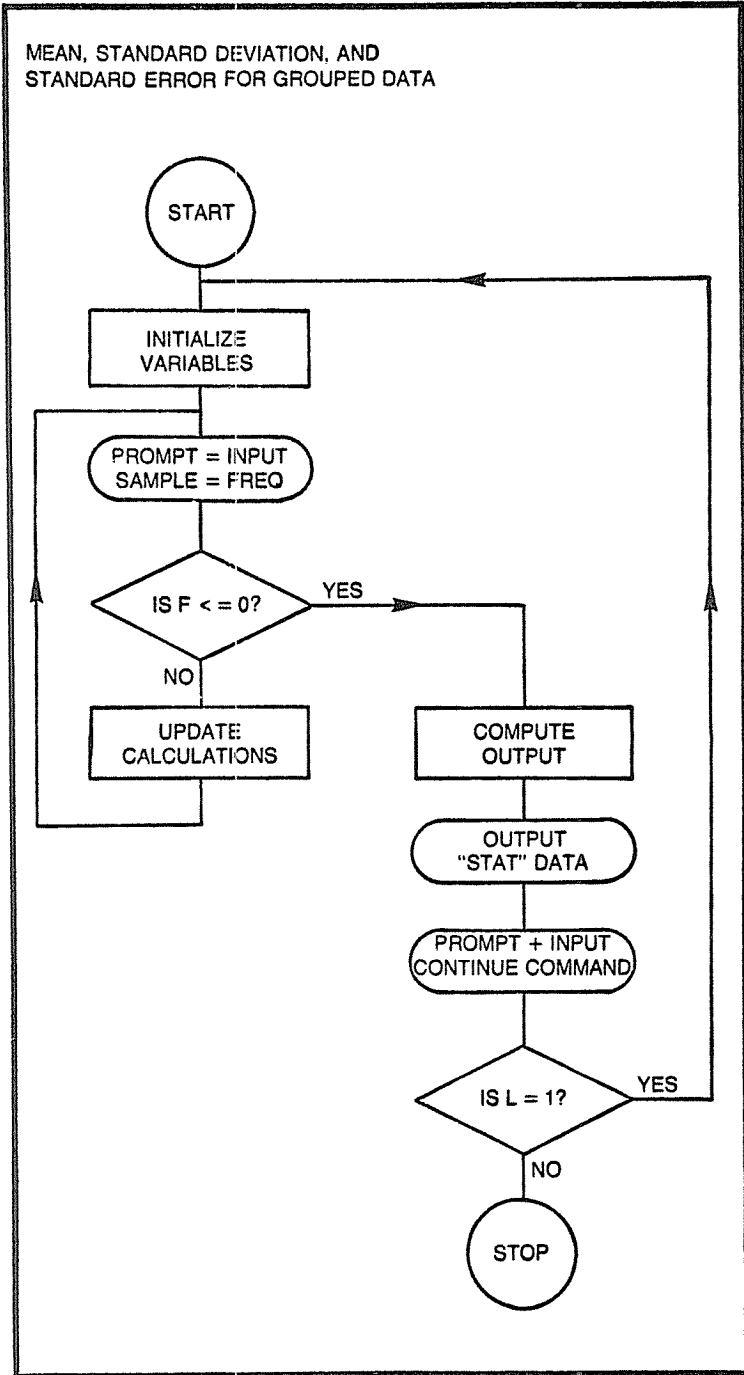
MEAN, STANDARD DEVIATION, AND
STANDARD ERROR FOR GROUPED DATA

```
10    REM THIS PROGRAM COMPUTES THE MEAN,
      STANDARD
20    REM DEVIATION AND STANDARD ERROR FOR
      GROUPED DATA
30    LET A = 0
40    LET B = 0
50    LET C = 0
60    LET D = 0
70    PRINT "ENTER SAMPLE VALUE AND FREQUENCY";
80    INPUT X,F
90    IF F < = 0 THEN 170
100   LET A = A + 1
110   LET B = B + F
120   LET G = X*F
130   LET C = C + G
140   LET E = (X↑2)*F
150   LET D = D + E
160   GOTO 70
170   LET H = C/B
180   LET I = SQR(D - (B*(H↑2)))
190   LET J = I/SQR(B)
200   PRINT
210   PRINT "NUMBER OF SAMPLES ENTERED = ";A
220   PRINT "MEAN = ";H
230   PRINT "STANDARD DEVIATION = ";I
240   PRINT "STANDARD ERROR = ";J
250   PRINT
260   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
270   INPUT L
280   IF L = 1 THEN 300
290   STOP
300   PRINT
310   GOTO 30
320   END
```

MOMENTS, SKEWNESS AND KURTOSIS

This program computes the first four moments where the first moment is the mean of the distribution and the second moment is the variance. Skewness is the departure of a frequency distribution from symmetry, and kurtosis is a property of distribution that expresses its relative peakedness.

FORMULAE

$$1^{ST}M = \overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$2^{ND} M = M_2 = \frac{1}{n} \sum x_i^2 - \overline{x}^2$$

$$3^{RD}M = M_3 = \frac{1}{n} \sum x_i^3 - \frac{3}{n} \overline{x} \sum x_i^2 + 2x^3$$

$$4^{TH}M = M_4 = \frac{1}{n} \sum x_i^4 - \frac{4}{n} \overline{x} \sum x_i^3 + \frac{6}{n} \overline{x}^2 \sum x_i^2 - 3\overline{x}$$

$$SKEWNESS = \gamma_1 = \frac{M_3}{M_2^{3/2}}$$

$$KURTOSIS \ \gamma_2 = \frac{M_4}{M_2^2}$$

EXAMPLE

NUMBER OF SAMPLES =
?
5
SAMPLE =
?
12
SAMPLE =
?
3
SAMPLE =
?
4

130

```
SAMPLE =
?
5
SAMPLE =
?
4
1ST MOMENT = 5.6
2ND MOMENT = 10.64
3RD MOMENT = 47.2319
4TH MOMENT = 347.331
MOMENT COEFFICIENTS
SKEWNESS = 1.36089 KURTOSIS = 3.06803
**************************
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
```

MOMENTS, SKEWNESS, AND KURTOSIS

```
                    ┌─────────┐
                    │  START  │
                    └────┬────┘
                         │
              ┌──────────┴──────────┐
              │      INITIALIZE     │
              │   A, B, C,...TO O   │
              └──────────┬──────────┘
                         │
              ╭──────────┴──────────╮
              │   PROMPT + INPUT    │
              │     NUMBER OF       │
              │      SAMPLES        │
              ╰──────────┬──────────╯
                         │
              ╭──────────┴──────────╮
              │   PROMPT + INPUT    │
              │      SAMPLES        │
              ╰──────────┬──────────╯
                         │
              ┌──────────┴──────────┐
              │     CALCULATE       │
              │     A, B, C, D...   │
              └──────────┬──────────┘
                         │
                    ◇ IS I = N? ◇
                         │ YES
              ┌──────────┴──────────┐
              │     CALCULATE       │
              │     MOMENTS +       │
              │    COEFFICIENTS     │
              └──────────┬──────────┘
                         │
              ╭──────────┴──────────╮
              │      OUTPUT         │
              │     MOMENTS +       │
              │    COEFFICIENTS     │
              ╰──────────┬──────────╯
                         │
              ╭──────────┴──────────╮
              │     CONTINUE        │
              │  MESSAGE + INPUT    │
              ╰──────────┬──────────╯
                         │
                    ◇ IS Z = 1? ◇
                         │ NO
                    ┌────┴────┐
                    │  STOP   │
                    └─────────┘
```
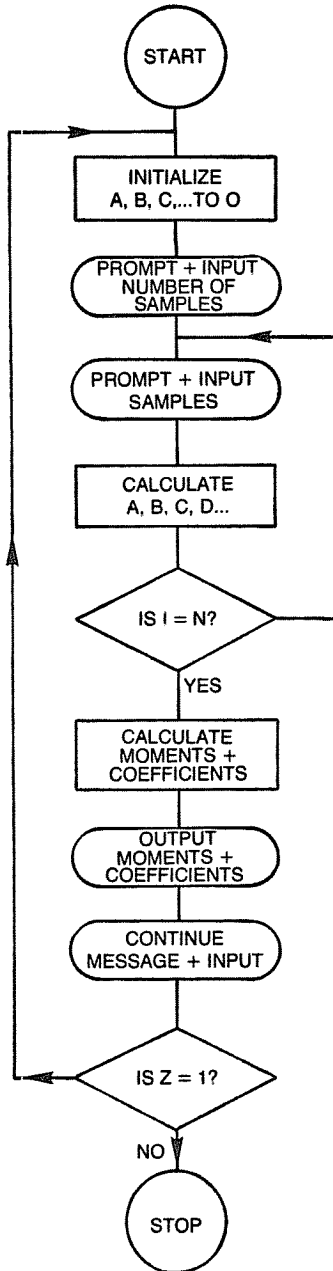
MOMENTS, SKEWNESS, AND KURTOSIS

```
10    REM THIS PROGRAM COMPUTES THE VALUES OF
      THE
20    REM FIRST 4 MOMENTS, SKEWNESS AND KUR-
      TOSIS
30    LET A = 0
40    LET B = 0
50    LET C = 0
60    LET D = 0
70    PRINT "NUMBER OF SAMPLES = "
80    INPUT N
90    FOR I = 1 TO N
100   PRINT "SAMPLE = "
110   INPUT X
120   LET A = A + X
130   LET B = B + (X↑2)
140   LET C = C + (X↑3)
150   LET D = D + (X↑4)
160   NEXT I
170   LET J = A/N
180   LET E = (B/N) – (J↑2)
190   LET F = (C/N) – ((3*J*B)/N) + (2*(J↑3))
200   LET G = (D/N) – ((4*J*C)/N)
210   LET H = G + ((6*(J↑2)*B)/N) – (3*(J↑4))
220   LET K = F/(E↑(3/2))
230   LET L = H/(E↑2)
240   PRINT "1ST MOMENT = ";J
250   PRINT "2ND MOMENT = ";E
260   PRINT "3RD MOMENT = ";F
270   PRINT "4TH MOMENT = ";H
280   PRINT "MOMENT COEFFICIENTS"
290   PRINT "SKEWNESS = ";K,"KURTOSIS = ";L
300   PRINT"*********************"
310   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
320   INPUT Z
330   IF Z = 1 THEN 350
340   STOP
350   PRINT
360   GOTO 30
370   END
```

NO REPETITIONS PROBABILITY

The user dictates the size of a population to be determined for a "No Repetitions" probability.

FORMULA

$$P = \left(1 - \frac{1}{M}\right)\left(1 - \frac{2}{M}\right)\cdots\left(1 - \frac{N-1}{M}\right)$$

where $M \geq N \geq 1$

EXAMPLE

```
**************************
TOTAL POPULATION
?
56
SIZE OF SAMPLE
?
2
PROBABILITY = .982143
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
**************************
TOTAL POPULATION
?
9
SIZE OF SAMPLE
?
4
PROBABILITY = .460905
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
```

NO REPETITIONS PROBABILITY

```
        ( START )
            │
            ▼
   ┌──────────────────┐
   │  PROMPT + INPUT  │
   │ TOTAL POPULATION │
   └──────────────────┘
            │
            ▼
   ┌──────────────────┐
   │  PROMPT + INPUT  │
   │  SIZE OF SAMPLE  │
   └──────────────────┘
            │
            ▼
   ┌──────────────────┐
   │    CALCULATE     │
   │   PROBABILITY    │
   └──────────────────┘
            │
            ▼
  NO      ╱ IS i = N – 1? ╲
 ◄────────╲               ╱
            ▼ YES
   ┌──────────────────┐
   │     OUTPUT       │
   │   PROBABILITY    │
   └──────────────────┘
            │
            ▼
   ┌──────────────────┐
   │  PROMPT + INPUT  │
   │ CONTINUE COMMAND │
   └──────────────────┘
            │
            ▼
         ╱ IS L = 1? ╲  YES
         ╲           ╱ ─────►
            ▼ NO
        ( STOP )
```
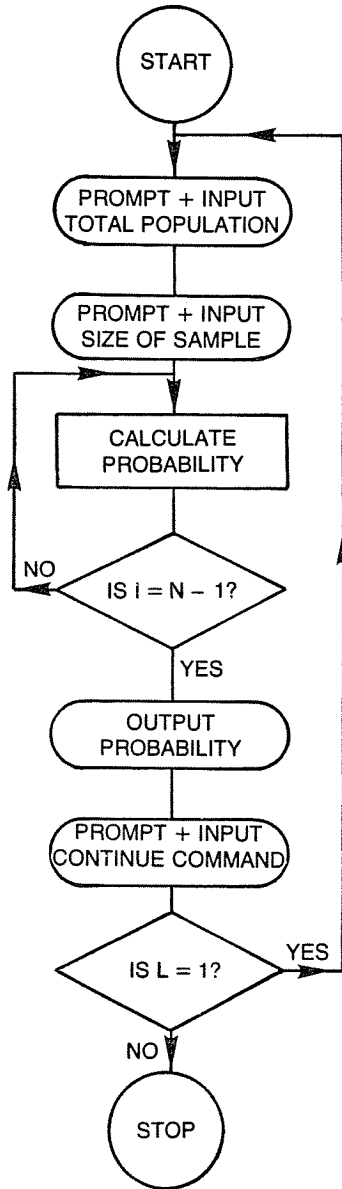
NO-REPETITIONS PROBABILITY

```
10    REM THIS PROGRAM COMPUTES THE "NO REP-
         ETITIONS"
20    REM PROBABILITY OF A SAMPLE
30    PRINT "***************************"
35    PRINT "TOTAL POPULATION"
40    INPUT M
50    PRINT "SIZE OF SAMPLE"
60    INPUT N
70    LET J = 1
80    FOR I = 1 TO N − 1
90    LET K = 1 − (I/M)
100   LET J = J*K
110   NEXT I
120   PRINT "PROBABILITY = ";J
130   PRINT
140   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
150   INPUT L
160   IF L = 1 THEN 30
170   STOP
180   END
```

NUMBER GUESS GAME

The object of this game is to guess in as few tries as possible the number chosen at random by the computer. (Hint: Use a binary search pattern.)

EXAMPLE

A RANDOM NUMBER HAS BEEN PICKED
TRY GUESSING IT, HINT THE NUMBER
IS FROM 1 TO 100
GOOD LUCK!!!
YOUR GUESS IS =
?
50
TOO HIGH, TRY AGAIN
YOUR GUESS IS =
?
25
TOO HIGH, TRY AGAIN
YOUR GUESS IS =
?
12
TOO LOW, TRY AGAIN
YOUR GUESS IS =
?
18
TOO LOW, TRY AGAIN
YOUR GUESS IS =
?
20
NOT BAD!!!-YOU GOT IT!!!
YOUR NUMBER OF TRIES WERE 5
IF YOU WANT TO TRY AGAIN TYPE 1
IF NOT TYPE 0
?
0
*END

NUMBER GUESS GAME

START

DERIVE
RANDOM NUMBER

PRINT
START MESSAGE

PROMPT + INPUT
GUESS

INCREMENT
GUESS COUNTER

IS X = R?

YES

PRINT
WINNING MESSAGE

OUTPUT
NUMBER OF
GUESSES

PROMPT + INPUT
CONTINUE
COMMAND

NO

PRINT
TOO HIGH
MESSAGE

IS X > R?

IS SC = 1?

YES

NO

PRINT
TOO LOW
MESSAGE

STOP
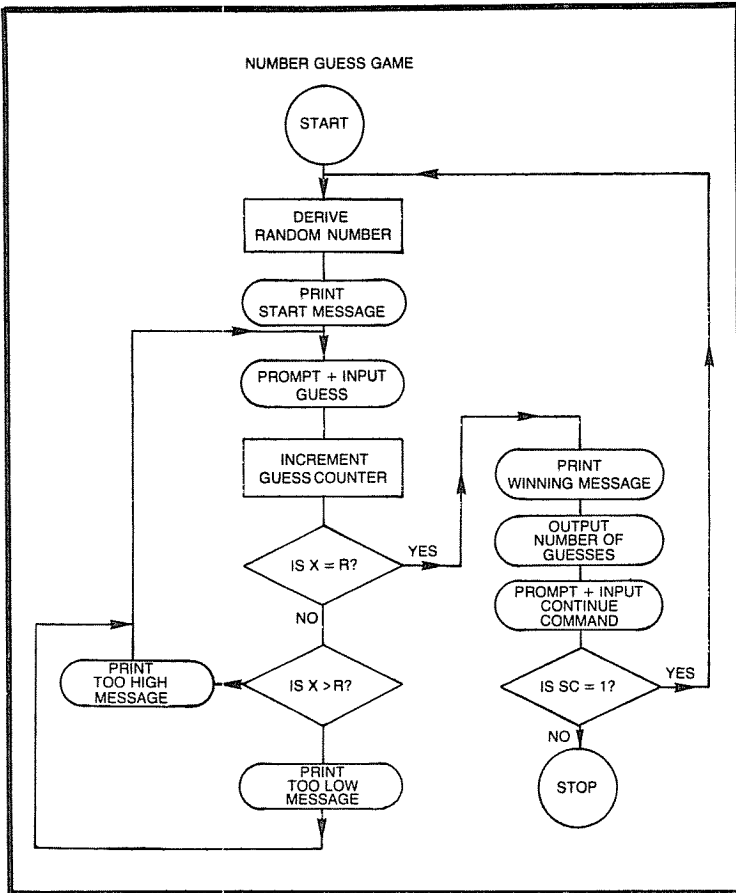
NUMBER GUESS GAME

```
10    REM THE COMPUTER CHOOSES A NUMBER FROM
20    REM 1 TO 100 AT RANDOM, THE OBJECT OF
30    REM THE GAME IS TO GUESS THE CHOSEN
40    REM NUMBER IN AS FEW GUESSES AS POSSIBLE
50    LET R = 1 + INT(100*RND)
60    LET Y = 0
70    PRINT "A RANDOM NUMBER HAS BEEN PICKED"
80    PRINT "TRY GUESSING IT, HINT THE NUMBER"
90    PRINT "IS FROM 1 TO 100"
100   PRINT "GOOD LUCK!!!"
110   PRINT
120   PRINT "YOUR GUESS IS = ";
130   INPUT X
```

```
140    LET Y = Y + 1
150    IF X = R THEN 180
160    IF X > R THEN 280
170    GOTO 300
180    PRINT "NOT BAD!!!-YOU GOT IT!!!"
190    PRINT "YOUR NUMBER OF TRIES WERE";Y
200    PRINT
210    PRINT "IF YOU WANT TO TRY AGAIN, TYPE 1"
220    PRINT "IF NOT TYPE 0"
230    INPUT L
240    IF L = 1 THEN 260
250    STOP
260    PRINT
270    GOTO 50
280    PRINT "TOO HIGH, TRY AGAIN"
290    GOTO 120
300    PRINT "TOO LOW, TRY AGAIN"
310    GOTO 120
320    END
```

ONE-ARM BANDIT

This computer program simulates the one-arm bandits that use three mechanical wheels. For detailed instructions, type yes to the instruction question in the program.

EXAMPLE

RUN
ARE INSTRUCTIONS REQUIRED
TYPE EITHER YES OR NO
?
NO
CHERRY CHERRY CHERRY
YOUR TOTAL EARNINGS ARE NOW $89
TO CONTINUE TYPE Y, IF NOT TYPE N
?
N
ONE-ARM BANDIT SAYS GOOD-BYE
*END
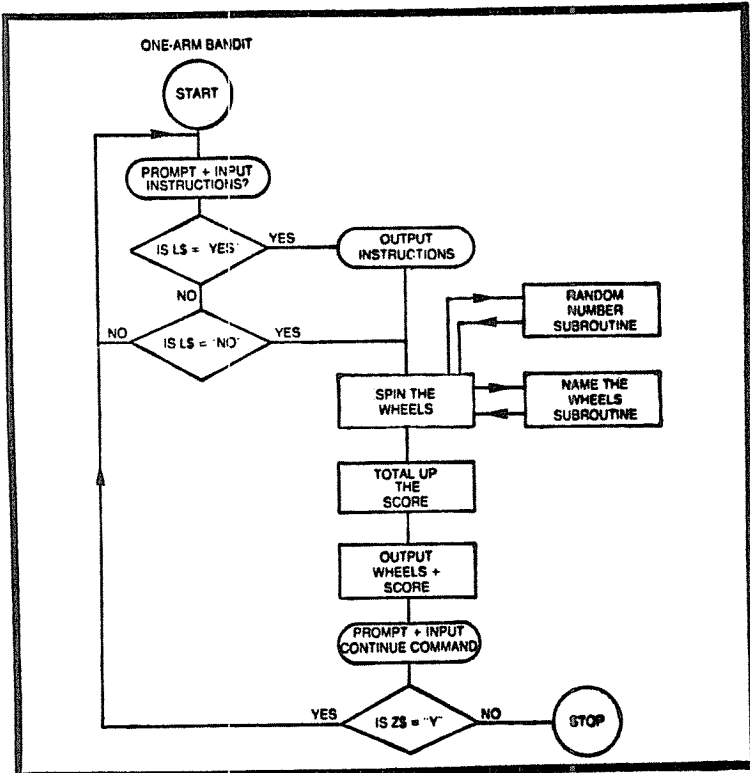
```
10    REM THIS PROGRAM SIMULATES THE MECHANI-
      CAL
20    REM THREE WHEEL ONE-ARM BANDIT
30    PRINT "ARE INSTRUCTIONS REQUIRED"
40    PRINT "TYPE EITHER YES OR NO"
50    INPUT L$
60    IF L$ = "YES" THEN 100
70    IF L$ = "NO" THEN 170
80    PRINT "INVALID COMMAND"
90    GOTO 30
100   PRINT "SCORING IS SIMPLE; 3 ORANGES, LEM-
      ONS OR"
110   PRINT "BANANAS EARN $10. 3 CHERRIES EARN
      $90."
120   PRINT "IF THE FIRST FRUIT IS AN APPLE YOU EARN
      $2"
130   PRINT "IF THE 1ST AND 2ND ARE APPLES YOU
      EARN $3"
140   PRINT "IF THE LAST FRUIT IS A CHERRY AND THE"
150   PRINT "OTHER TWO ARE THE SAME BUT NOT AP-
      PLES YOU"
160   PRINT "EARN $10. EACH TURN COSTS $1. GOOD-
      LUCK"
170   LET J = 0
180   PRINT
190   GOSUB 590
200   LET S1 = S
210   GOSUB 590
220   LET S2 = S
230   GOSUB 590
240   LET S3 = S
250   LET S = S1
260   GOSUB 610
270   LET S1$ = S$
280   LET S = S2
290   GOSUB 610
300   LET S2$ = S$
310   LET S = S3
320   GOSUB 610
330   LET S3$ = S$
340   IF S1$ = "CHERRY" THEN 380
350   IF S1$ = "APPLE" THEN 420
```

```
360    IF S1$ = S2$ THEN 440
370    GOTO 460
380    IF S1$ = S2$ THEN 400
390    GOTO 460
400    IF S2$ = S3$ THEN 480
410    GOTO 460
420    IF S1$ = S2$ THEN 500
430    GOTO 520
440    IS S2$ = S3$ THEN 540
450    IF S3$ = "CHERRY" THEN 540
460    LET J = J -- 1
470    GOTO 550
480    LET J = J + 89
490    GOTO 550
500    LET J = J + 2
510    GOTO 550
520    LET J = J + 1
530    GOTO 550
540    LET J = J + 9
550    PRINT
560    PRINT S1$;" ";S2$;" ";S3$
570    PRINT "YOUR TOTAL EARNINGS ARE NOW $";J
580    GOTO 720
590    LET S = 1 + INT(5*RND)
600    RETURN
610    ON S GOTO 620, 640, 660, 680, 700
620    LET S$ = "CHERRY"
630    GOTO 710
640    LET S$ = "APPLE"
650    GOTO 710
660    LET S$ = "LEMON"
670    GOTO 710
680    LET S$ = "ORANGE"
690    GOTO 710
700    LET S$ = "BANANA"
710    RETURN
720    PRINT
730    PRINT "TO CONTINUE TYPE Y, IF NOT TYPE N"
740    INPUT Z$
750    IF Z$ = "Y" THEN 780
760    PRINT "ONE-ARM BANDIT SAYS GOOD-BYE"
770    STOP
780    PRINT
790    GOTO 190
800    END
```

PERMUTATIONS AND COMBINATIONS

This program computes permutations and combinations, where N is the number of items available and R is the size of the groups under consideration.

FORMULAE

$$P = \frac{N!}{(N - R)!} \qquad\qquad C = \frac{N!}{R!(N - R)!}$$

EXAMPLE

ENTER N,R
?
12,5
PERMUTATIONS = 95040
COMBINATIONS = 792
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
ENTER N,R
?
23,21
PERMUTATIONS = 1.2926E22
COMBINATIONS = 253
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

PERMUTATIONS AND COMBINATIONS
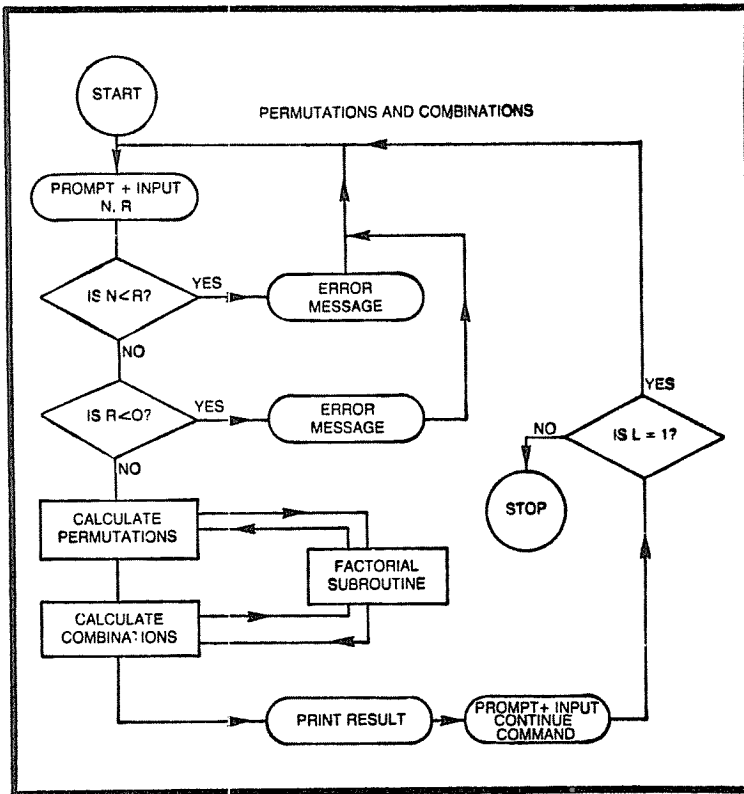
PERMUTATIONS AND COMBINATIONS

```
10    REM THIS PROGRAM COMPUTES PERMUTATIONS
20    REM AND COMBINATIONS, WHERE N = NUMBER
      OF
30    REM ITEMS AND R = SIZE OF GROUP SELECTED
40    PRINT "ENTER N,R";
50    INPUT N,R
60    IF N < R   THEN 270
70    IF R < 0   THEN 290
80    LET T = N
90    GOSUB 310
100   LET A = T
110   LET T = N – R
120   GOSUB 310
130   LET B = T
140   LET P = A/B
150   LET T = R
```

```
160    GOSUB 310
170    LET C = A/(T*B)
180    PRINT "PERMUTATIONS = ";P
190    PRINT "COMBINATIONS = ";C
200    PRINT
210    PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
220    INPUT L
230    IF L = 1 THEN 250
240    STOP
250    PRINT
260    GOTO 40
270    PRINT "N < R,INVALID INPUT"
280    GOTO 40
290    PRINT "R < 0,INVALID INPUT"
300    GOTO 40
310    IF T = 0 THEN 390
320    IF T = 1 THEN 390
330    LET J = 1
340    FOR I = 2 TO T
350    LET J = J*I
360    NEXT I
370    LET T = J
380    GOTO 400
390    LET T = 1
400    RETURN
410    END
```

PI-NETWORK IMPEDANCE MATCHING

Often between two resistive impedanced Z_1 and Z_2 a lossless network is desired. The computer expects the following information: $Z_1 - Z_2$, desired system Q and the operating frequency.

FORMULAE
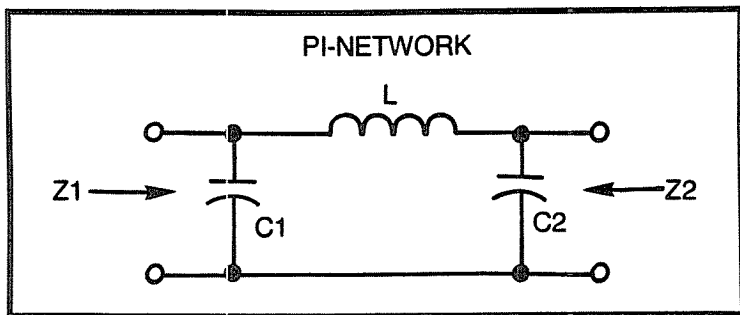
$Z_1 > Z_2$, $f = frequency$ and Q is desired system Q

$$C_1 = \frac{1}{2\pi f X_{C1}} \quad C_2 = \frac{1}{2\pi f X_{C2}}, \quad L = \frac{X_1}{2\pi f} \frac{Z_2}{Z_1}(Q^2 + 1) > 1$$

and where

$$X_{C1} = \frac{Z_1}{Q}, \quad X_{C2} = \frac{Z_2}{\left(\frac{Z_2}{Z_1}(Q^2 + 1) - 1\right)^{.5}}$$

$$X_L = \frac{QZ_1}{Q^2 + 1}\left(1 + \frac{Z_2}{QX_{C2}}\right)$$

EXAMPLE

ENTER 1ST IMPEDANCE
?
345
ENTER 2ND IMPEDANCE
?
300
ENTER DESIRED SYSTEM Q
?
20

146

ENTER OPERATING FREQUENCY
?
500
COMPONENTS OF PI-NETWORK
CAPACITOR 1 = 1.84527 E − 5
CAPACITOR 2 = 1.97846 E − 5
INDUCTOR = 1.05836 E − 2
TO CONTINUE TYPE 1, IF NOT 0
?
0
*END

```
              ┌─────────┐
              │  START  │
              └────┬────┘
                   │ ◄──────────────┐
          ╭────────┴────────╮       │
          │ PROMPT + INPUT  │       │
          │ SYSTEM VARIABLES│       │
          ╰────────┬────────╯       │
          ┌────────┴────────┐       │
          │    COMPUTE      │       │
          │   COMPONENTS    │       │
          │   OF NETWORK    │       │
          └────────┬────────┘       │
          ╭────────┴────────╮       │
          │    OUTPUT       │       │
          │ COMPONENT-VALUES│       │
          ╰────────┬────────╯       │
          ╭────────┴────────╮       │
          │ PROMPT + INPUT  │       │
          │   CONTINUE      │       │
          │   COMMAND       │       │
          ╰────────┬────────╯       │
               ╱───┴───╲   YES      │
              ╱ IS X = 1?╲──────────┘
               ╲───┬───╱
                   │ NO
              ┌────┴────┐
              │  STOP   │
              └─────────┘
```

PI-NETWORK IMPEDANCE MATCHING
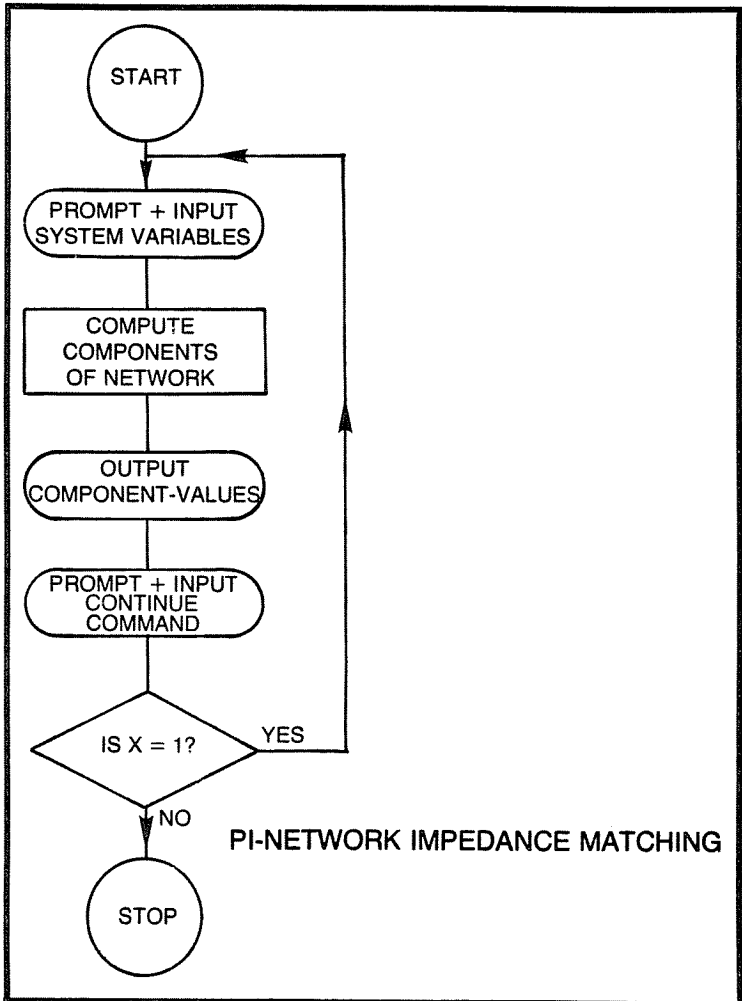
PI-NETWORK IMPEDANCE MATCHING

```
10    REM THIS PROGRAM COMPUTES THE COMPO-
         NENTS
20    REM OF A PI-NETWORK. TO MATCH TWO
30    REM IMPEDANCES
40    PRINT "ENTER 1ST IMPEDANCE"
50    INPUT Z1
60    PRINT "ENTER 2ND IMPEDANCE"
70    INPUT Z2
80    PRINT "ENTER DESIRED SYSTEM Q"
90    INPUT Q
100   PRINT "ENTER OPERATING FREQUENCY"
110   INPUT A
120   LET A = Z1/Q
130   LET C = ((Z2/Z1)*((Q↑2) + 1)) − 1
140   LET B = Z2/SQR(C)
150   LET E = (Z2/(Q*B)) + 1
160   LET D = E*((Q*Z1)/((Q↑2) + 1))
170   LET P = 6.28319
180   LET C1 = 1/(P*F*A)
190   LET C2 = 1/(P*F*B)
200   LET L = D/(P*F)
210   PRINT "COMPONENTS OF PI-NETWORK"
220   PRINT "CAPACITOR 1 = ";C1
230   PRINT "CAPACITOR 2 = ";C2
240   PRINT "INDUCTOR = ";L
250   PRINT
260   PRINT " TO CONTINUE TYPE 1, IF NOT 0"
270   INPUT X
280   IF X = 1 THEN 300
290   STOP
300   PRINT
310   GOTO 40
320   END
```

POINTS ON THE CIRCUMFERENCE

This program computes N equally spaced points on the circumference of a circle. Given radius and center of the circle, this program computes the rectangular coordinates of equally spaced points X_i, Y_i.

FORMULAE

$$X_{I+1} = X_0 + R\ Cos(\Theta + ZI)$$
$$Y_{I+1} = Y_0 + R\ Sin(\Theta + ZI)$$
$$where\ Z = 2\pi\ /N$$

POINTS ON A CIRCUMFERENCE
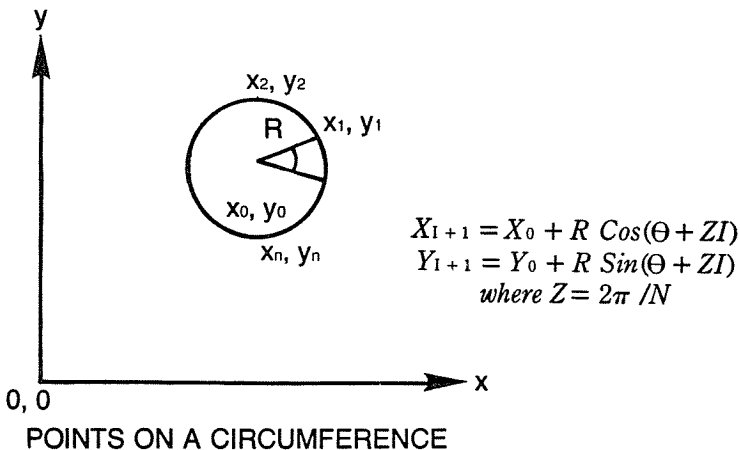
EXAMPLE

CENTER OF CIRCLE(X0,Y0) =
?
2,2
ANGLE IN DEGREES OF FIRST POINT =
?
90
NUMBER OF POINTS DESIRED =
?
2
RADIUS OF CIRCLE =
?
1
COORDINATES
POINT 1 X = 3 Y = 2
POINT 2 X = 2 Y= 1

POINTS ON THE CIRCUMFERENCE

```
        ┌─────────┐
        │  START  │
        └─────────┘
             │
    ┌─────────────────┐
    │ PROMPT + INPUT  │
    │  CIRCLE CENTER  │
    └─────────────────┘
             │
    ┌─────────────────┐
    │ PROMPT + INPUT  │
    │ ANGLE OF 1st POINT │
    └─────────────────┘
             │
    ┌─────────────────┐
    │ PROMPT + INPUT  │
    │   # OF POINTS   │
    └─────────────────┘
             │
    ┌─────────────────┐
    │ PROMPT + INPUT  │
    │  CIRCLE RADIUS  │
    └─────────────────┘
             │
    ┌─────────────────┐
    │      PRINT      │
    │  COORDINATES    │
    └─────────────────┘
             │
    ┌─────────────────┐
    │    COMPUTE      │
    │  COORDINATES    │
    └─────────────────┘
             │
    ┌─────────────────┐
    │    OUTPUT       │
    │  COORDINATES    │
    └─────────────────┘
             │
        ╱─────────╲
   NO  ╱  IS I = N – 1?  ╲
   ◄──╲           ╱
        ╲─────────╱
           │ YES
    ┌─────────────────┐
    │ PROMPT + INPUT  │
    │    CONTINUE     │
    │    COMMAND      │
    └─────────────────┘
             │
        ╱─────────╲
       ╱  IS L = 1?  ╲  YES
       ╲           ╱───►
        ╲─────────╱
           │ NO
        ┌─────────┐
        │  STOP   │
        └─────────┘
```

150
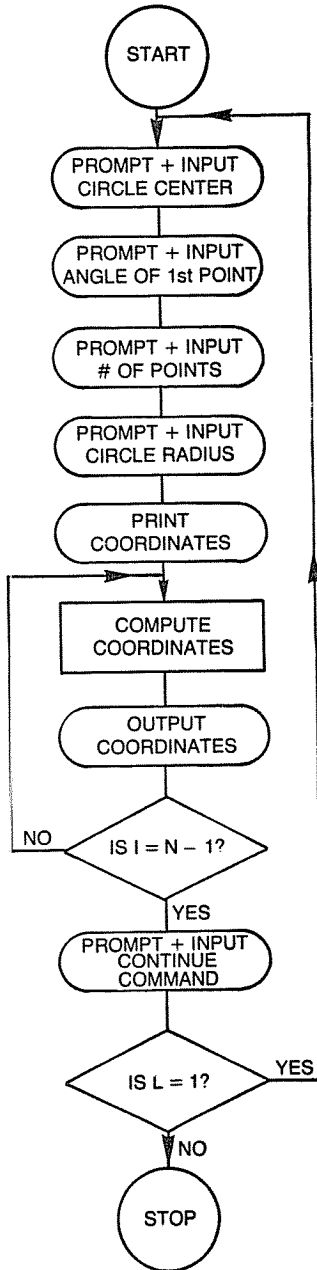
POINTS ON THE CIRCUMFERENCE

```
10   REM THIS PROGRAM COMPUTES N EQUALLY
     SPACED
20   REM POINTS ON THE CIRCUMFERENCE OF A
     CIRCLE
30   PRINT "CENTER OF CIRCLE (X0,Y0) = ";
40   INPUT X0,Y0
50   PRINT "ANGLE IN DEGREES OF 1ST POINT = ";
60   INPUT W
70   LET W = (W*3.14159)/180
80   PRINT "NUMBER OF POINTS DESIRED = ";
90   INPUT N
100  PRINT "RADIUS OF A CIRCLE = ";
110  INPUT R
120  LET Z = 6.28319/N
130  PRINT
140  PRINT "COORDINATES"
150  FOR I = 0 TO N − 1
160  LET X = X0 + (R*COS(W + Z*I))
170  LET Y = Y0 + (R*SIN(W + (Z*I)))
180  LET P = I + 1
190  PRINT "POINT;";P,"X = ";X,"Y = ";Y
200  NEXT I
210  PRINT
220  PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
230  INPUT L
240  IF L = 1 THEN 260
250  STOP
260  PRINT
270  GOTO 30
280  END
```

POLAR TO RECTANGULAR CONVERSION

This program converts given polar coordinates into rectangular coordinates.

FORMULAE

POLAR TO RECTANGULAR

$$X = R\ Cos\ \Theta$$
$$Y = R\ Sin\ \Theta$$
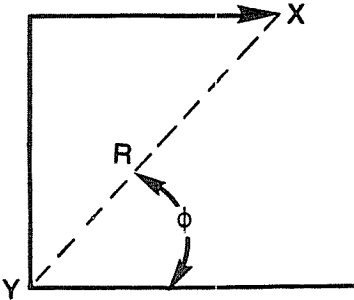
EXAMPLE

POLAR COORDINATES
ANGLE W IN DEGREES =
?
45
MAGNITUDE R =
?
5
RECTANGULAR COORDINATES
X = 3.53553 Y = 3.53553
TYPE 1 TO CONTINUE, 0 TO STOP
?
POLAR COORDINATES
ANGLE W IN DEGREES =
?
20
MAGNITUDE R =
?
1
RECTANGULAR COORDINATES
X = .939693 Y = .34202
TYPE 1 TO CONTINUE, 0 STOP
?
0
*END

POLAR TO RECTANGULAR CONVERSION

START

PRINT
POLAR
COORDINATES

PROMPT + INPUT
ANGLE

PROMPT + INPUT
MAGNITUDE

COMPUTE
P→R

PRINT
RECT. COORDINATES

OUTPUT
X, Y

PROMPT + INPUT
CONTINUE
COMMAND

IS L = 1?   YES

NO

STOP
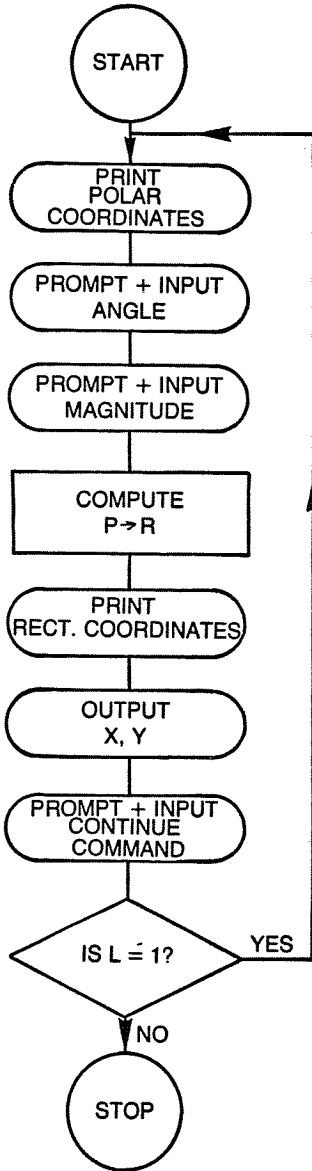
POLAR TO RECTANGULAR CONVERSION

```
10    REM THIS PROGRAM CONVERTS GIVEN POLAR
         COORDINATES
20    REM INTO RECTANGULAR COORDINATES
30    PRINT "POLAR COORDINATES"
40    PRINT "ANGLE W IN DEGREES = ";
50    INPUT W
60    LET W = (W*3.14159)/180
70    PRINT "MAGNITUDE R = ";
80    INPUT R
90    LET X = R*COS(W)
100   LET Y = R*SIN(W)
110   PRINT "RECTANGULAR COORDINATES"
120   PRINT "X = ";X,"Y = ";Y
130   PRINT
140   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
150   INPUT L
160   IF L = 1 THEN 180
170   STOP
180   PRINT
190   GOTO 30
200   END
```

PRIME TEST

This program tests a given number to see whether or not it is prime. If so, it is thus indicated and if not, the smallest factor returned. The program will continue to cycle until a zero is entered as a test number.

EXAMPLES

ENTER THE TEST NUMBER,ZERO TO STOP
?
45
45 IS NOT A PRIME 3 IS THE SMALLEST FACTOR
ENTER THE TEST NUMBER,ZERO TO STOP
?
120078
120078 IS NOT A PRIME 2 IS THE SMALLEST FACTOR
ENTER THE TEST NUMBER,ZERO TO STOP
?
121
121 IS NOT A PRIME 11 IS THE SMALLEST FACTOR
ENTER THE TEST NUMBER,ZERO TO STOP
?
179
179 IS A PRIME
ENTER THE TEST NUMBER,ZERO TO STOP
?
0
GOOD-BYE FROM THE PRIME TESTER
*END
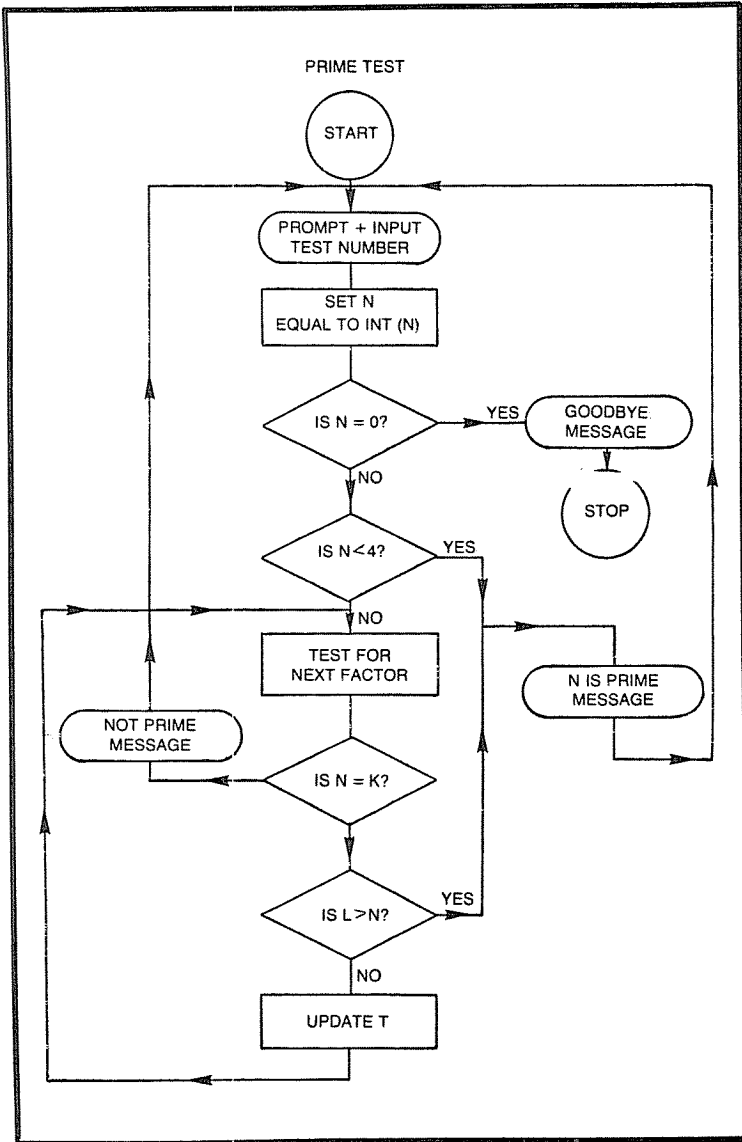
PRIME TESTER

10   REM THIS PROGRAM TESTS IF A NUMBER IS PRIME

20   REM IT CONTINUES TO CYCLE UNTIL ZERO IS EN-
     TERED

30   PRINT "ENTER THE TEST NUMBER, ZERO TO
     STOP"

```
40    INPUT N
50    LET N = INPUT (N)
60    IF N = 0 THEN 220
70    IF N < 4 THEN 180
80    LET I = 0
90    LET T = 2
100   LET J = INT(N/T)
110   LET K = J*T
120   IF N = K THEN 200
130   LET I = I + 1
140   LET L = T*T
150   IF L > N THEN 180
160   LET T = (I*2) + 1
170   GOTO 100
180   PRINT N; "IS A PRIME"
190   GOTO 30
200   PRINT N; "IS NOT A PRIME";T;"IS THE SMALLEST
         FACTOR"
210   GOTO 30
220   PRINT "GOOD-BYE FROM THE PRIME TESTER"
230   END
```

QUADRATIC EQUATIONS

This program solves for x in a quadratic equation where a, b and c are given. Both real and complex roots are found.

FORMULAE

$$ax^2 + bx + c = \phi$$

$$x_1, x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$D = (b^2 - 4ac)/4a^2$$

$$D \geq \phi \quad \text{roots are real} \quad D < \phi \quad \text{roots are complex}$$

$$D \geq \phi$$

$$\text{IF} \quad -\frac{b}{2a} \geq \phi \quad X_1 = -\frac{b}{2a} + \sqrt{D}$$

$$\text{IF} - \frac{b}{2a} < \phi \quad X_1 = -\frac{b}{2a} - \sqrt{D}$$

$$X_2 = \frac{C}{X_1 a}$$

$$D < O$$

$$U + Vi = \frac{-b}{2a} \pm \frac{\sqrt{4ac - b^2}}{2a} i$$

EXAMPLES

ENTER VALUES FOR A,B AND C
?
1,1,0
ROOTS ARE REAL
1ST ROOT $=-1$
2ND ROOT $= 0$
****

TYPE 1 TO CONTINUE, 0 TO STOP
?
1
ENTER VALUES FOR A,B AND C
?
10,10,10
ROOTS ARE COMPLEX
REAL PART = − 50
IMAGINARY PART = 86.6025
****
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

```
                        ┌─────────┐
                        │  START  │
                        └─────────┘
                             │
                    ┌────────────────┐
           ┌───────▶│ PROMPT + INPUT │      QUADRATIC EQUATIONS
           │        │    A, B, C     │
           │        └────────────────┘
           │                │
           │        ┌────────────────┐
           │        │    COMPUTE     │
           │        │       D        │
           │        └────────────────┘
           │                │
           │            ╱IS D70╲ ──YES──▶ ┌──────────┐
           │            ╲      ╱          │ COMPUTE  │
           │               NO             │    E     │
           │                │             └──────────┘
           │        ┌────────────────┐         │
           │        │    COMPUTE     │      ╱IS E >= 0╲ ──YES──▶ ┌──────────┐
           │        │     X, Y       │      ╲         ╱          │COMPUTE Z │
           │        └────────────────┘          NO               │ ( + √D)  │
           │        ┌────────────────┐           │               └──────────┘
           │        │     PRINT      │    ┌────────────┐               │
           │        │   ROOTS ARE    │    │ COMPUTE Z  │               │
           │        │    COMPLES     │    │  ( − √D)   │               │
           │        └────────────────┘    └────────────┘               │
           │        ┌────────────────┐          │◀────────────────────┘
           │        │    OUTPUT      │    ┌────────────┐
           │        │ REAL-IMAGINARY │    │  COMPUTE   │
           │        └────────────────┘    │     W      │
           │        ┌────────────────┐    └────────────┘
           │        │ PROMPT + INPUT │          │
           │        │   CONTINUE     │    ┌────────────┐
           │        │   COMMAND      │    │   PRINT    │
           │        └────────────────┘    │ROOTS ARE REAL│
           │                │             └────────────┘
           │  YES     ╱IS L = 1?╲              │
           └────────  ╲        ╱        ┌────────────┐
                          NO            │   OUTPUT   │
                           │            │   ROOTS    │
                      ┌─────────┐       └────────────┘
                      │  STOP   │            │
                      └─────────┘          (   )
```
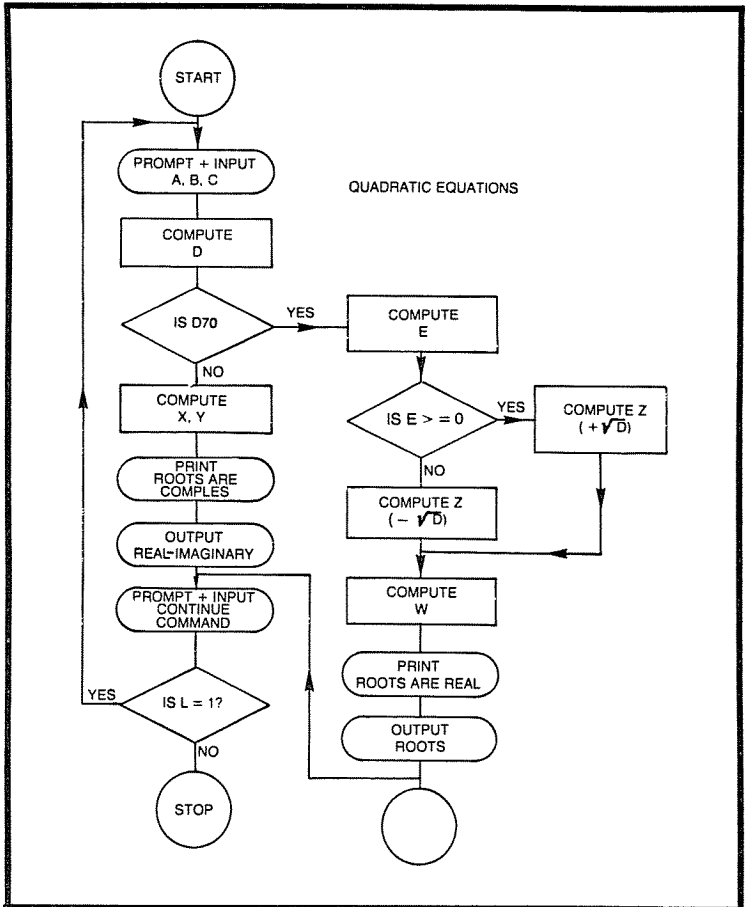
159

QUADRATIC EQUATION

```
10    REM THIS PROGRAM COMPUTES THE SOLUTION
      TO A
20    REM QUADRATIC EQUATION
30    PRINT "ENTER VALUES FOR A, B AND C"
40    INPUT A,B,C
50    LET D = ((B↑2) – (4*A*C))/(4*A↑2)
60    IF D > = 0 THEN 130
70    LET X = B/(2*A)
80    LET Y = (SQR((4*A*C) – B 2))/(2*A)
90    PRINT "ROOTS ARE COMPLEX"
100   PRINT "REAL PART = ";X
110   PRINT "IMAGINARY PART = ";Y
120   GOTO 220
130   LET E = – B/(2*A)
140   IF E > = 0 THEN 170
150   LET Z = E – SQR(D)
160   GOTO 180
170   LET Z = E + SQR(D)
180   LET W = C/(Z*A)
190   PRINT "ROOTS ARE REAL"
200   PRINT 1ST ROOT = ";Z
210   PRINT "2ND ROOT = ";W
220   PRINT "*************************"
230   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
240   INPUT L
250   IF L = 1 THEN 270
260   STOP
270   PRINT
280   GOTO 30
290   END
```

RECTANGULAR TO POLAR CONVERSION

This program computes rectangular coordinates that are supplied by the user into polar coordinates.

FORMULAE

$$\phi = TAN^{-1}\frac{Y}{X} \quad R = \sqrt{X^2 + Y^2}$$

EXAMPLES

X =
?
10
Y =
?
10
POLAR COORDINATES
ANGLE IN DEGREES = 45.0000
MAGNITUDE = 14.1421
****
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
X =
?
34
Y =
?
32
POLAR COORDINATES
ANGLE IN DEGREES = 43.2643
MAGNITUDE = 46.6904
****
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

RECTANGULAR TO POLAR CONVERSION

START

PROMPT + INPUT RECTANGULAR COOR.

RECTANGULAR TO POLAR CONVERSION

IS X + Y = 0? — YES → LET 0 = O / R + O

NO

IS X + 0? — YES → IS Y + O? — YES → LET 0 = 90 / R = Y

NO

LET 0 = – 90 / R = ABS (Y)

IS Y + O? — YES → IS X + O? — YES → LET 0 + O / R + X

NO

CALCULATE POLAR COORDINATES

LET 0 + 180 / R +ABS (X)

OUTPUT POLAR COORDINATES

PROMPT + INPUT CONTINUE COMMAND

IS L = 1? — YES

NO

STOP
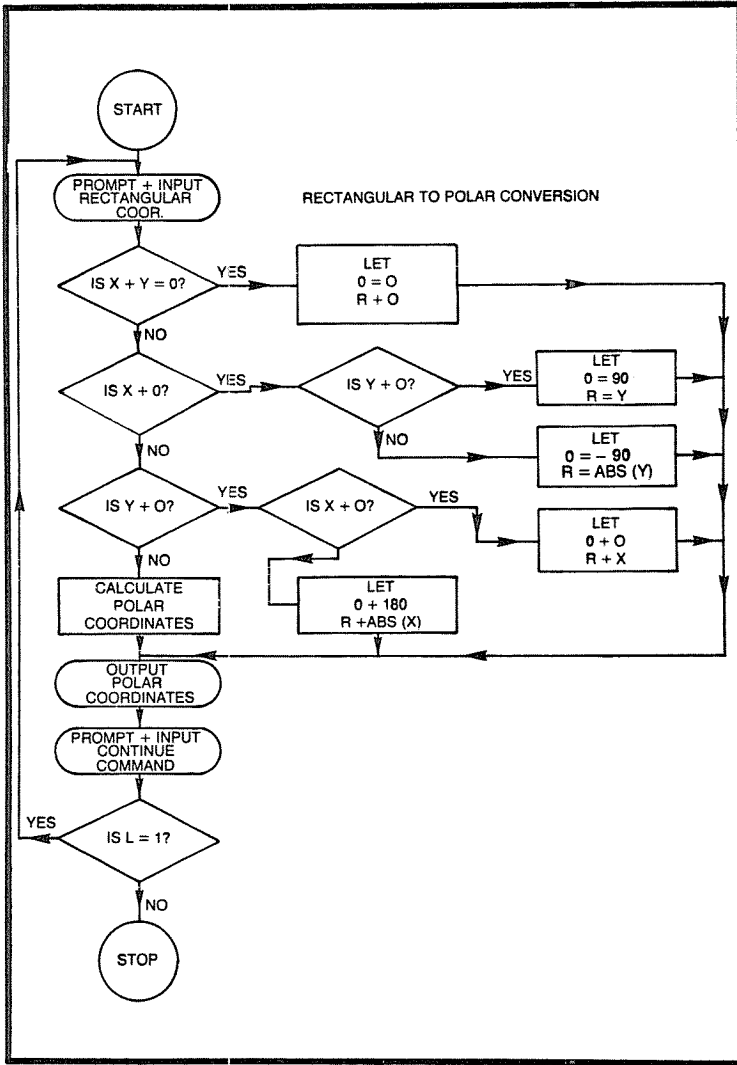
RECTANGULAR TO POLAR CONVERSION

10   REM THIS PROGRAM CONVERTS GIVEN RECTAN-GULAR

20   REM COORDINATES INTO POLAR COORDINATES

30   PRINT "RECTANGULAR COORDINATES"

40   PRINT "X = "

50   INPUT X

60   PRINT "Y="

```
70    INPUT Y
80    IF X + Y = 0 THEN 150
90    IF X = 0 THEN 180
100   IF Y = 0 THEN 250
110   LET W = ATN(Y/X)
120   LET W = (W*180)/3.14159
130   LET R = SQR(X↑2 + Y↑2)
140   GOTO 310
150   LET W = 0
160   LET R = 0
170   GOTO 310
180   IF Y> 0 THEN 220
190   LET W = - 90
200   LET R = ABS(Y)
210   GOTO 310
220   LET W = 90
230   LET R = Y
240   GOTO 310
250   X > 0 THEN 290
260   LET W = 180
270   LET R = ABS (X)
280   GOTO 310
290   LET W = 0
300   LET R = X
310   PRINT "POLAR COORDINATES"
320   PRINT "ANGLE IN DEGREES = "; W
330   PRINT "MAGNITUDE = "; R
340   PRINT "************************"
350   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
360   INPUT L
370   IF L = 1 THEN 390
380   STOP
390   PRINT
400   GOTO 40
410   END
```

RECTANGULAR SECTIONS

This program computes various parameters: moment of inertia, polar moment of inertia, and area of section connected with a rectangular section.

FORMULAE

RECTANGULAR SECTION
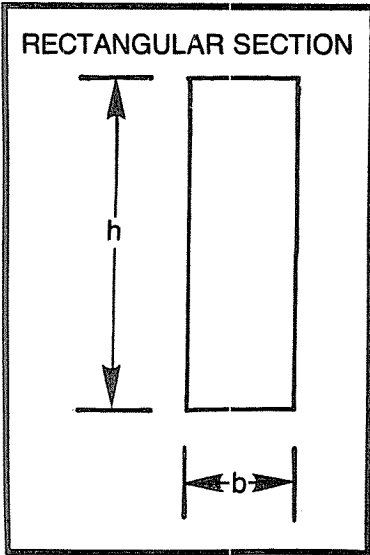
$$I = \frac{bh^3}{12} \quad \begin{array}{l} I \text{ and } J \text{ is in}(in^4) \\ A \text{ is in } (in^2) \end{array}$$

$$J = \frac{bh(b^2 + h^2)}{12}$$

$$A = bH$$

EXAMPLE

BASE =
?
3
HEIGHT =
?
5
MOMENT OF INERTIA = 31.25
POLAR MOMENT OF INERTIA = 42.50
AREA OF SECTION = 15
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

RECTANGULAR SECTIONS
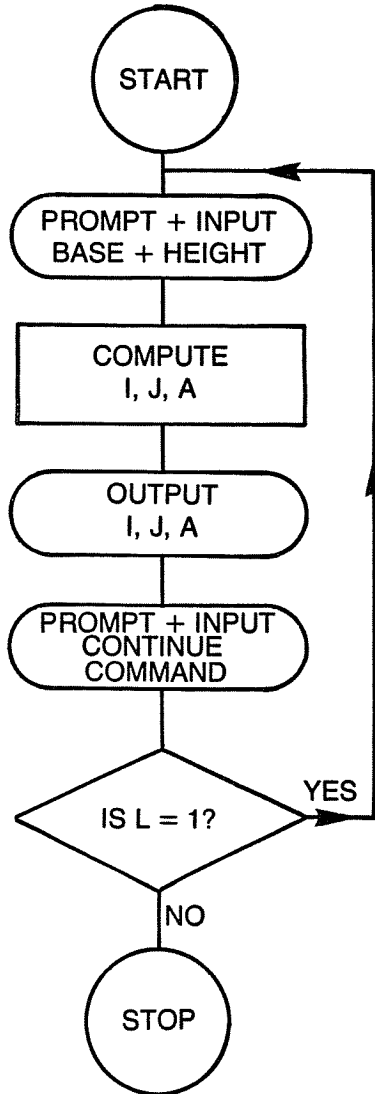
RECTANGULAR SECTIONS

```
10    REM THIS PROGRAM COMPUTES THE VARIOUS
      PARAMETERS
20    REM CONNECTED WITH A RECTANGULAR SEC-
      TION
30    PRINT "BASE = ";
40    INPUT B
50    PRINT "HEIGHT = "
60    INPUT H
70    LET I = (B*(H↑3))/12
80    LET J = (B*H*(B↑2 + H↑2))/12
90    LET A = B*H
100   PRINT "MOMENT OF INERTIA = ";I
110   PRINT "POLAR MOMENT OF INERTIA = ";J
120   PRINT "AREA OF SECTION = ";A
130   PRINT
140   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
150   INPUT L
160   IF L = 1 THEN 180
170   STOP
180   PRINT
190   GOTO 30
200   END
```

RESISTIVE ATTENUATOR DESIGN

This program computes the required three resistors to form either a Pi- or T-type resistive attenuator. This type of attenuator allows the user to choose a loss other than that of minimum.

FORMULAE
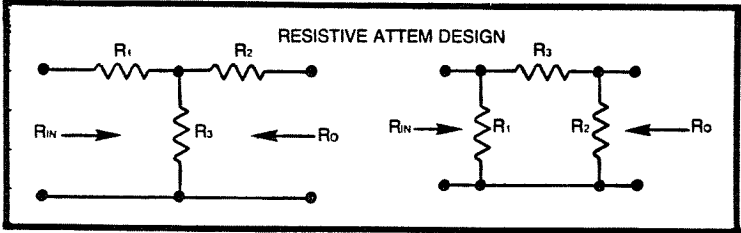
$R_{IN} > R_O$, and $N =$ desired loss $\geq$ minimum loss

$$minimum\ loss = 1\phi log\left(\sqrt{\frac{R_{IN}}{R_O}} + \sqrt{\frac{R_{IN}}{R_O} - 1}\right)^2$$

T-type

$$R_3 = \frac{2\sqrt{N\ R_{IN}R_O}}{N-1}$$

$$R_1 = R_{IN}\left(\frac{N+1}{N-1}\right) - R_3$$

$$R_2 = R_O\left(\frac{N+1}{N-1}\right) - R_3$$

π-type

$$R_3 = \frac{1}{2}(N-1)\left(\frac{R_{IN}R_O}{N}\right)^{1/2}$$

$$R_1 = \frac{1}{\dfrac{1}{R_{IN}}\left(\dfrac{N+1}{N-1}\right) - \dfrac{1}{R_3}}$$

$$R_2 = \frac{1}{\dfrac{1}{R_O}\left(\dfrac{N+1}{N-1}\right) - \dfrac{1}{R_3}}$$

EXAMPLE

INPUT RESISTANCE R(IN) =
?
500
OUTPUT RESISTANCE R(0) =
?
100
MINIMUM SYSTEM LOSS IN DECIBELS = 12.54
ENTER DESIRED LOSS IN DECIBELS
?
20
R(IN) = 500 R(0) = 100
DESIRED LOSS = 20
T ATTENUATOR
RESISTOR 1 = 464.9
RESISTOR 2 = 56.85
RESISTOR 3 = 45.17
PI ATTENUATOR
RESISTOR 1 = 879.6
RESISTOR 2 = 107.5
RESISTOR 3 = 1107
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

RESISTIVE ATTENUATOR DESIGN

START

PROMPT + INPUT
R(IN) + R(O)

COMPUTE
MINIMUM
LOSS

OUTPUT
MINIMUM LOSS

PROMPT + INPUT
DESIRED LOSS

COMPUTE
COMPONENTS

OUTPUT
LOSS

OUTPUT
T-TYPE
COMPONENTS

OUTPUT
π-TYPE
COMPONENTS

PROMPT + INPUT
CONTINUE COMMAND

IS Q = 1?   YES

NO

STOP
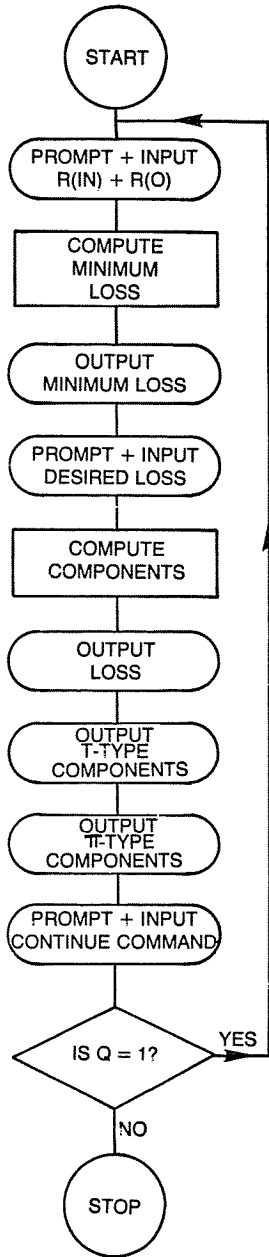
```
10    REM THIS PROGRAM COMPUTES THE COMPO-
      NENTS
20    REM REQUIRED FOR A PI OR T TYPE
30    REM RESISTIVE ATTENUATOR
40    PRINT "INPUT RESISTANCE R(IN) = ";
50    INPUT X
60    PRINT "OUTPUT RESISTANCE R(0) = ";
70    INPUT Y
80    LET Z = X/Y
90    LET Q = (SQR(Z) + SQR(Z - 1))↑2
100   LET M = 10*(LOG(Q)/LOG(10))
110   PRINT "MINIMUM SYSTEM LOSS IN DECIBELS
      = ";M
120   PRINT "ENTER DESIRED LOSS IN DECIBELS";
130   INPUT L
140   LET N = 10 (L/10)
150   LET W = N - 1
160   LET U = N + 1
170   LET A = 2*(SQR(X*Y*N))
180   LET B = (X*(U/W)) - A
190   LET C = (Y*(U/W)) - A
200   LET D = (W*SQR((X*Y)/N))/2
210   LET E = 1/((U/W)/X) - (1/D)
220   LET F = 1/((U/W)/Y) - (1/D)
230   PRINT "R(IN) = ";X,"R(0) = ";Y
240   PRINT "DESIRED LOSS = ";L
250   PRINT
260   PRINT "T ATTENUATOR"
270   PRINT "RESISTOR 1 = ";B
280   PRINT "RESISTOR 2 = ";C
290   PRINT "RESISTOR 3 = ";A
300   PRINT
310   PRINT "PI ATTENUATOR"
320   PRINT "RESISTOR 1 = ";E
330   PRINT "RESISTOR 2 = ";F·
340   PRINT "RESISTOR 3 = ";D
350   PRINT
360   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
370   INPUT Q
380   IF Q = 1 THEN 400
390   STOP
400   PRINT
410   GOTO 40
420   END
```

SIMULTANEOUS EQUATIONS IN TWO UNKNOWNS

The user supplies the components of two AX + BY = C type equations, also giving the computer the opportunity to state if the solution is impossible.

FORMULAE

$$X = \frac{ED - BF}{AD - BC} = \frac{\begin{vmatrix} E & B \\ F & D \end{vmatrix}}{\begin{vmatrix} A & B \\ C & D \end{vmatrix}} \quad Y = \frac{AF - EC}{AD - BC} = \frac{\begin{vmatrix} A & E \\ C & F \end{vmatrix}}{\begin{vmatrix} A & B \\ C & D \end{vmatrix}}$$

EXAMPLE

AX + BY = E
CX + DY = F
ENTER PARAMETERS, A,B,C,D,E,F
?
10,20,45,23,56,78
SOLUTION X = 9.6 Y = − 2.55
****
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
AX + BY = E
CX + DY = F
ENTER PARAMETERS A,B,C,D,E,F
10,10,789,2,2,6
NO SOLUTION, OR NO UNIQUE SOLUTION EXISTS
****
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

SIMULTANEOUS EQUATIONS IN TWO UNKNOWNS

```
                    ┌─────────┐
                    │  START  │
                    └─────────┘
                         │
              ┌──────────────────────┐
              │     PROMPT +         │
              │   INPUT PARAMETER    │
              └──────────────────────┘
                         │
              ┌──────────────────────┐
              │     CALCULATE        │
              │         M            │
              └──────────────────────┘
                         │
                   ╱─────────╲        YES
                  ╱  IS M = 0? ╲──────────────┐
                   ╲─────────╱                │
                         │ NO                 │
              ┌──────────────────┐    ┌───────────────┐
              │    CALCULATE     │    │    OUTPUT     │
              │    X AND Y       │    │  NO SOLUTION  │
              └──────────────────┘    │    MESSAGE    │
                         │            └───────────────┘
              ┌──────────────────┐            │
              │     OUTPUT       │            │
              │    X AND  Y      │            │
              └──────────────────┘            │
                         │                    │
              ┌──────────────────┐            │
              │ OUTPUT MESSAGE   │            │
              │       +          │◄───────────┘
              │ CONTINUE INPUT   │
              └──────────────────┘
                         │
                   ╱─────────╲        YES
                  ╱   L = 1?   ╲──────────────►
                   ╲─────────╱
                         │ NO
                    ┌─────────┐
                    │  STOP   │
                    └─────────┘
```
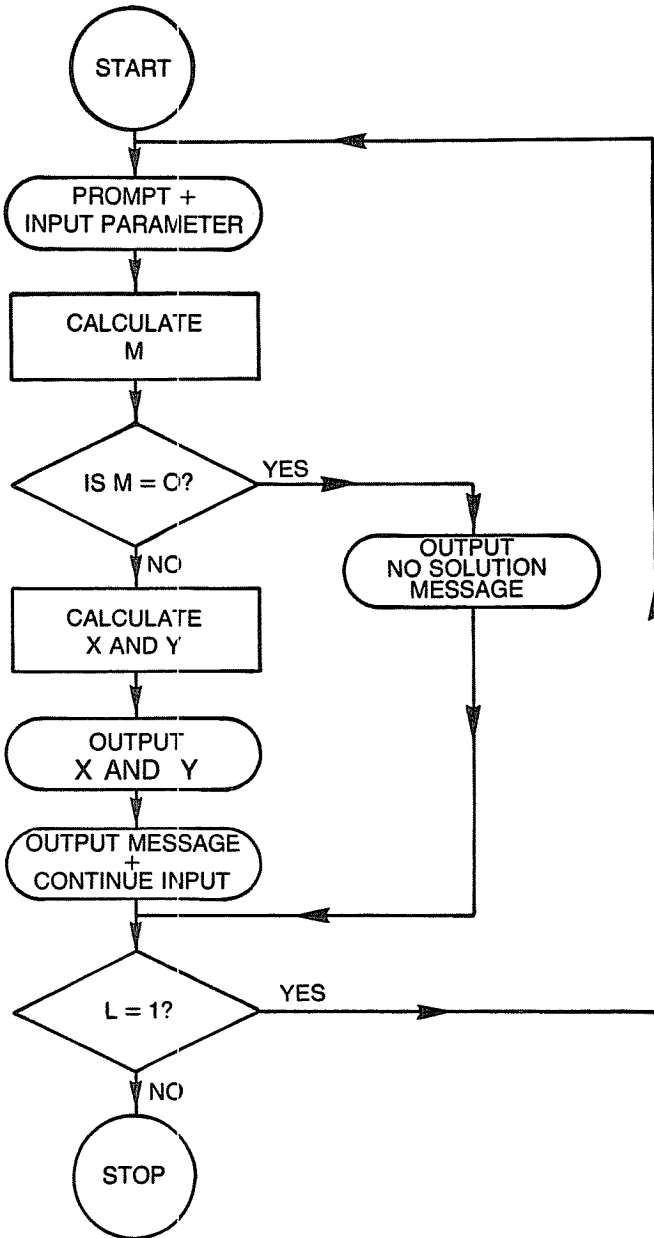
SIMULTANEOUS EQUATIONS IN TWO UNKNOWNS

```
10    REM THIS PROGRAM COMPUTES SIMULTANEOUS
      EQUATIONS
20    REM IN TWO UNKNOWNS
30    PRINT "AX + BY = E"
40    PRINT "CX + DY = F"
50    PRINT "ENTER PARAMETERS A,B,C,D,E,F"
60    INPUT A,B,C,D,E,F
70    LET M = (A*D) − (B*C)
80    IF M = 0 THEN 140
90    LET X = ((E*D) − (B*F))/M
100   LET Y = ((A*F) − (E*C))/M
110   PRINT "SOLUTION","X = ";X,"Y = ";Y
120   PRINT "*********************"
130   GOTO 160
140   PRINT "NO SOLUTION, OR NO UNIQUE SOLUTION
      EXISTS"
150   PRINT "********************"
160   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
170   INPUT L
180   IF L = 1 THEN 200
190   STOP
200   PRINT
210   GOTO 30
220   END
```

SIMULTANEOUS EQUATIONS IN THREE UNKNOWNS

The computer solves a system of three equations in three unknowns with the parameters supplied by the user.

EXAMPLE

SOLVE FOR 3 EQUATIONS IN 3 UNKNOWNS
OF THE TYPE AX + BY + CZ =D
ENTER FIRST EQUATION(A,B,C,D)
?
1,4,6,3
ENTER SECOND EQUATION(A,B,C,D)
?
− 3,8,0, − 2
ENTER THIRD EQUATION(A,B,C,D)
?
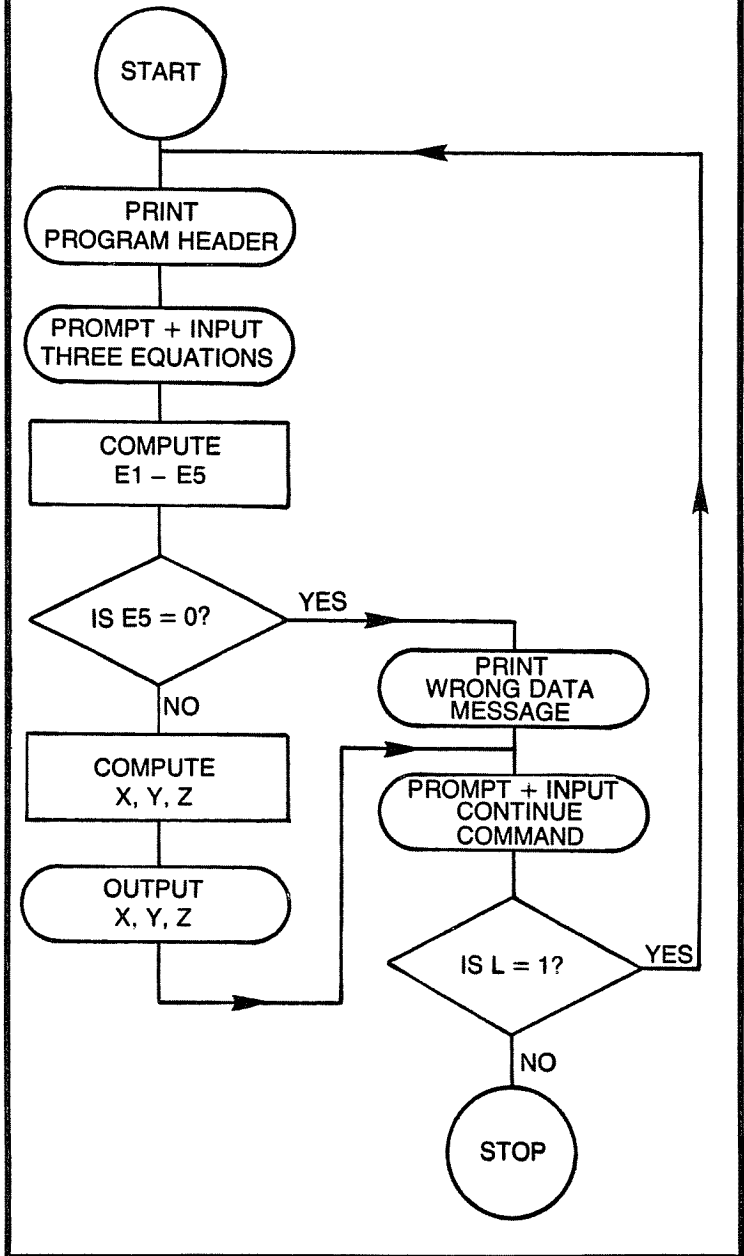4, − 5,2,6
SOLUTION
X = 2.65516 Y = .741935 Z = − .435483
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

SIMULTANEOUS EQUATIONS IN THREE UNKNOWNS

```
                          ┌─────────┐
                          │  START  │
                          └────┬────┘
                               │            ┌──────────────────◄───┐
                               ▼            ▼                       │
                     ╭───────────────╮                             │
                     │     PRINT     │                             │
                     │ PROGRAM HEADER│                             │
                     ╰───────┬───────╯                             │
                     ╭───────────────╮                            │
                     │ PROMPT + INPUT│                            │
                     │THREE EQUATIONS│                            │
                     ╰───────┬───────╯                            │
                     ┌───────────────┐                           │
                     │    COMPUTE    │                           │
                     │    E1 − E5    │                           │
                     └───────┬───────┘                           │
                             │                                   │
                           ◇───◇   YES                           │
                     ◇  IS E5 = 0? ◇──────────►──┐              │
                           ◇───◇                  │              │
                             │ NO                 ▼              │
                     ┌───────────────┐   ╭──────────────╮       │
                     │    COMPUTE    │   │    PRINT     │       │
                     │    X, Y, Z    │   │  WRONG DATA  │       │
                     └───────┬───────┘   │   MESSAGE    │       │
                             │           ╰──────┬───────╯       │
                     ╭───────────────╮   ╭──────────────╮       │
                     │    OUTPUT     │   │PROMPT + INPUT│       │
                     │    X, Y, Z    │   │   CONTINUE   │       │
                     ╰───────────────╯   │   COMMAND    │       │
                             │           ╰──────┬───────╯       │
                             └──────►──┐        │               │
                                       ▼      ◇───◇             │
                                           ◇IS L = 1?◇──YES─────┘
                                             ◇───◇
                                               │ NO
                                          ┌─────────┐
                                          │  STOP   │
                                          └─────────┘
```

SIMULTANEOUS EQUATIONS IN THREE UNKNOWNS

```
10    REM THIS PROGRAM COMPUTES THE SOLUTION
         TO
20    REM A SYSTEM OF 3 EQUATIONS IN 3 UNKNOWNS
30    PRINT "SOLVE FOR 3 EQUATIONS IN 3 UNKNOWNS"
40    PRINT "OF THE TYPE AX + BY + CZ = D"
50    PRINT "ENTER 1ST EQUATION (A,B,C,D)"
60    INPUT A1,B1,C1,D1
70    PRINT "ENTER 2ND EQUATION (A,B,C,D)"
80    INPUT A2,B2,C2,D2
90    PRINT "ENTER 3RD EQUATION (A,B,C,D)
100   INPUT A3,B3,C3,D3
110   LET E1 = ((B1*A2)/A1) − B2
120   LET E2 = ((C1*A2)/A1) − C2
130   LET E3 = ((B1*A3)/A1) − B3
140   LET E4 = ((C1*A3)/A1) − C3
150   LET E5 = ((E1*E4) − (E2*E3))
160   IF E5 = 0 THEN 250
170   LET E6 = ((D1*A2)/A1) − D2
180   LET E7 = ((D1*A3)/A1) − D3
190   LET Y = ((E6*E4) − (E2*E7))/E5
200   LET Z = ((E1*E7) − (E6*E3))/E5
210   LET X = (D1/A1) − ((B1/A1*Y) − ((C1/A1)*Z)
220   PRINT "SOLUTION"
230   PRINT "X = ";X,"Y = ";Y,"Z = ";Z
240   GOTO 260
250   PRINT "INSUFFICIENT OR ERRONEOUS DATA EN-
         TERED"
260   PRINT
270   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
280   INPUT L
290   IF L = 1 THEN 310
300   STOP
310   PRINT
320   GOTO 30
330   END
```

SPACE WARS (1)

The game of Space Wars, as simulated by this program, is a battle between two ships, one the enemy, the other the player. The player has the following options: rotate the ship, move the ship, fire the laser cannon, fire the laser, or self-destruct. The object of this game is to destroy the enemy. The enemy, also being armed with a laser cannon and a laser, fires on you, so be careful.

EXAMPLE

RUN
ARE INSTRUCTIONS REQUIRED? TYPE EITHER
YES OR NO
?
YES
THERE ARE 5 EXECUTIVE COMMANDS; TURN THE
SHIP(1), MOVE(2), FIRE LASER CANNON(3),
FIRE LASER(4) AND SELF-DESTRUCT(5)
THE CANNON MUST BE FIRED WITHIN 10 DEGREES
OF 90 TO BE EFFECTIVE. NEGATIVE DEG TURNS TO-
    WARDS
0 AND POSITIVE DEG TOWARDS 180. ENTERING NEGATIVE
    KM
MOVES YOU TOWARDS THE ENEMY, WHILE POSITIVE
    MOVES
YOU AWAY. LASER EFFECTIVENESS IS RANDOM, DUE TO
SHIELDING, DISTANCE AND INTERSTELLAR DEBRIS
****************GOOD-LUCK****************
MAY THE FORCE BE WITH YOU
DISTANCE TO ENEMY 3.7 E0 3KM
BEARING IS 87 DEGREES
THE SKY FIGHTER HAS FIRED HIS LASER
YOUR TOTAL ENERGY IS NOW 9.75E03 UNITS
THE ENEMY HAS 9.5E03 UNITS OF ENERGY LETS
WHICH COMMAND DO YOU WISH TO EXECUTE
?
2
HOW MANY KM TO TRANSVERSE
?
− 3.3E03
DISTANCE TO ENEMY 400KM
BEARING IS 87 DEGREES
THE SKY FIGHTER HAS FIRED HIS LASER

YOUR TOTAL ENERGY IS NOW 9.32E03 UNITS
THE ENEMY HAS 9.0E03 UNITS OF ENERGY LEFT
WHICH COMMAND DO YOU WISH TO EXECUTE
?
5
YOU HAVE INSTRUCTED THE ON-BOARD COMPUTER
TO SELF-DESTRUCT, THE REACTOR HAS GONE
CRITICAL, YOU HAVE GONE TO MEET THE FORCE
YOUR DESTRUCTION HAS ALSO DESTROYED
THE SKY FIGHTER, YOU WILL BE REMEMBERED
AS A HERO
SPACE WARS IS OVER
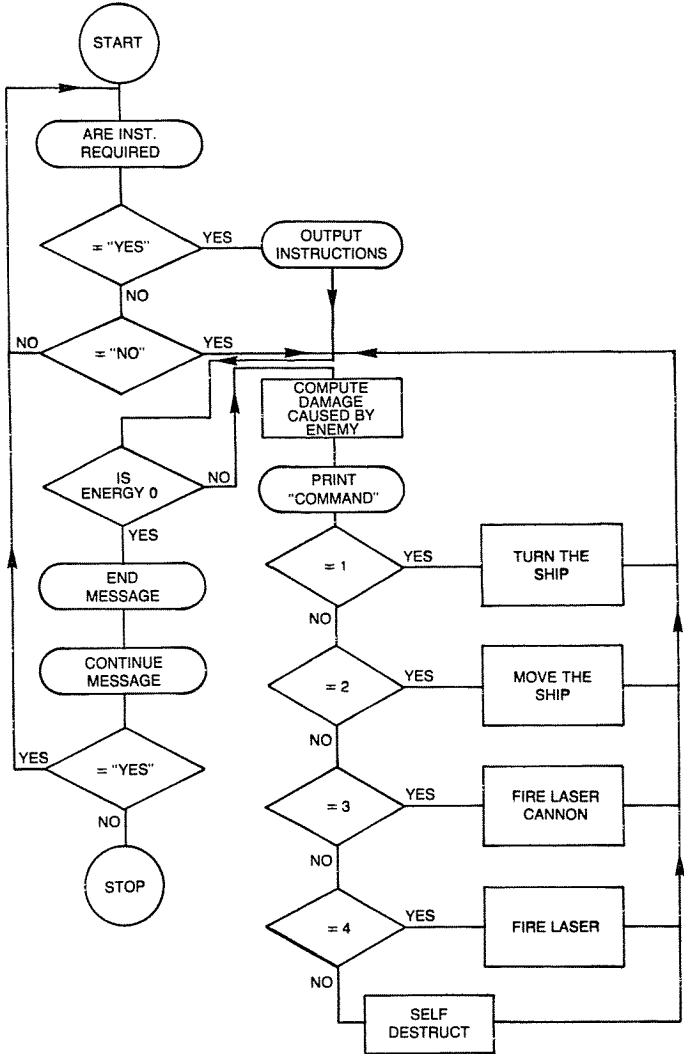TO PLAY SPACE WARS AGAIN TYPE GO,
OTHERWISE TYPE NO
?
NO
SPACE WARS SAYS GOOD-BYE
*END

SPACE WARS (1)

START

ARE INST. REQUIRED

≠ "YES" — YES → OUTPUT INSTRUCTIONS

NO

= "NO" — YES

NO

IS ENERGY 0 — NO → PRINT "COMMAND"

COMPUTE DAMAGE CAUSED BY ENEMY

YES

END MESSAGE

CONTINUE MESSAGE

= "YES" — YES

NO

STOP

= 1 — YES → TURN THE SHIP

NO

= 2 — YES → MOVE THE SHIP

NO

= 3 — YES → FIRE LASER CANNON

NO

= 4 — YES → FIRE LASER

NO

SELF DESTRUCT

SPACE WARS (1)

```
10    REM THIS PROGRAM IS THE GAME OF SPACE
      WARS
20    REM TWO SHIPS BATTLE, YOU MUST DESTROY
      THE
30    REM ENEMY TO SAVE THE REPUBLIC
40    RANDOMIZE
50    PRINT "ARE INSTRUCTIONS REQUIRED? TYPE
      EITHER"
60    PRINT "YES OR NO"
70    INPUT A$
80    IF A$ = "YES" THEN 120
90    IF A$ = "NO" THEN 250
100   PRINT "INVALID RESPONSE"
110   GOTO 50
120   PRINT
130   PRINT "THERE ARE 5 EXECUTIVE COMMANDS;
      TURN THE"
140   PRINT "SHIP(1), MOVE(2), FIRE LASER CANNON(3),"
150   PRINT "FIRE LASER(4) AND SELF-DESTRUCT(5)"
160   PRINT "THE CANNON MUST BE FIRED WITHIN 10
      DEGREES"
170   PRINT "OF 90 TO BE EFFECTIVE. NEGATIVE DEG
      TURNS TOWARDS"
180   PRINT "0 AND POSITIVE DEG TOWARDS 180. EN-
      TERING NEGATIVE KM"
190   PRINT "MOVES YOU TOWARDS THE ENEMY, WHILE
      POSITIVE MOVES"
200   PRINT "YOU AWAY. LASER EFFECTIVENESS IS
      RANDOM, DUE TO"
210   PRINT "SHIELDING, DISTANCE AND INTERSTEL-
      LAR DEBRIS"
220   PRINT
230   PRINT "*************GOOD-LUCK****************"
240   PRINT "MAY THE FORCE BE WITH YOU"
250   LET E1 = 1E04
260   LET E2 = E1
270   LET D = 1E03 + INT(5E05*RND)
280   LET B = 1 + INT(180*RND)
290   GOSUB 340
300   GOSUB 390
310   LET E1 = E1 − D1
320   GOSUB 500
```

```
330    GOTO 690
340    IF D > = 1E05 THEN 370
350    LET L = 1
360    GOTO 380
370    LET L = 0
380    RETURN
390    IF L = 0 THEN 440
400    LET H2 = (1 + INT(100*RND))/100
410    LET D1 = 500*H2
420    LET E2 = E2 - 500
430    GOTO 490
440    LET M1 = 1 + INT(2*RND)
450    IF M1 = 1 THEN 470
455    LET D1 = 1000
460    GOTO 480
470    LET D1 = 0
480    LET E2 = E2 - 1000
490    RETURN
500    PRINT
510    PRINT "DISTANCE TO ENEMY ";D;"KM"
520    PRINT "BEARING IS ";B;" DEGREES"
530    IF L = 1 THEN 560
540    PRINT "THE SKY FIGHTER HAS FIRED THE LASER
           CANNON"
550    GOTO 570
560    PRINT "THE SKY FIGHTER HAS FIRED HIS LASER"
570    PRINT "YOUR TOTAL ENERGY IS NOW";E1;"
           UNITS"
580    PRINT "THE ENEMY HAS ";E2;" UNITS OF ENERGY
           LEFT"
590    IF E1 < = 0 THEN 620
600    IF E2 < = 0 THEN 650
610    GOTO 680
620    PRINT "YOUR ENERGY LEVEL IS ZERO, THE
           ENEMY"
630    PRINT "HAS WON, YOU HAVE BECOME ONE WITH
           THE FORCE!"
640    GOTO 1180
650    PRINT "THE ENEMY HAS RUN OUT OF ENERGY,
           YOU"
660    PRINT "HAVE WON"
670    GOTO 1180
680    RETURN
```

```
690    PRINT
700    PRINT "WHICH COMMAND DO YOU WISH TO EXE-
          CUTE"
710    INPUT C
720    ON C GOTO 730, 840, 1010, 1100, 1140
730    PRINT "HOW MANY DEGREES OF ROTATION"
740    INPUT B1
750    IF B + B1 = 0 THEN 800
760    IF B + B1 > 180 THEN 820
770    LET B = B + B1
780    LET E1 = E1 - (10*ABS(B1))
790    GOTO 290
800    PRINT "YOUR ANGLE MUST BE GREATER THAN 0
          DEGREES"
810    GOTO 730
820    PRINT "YOUR ANGLE MUST BE LESS THAN 181 DE-
          GREES"
830    GOTO 730
840    PRINT "HOW MANY KM TO TRANSVERSE"
850    INPUT K
860    IF D + K = 0 THEN 910
870    IF D + K > 1E06 THEN 960
880    LET D = D + K          K/1000))
890    LET E1 = E1 - (ABS(
900    GOTO 290
910    PRINT "YOU HAVE TRIED TO CLOSE THE DIS-
          TANCE TO ZERO"
920    PRINT "BETWEEN YOU AND THE ENEMY, THE
          ON-BOARD"
930    PRINT "COMMAND COMPUTER WILL NOT EXE-
          CUTE THIS MANEUVER"
940    PRINT
950    GOTO 840
960    PRINT "YOU HAVE TRIED TO EXCEED THE DIS-
          TANCE WHERE"
970    PRINT "ANY OF YOUR WEAPONS ARE EFFECTIVE"
980    PRINT "THE ON-BOARD COMPUTER WILL NOT"
990    PRINT "EXECUTE THIS MANEUVER"
1000   GOTO 840
1010   LET E1 = E1 - 1000
1020   IF B > = 80 THEN 1050
1030   PRINT "YOUR ANGLE IS TOO SMALL, YOU HAVE
          MISSED"
```

```
1040    GOTO 290
1050    IF B < = 100 THEN 1080
1060    PRINT "YOUR ANGLE IS TOO GREAT, YOU HAVE
        MISSED"
1070    GOTO 290
1080    LET E2 = E2 − 1000
1090    GOTO 290
1100    LET E1 = E1 − 500
1110    LET D2 = (1 + INT(100*RND))/100
1120    LET E2 = E2 − (D2*500)
1130    GOTO 290
1140    PRINT "YOU HAVE INSTRUCTED THE ON-BOARD
        COMPUTER"
1150    PRINT "TO SELF-DESTRUCT, THE REACTOR HAS
        GONE"
1160    PRINT "CRITICAL, YOU HAVE GONE TO MEET THE
        FORCE"
1170    IF D < = 500 THEN 1200
1180    PRINT "SPACE WARS IS OVER"
1190    GOTO 1240
1200    PRINT "YOUR DESTRUCTION HAS ALSO DE-
        STROYED"
1210    PRINT "THE SKY FIGHTER, YOU WILL BE REMEM-
        BERED"
1220    PRINT "AS A HERO"
1230    GOTO 1180
1240    PRINT
1250    PRINT "TO PLAY SPACE WARS AGAIN TYPE GO,"
1260    PRINT "OTHERWISE TYPE NO"
1270    INPUT Z$
1280    IF Z$ = "GO" THEN 1310
1290    PRINT "SPACE WARS SAYS GOOD-BYE"
1300    STOP
1310    PRINT
1320    GOTO 50
1330    END
```

SPACE WARS (2)

This computer simulation requires considerably more memory than any other program in the book. For the user who is memory limited, the following may be done. Deleting the REM statements, removing the instructions, and reducing the string lengths in the messages will reduce the memory requirement by about 50 percent.

EXAMPLE

RUN
ARE INSTRUCTIONS FOR SPACE WARS REQUIRED? TYPE EITHER YES OR NO
?
YES
*******************SPACE WARS*****************
THE DEATH STAR SPACE STATION, YOUR GOAL, IS HEAV-
ILY
SHIELDED AND MOUNTS MORE FIREPOWER THAN HALF
THE IMPERIAL FLEET. BUT, ITS DEFENSES WERE
PRIMARILY DESIGNED TO FEND OFF LARGE-SCALE CAPI-
TAL
SPACE-SHIP ASSAULTS. A SMALL, ONE- OR TWO-MAN
X-WING FIGHTER SHOULD BE ABLE TO SLIP THROUGH
ITS DEFENSIVE SCREENS. YOUR MISSION IS TO DESTROY
THE DEATH STAR!!! ON ITS SURFACE THERE IS A SMALL
THERMAL EXHAUST PORT. ITS SIZE BELIES ITS IMPOR-
TANCE
AS IT APPEARS TO BE AN UNSHIELDED SHAFT THAT RUNS
DIRECTLY INTO THE MAIN REACTOR SYSTEM POWERING
THE DEATH STAR SPACE STATION. SINCE THIS SERVES
AS AN EMERGENCY OUTLET FOR WASTE HEAT IN THE
EVENT OF REACTOR OVERPRODUCTION, ITS USEFUL-
NESS
WOULD BE ELIMINATED BY ENERGY-PARTICLE SHIELD-
ING.
A DIRECT HIT WOULD INITIATE A CHAIN REACTION THAT
WOULD DESTROY THE STATION, THUS PROTECTING THE
REPUBLIC
*********EXECUTIVE COMMANDS ARE*********
(1) FIRE HIGH-ENERGY TORPEDO
(2) FIRE LASER CANNON
(3) FIRE LASER
(4) PROPULSION OF X-WING

184

THE BATTLE COMPUTER OPTION MAY BE USED WITH COMMANDS
2 AND 3. THE ENERGY TORPEDO IS USED TO DESTROY THE DEATH STAR WITH, EACH TORPEDO EXPENDS 20,000 UNITS OF ENERGY. THE LASER CANNON MAY BE USED AGAINST THE SKY FIGHTERS, IT REQUIRES 5,000 UNITS THE LASER USES 1,000 ENERGY UNITS PER SHOT AND IT IS ALSO USED AGAINST THE SKY FIGHTERS.
THE BATTLE COMPUTER REQUIRES 500 ENERGY UNITS, BUT,
GUARANTEES A DIRECT HIT ON A SKY FIGHTER. TO DESTROY A SKY FIGHTER YOU MUST DEPLETE IT OF ENERGY.
MOVING THE X-WING SPACE CRAFT IS IMPERATIVE AS THE ENERGY TORPEDO MUST BE FIRED WITHIN 1000KM OF THE DEATH STAR. X-WING PROPULSION REQUIRES 1 ENERGY UNIT PER KM
****
GOOD-LUCK
MAY THE FORCE BE WITH YOU
****
DISTANCE TO DEATH STAR IS NOW 1E05KM
THE SKY FIGHTER HAS FIRED HIS LASER CANNON
THE DARK LORD IS EXTREMELY DANGEROUS!!!
THE SMITH LORD HAS USED A LASER CANNON ENERGY BEAM
YOUR TOTAL ENERGY IS NOW 4.98E05
WHICH COMMAND DO YOU WISH TO EXECUTE
?
4
HOW MANY UNITS OF ENERGY DO YOU WISH TO FEED TO THE HYPER-ATOMIC DRIVE UNIT, (1 UNIT/1KM)
CAUTION TOO MUCH ENERGY WILL OVER-HEAT THE REACTOR, INPUT NO MORE THAN 22,500 UNITS AT ANY ONE TIME
?
30,000
IN WHICH DIRECTION, AWAY (A) OR TOWARDS (T) THE DEATH STAR
?
T
YOU HAVE WASTED 3.0E04 UNITS OF ENERGY
THE REACTOR IS CRITICALLY OVERHEATED

DISTANCE TO DEATH STAR IS NOW 1E05KM
THE SKY FIGHTER HAS FIRED HIS LASER CANNON
RADER'S ON-BOARD ATTACK COMPUTER HAS MATCHED
YOUR COURSE, HIS WEAPONS ARE READY
THE SMITH LORD HAS USED A LASER CANNON ENERGY
  BEAM
YOUR TOTAL ENERGY IS NOW 4.39E05 UNITS
WHICH COMMAND DO YOU WISH TO EXECUTE
?
4
HOW MANY UNITS OF ENERGY DO YOU WISH TO FEED TO
THE HYPER-ATOMIC DRIVE UNIT, (1 UNIT/1KM)
CAUTION TOO MUCH ENERGY WILL OVER-HEAT
THE REACTOR, INPUT NO MORE THAN 22,500 UNITS
AT ANY ONE TIME
?
20,000
IN WHICH DIRECTION, AWAY (A) OR TOWARDS (T)
THE DEATH STAR
?
T
DISTANCE TO DEATH STAR IS NOW 8.0E04KM
THE SKY FIGHTER HAS FIRED HIS LASER CANNON
CAUTION GARTH RADER IS THE BEST SHOT IN THE
IMPERIAL FLEET, PLUS HE USES THE BAD SIDE OF THE
  FORCE
THE SMITH LORD HAS USED A LASER CANNON ENERGY
  BEAM
YOUR TOTAL ENERGY IS NOW 4.1E05 UNITS
WHICH COMMAND DO YOU WISH TO EXECUTE
?
///
TO PLAY SPACE WARS AGAIN TYPE GO, IF
NOT TYPE NO
?
NO
SPACE WARS SAY GOOD-BYE AND MAY THE FORCE
  BE WITH YOU
*END
The ////// indicate where the program was terminated, this simula-
tion may be played for a considerable length of time. In the interest of
saving space and leaving the unexpected to the user only a portion of
a typical run has been shown.

186

```
10   REM THIS COMPUTER SIMULATION IS AN AD-
     VANCED VERSION OF
20   REM THE SPACE WARS GAME. YOUR MISSION IS
     TO DESTROY THE
30   REM DEATH STAR. YOU MAY BE ATTACKED BY
     THE DEATH STAR'S
40   REM DEFENSES AND BY THE SKY FIGHTERS
50   RANDOMIZE
60   PRINT "ARE INSTRUCTIONS FOR SPACE WARS
     REQUIRED? TYPE"
70   PRINT "EITHER YES OR NO"
80   INPUT A$
90   IF A$ = "YES" THEN 130
100  IF A$ = "NO" THEN 540
110  PRINT "YOU HAVE ISSUED AN INVALID RE-
     SPONSE"
120  GOTO 60
130  PRINT
140  PRINT "*****************SPACE
     WARS***************"
150  PRINT "THE DEATH STAR SPACE STATION, YOUR
     GOAL, IS HEAVILY"
160  PRINT "SHIELDED AND MOUNTS MORE
     FIREPOWER THAN HALF"
170  PRINT "THE IMPERIAL FLEET. BUT, ITS DE-
     FENSES WERE"
180  PRINT "PRIMARILY DESIGNED TO FEND OFF
     LARGE-SCALE CAPITAL"
190  PRINT "SPACE-SHIP ASSAULTS. A SMALL, ONE-OR
     TWO-MAN"
200  PRINT "X-WING FIGHTER SHOULD BE ABLE TO
     SLIP THROUGH"
210  PRINT "ITS DEFENSIVE SCREENS. YOUR MISSION,
     IS TO DESTROY"
220  PRINT "THE DEATH STAR!!! ON ITS SURFACE
     THERE IS A SMALL"
230  PRINT "THERMAL EXHAUST PORT. ITS SIZE BE-
     LIES ITS IMPORTANCE"
240  PRINT "AS IT APPEARS TO BE AN UNSHIELDED
     SHAFT THAT RUNS"
250  PRINT "DIRECTLY INTO THE MAIN REACTOR
     SYSTEM, POWERING"
```

```
260   PRINT "THE DEATH STAR SPACE STATION. SINCE
      THIS SERVES"
270   PRINT "AS AN EMERGENCY OUTLET FOR WASTE
      HEAT IN THE"
280   PRINT "EVENT OF REACTOR OVERPRODUCTION,
      ITS USEFULNESS'
290   PRINT "WOULD BE ELIMINATED BY ENERGY-
      PARTICLE SHIELDING"
300   PRINT "A DIRECT HIT WOULD INITIATE A CHAIN
      REACTION THAT"
310   PRINT "WOULD DESTROY THE STATION, THUS
      PROTECTING THE"
320   PRINT "REPUBLIC"
330   PRINT "*******EXECUTIVE COMMANDS
      ARE********"
340   PRINT "(1) FIRE HIGH-ENERGY TORPEDO"
350   PRINT "(2) FIRE LASER CANNON"
360   PRINT "(3) FIRE LASER"
370   PRINT "(4) PROPULSION OF X-WING"
380   PRINT "THE BATTLE COMPUTER OPTION MAY BE
      USED WITH COMMANDS"
390   PRINT "2 AND 3. THE ENERGY TORPEDO IS USED
      TO DESTROY"
400   PRINT "THE DEATH STAR WITH, EACH TORPEDO
      EXPENDS 20,000"
410   PRINT "UNITS OF ENERGY. THE LASER CANNON
      MAY BE USED"
420   PRINT "AGAINST THE SKY FIGHTERS, IT RE-
      QUIRES 5,000 UNITS"
430   PRINT "THE LASER USES 1,000 ENERGY UNITS
      PER SHOT AND IT"
440   PRINT "IS ALSO USED AGAINST THE SKY FIGHT-
      ERS."
450   PRINT "THE BATTLE COMPUTER REQUIRES 500
      ENERGY UNITS, BUT,"
460   PRINT "GUARANTEES A DIRECT HIT ON A SKY
      FIGHTER. TO DESTROY"
465   PRINT "A SKY FIGHTER YOU MUST DEPLETE IT OF
      ENERGY."
470   PRINT "MOVING THE X-WING SPACE CRAFT IS IM-
      PERATIVE AS"
480   PRINT "THE ENERGY TORPEDO MUST BE FIRED
      WITHIN 1000KM"
```

```
490    PRINT "OF THE DEATH STAR. X-WING PROPUL-
       SION REQUIRES"
500    PRINT "1 ENERGY UNIT PER KM"
510    PRINT"***********************************************"
520    PRINT" GOOD-LUCK "
530    PRINT "MAY THE FORCE BE WITH YOU"
535    PRINT"***********************************************"
536    REM X-WING ENERGY AND SKY ENERGY
540    LET X1 = 5E05
550    LET T1 = 1E04
560    LET T2 = 5E04
570    LET D = 1E05
580    GOSUB 630
590    GOSUB 730
600    GOSUB 830
610    GOSUB 1040
620    GOSUB 1500
625    GOTO 580
630    IF D > 1E04 THEN 660
640    LET L = 1
650    GOTO 700
660    LET L = 0
670    LET H = (1 + INT(100*RND))/100
680    LET E1 = 5000*H
690    GOTO 720
700    LET H = (1 + INT(100*RND))/100
710    LET E1 = 1000*H
720    RETURN
730    IF D > 5E03 THEN 760
740    LET K = 1
750    GOTO 800
760    LET K = 0
770    LET H = (1 + INT(100*RND))/100
780    LET E2 = 8000*H
790    GOTO 820
800    LET H = (1 + INT(100*RND))/100
810    LET E2 = 3000*H
820    RETURN
830    IF D < 3E03 THEN 860
840    LET E3 = 0
850    GOTO 880
860    LET H = (1 = INT(100*RND))/100
870    LET E3 = 2E04*H
```

```
880    LET T1 = T1 - E1
890    LET T2 = T2 - E2
900    IF T1 < = 0 THEN 920
910    GOTO 950
920    LET E1 = 0
930    LET Y = 1
940    GOTO 960
950    LET Y = 0
960    IF T2 < = 0 THEN 980
970    GOTO 1010
980    LET E2 = 0
990    LET Z = 1
1000   GOTO 1020
1010   LET Z = 0
1020   LET X1 = X1 - E1 - E2 - E3
1030   RETURN
1040   PRINT "DISTANCE TO DEATH STAR IS NOW";
         D;"KM"
1045   IF Y = 1 THEN 1100
1050   IF L = 0 THEN 1080
1060   PRINT "THE SKY FIGHTER HAS FIRED HIS LASER"
1070   GOTO 1110
1080   PRINT "THE SKY FIGHTER HAS FIRED HIS LASER
         CANNON"
1090   GOTO 1110
1100   PRINT "THE SKY FIGHTER IS OUT OF ACTION!!!"
1110   IF Z = 1 THEN 1180
1120   GOTO 1220
1130   IF K = 0 THEN 1160
1140   PRINT"THE DARK LORD HAS FIRED HIS HIGH
         ENERGY LASER"
1150   GOTO 1360
1160   PRINT "THE SMITH LORD HAS USED A LASER CAN-
         NON ENERGY BEAM"
1170   GOTO 1360
1180   PRINT "GARTH RADER HAS EXPENDED ALL HIS
         WEAPON'S ENERGY"
1190   PRINT "SUPPLY. HE IS CURRENTLY ESCAPING TO
         THE ENDS OF"
1200   PRINT "THE GALAXY. ***THE FORCE IS WITH
         YOU***"
1210   GOTO 1360
```

```
1220   LET C = 1 + (5*RND)
1230   ON C GOTO 1240, 1270, 1290, 1310, 1330
1240   PRINT "*CAUTION*GARTH RADER IS THE BEST
         SHOT IN THE"
1250   PRINT "IMPERIAL FLEET, PLUS HE USES THE BAD
         SIDE OF THE FORCE"
1260   GOTO 1130
1270   PRINT "THE DARK LORD IS EXTREMELY
         DANGEROUS!!!"
1280   GOTO 1130
1290   PRINT "**CAUTION RADER IS INHUMANLY ACCU-
         RATE CAUTION**"
1300   GOTO 1130
1310   PRINT "THE SMITH LORD'S PRECISION IS AWE-
         SOME"
1320   GOTO 1130
1330   PRINT "RADER'S ON-BOARD ATTACK COMPUTER
         HAS MATCHED"
1340   PRINT "YOUR COURSE, HIS WEAPONS ARE
         READY"
1350   GOTO 1130
1360   IF D < = 3E03 THEN 1380
1370   GOTO 1410
1380   PRINT "***YOU ARE CLOSER THAN 3000KM TO
         THE SPACE STATION"
1390   PRINT "THE DEATH STAR'S AUTOMATIC DE-
         FENSE NETWORK HAS BEEN"
1400   PRINT "ACTIVATED. ***USE EXTREME CAU-
         TION***"
1410   PRINT
1415   PRINT "YOUR TOTAL ENERGY IS NOW ";X1;"
         UNITS"
1420   IF X1 < 2E04 THEN 1140
1430   GOTO 1490
1440   PRINT "YOU HAVE DEPLETED YOUR ENERGY
         SUPPLY, THE DEATH"
1450   PRINT "STAR WILL NOW DESTROY YOUR HOME
         PLANET"
1460   PRINT "YOU WILL BE A HERO NOWHERE AND RE-
         MEMBERED BY NONE"
1470   PRINT "*****YOU HAVE MISUSED THE
         FORCE*****"
1480   GOTO 2690
```

```
1490    RETURN
1500    PRINT
1510    PRINT "WHICH COMMAND DO YOU WISH TO EXE-
           CUTE"
1520    INPUT B
1530    ON B GOTO 1540, 1690, 2030, 2320
1540    IF D < = 1000 THEN 1590
1550    PRINT "YOU HAVE WASTED A TORPEDO, YOU ARE
           FARTHER"
1560    PRINT "AWAY THAN 1000KM"
1570    LET X1 = X1 - 2E04
1580    GOTO 2680
1590    LET H = 1 + (INT(100*RND))
1600    IF H > = 50 THEN 1640
1610    PRINT "YOU SHOULD HAVE USED THE FORCE,
           YOU HAVE MISSED"
1620    LET X1 = X1 - 2E04
1630    GOTO 2680
1640    PRINT "THE FORCE WAS WITH YOU, YOU HAVE
           SINGLE-HANDED"
1650    PRINT "DESTROYED THE DEATH STAR. YOU
           HAVE SAVED THE"
1660    PRINT "REPUBLIC AND PRINCESS LEAH ARGONA
           WILL LOVE"
1670    PRINT "YOU FOREVER."
1680    GOTO 2740
1690    PRINT "THE CANNON IS READY, DO YOU WISH
           COMPUTER ASSISTANCE"
1700    PRINT "ENTER EITHER YES OR NO"
1710    INPUT C$
1720    IF C$ = "YES" THEN 1820
1730    IF C$ = "NO" THEN 1880
1740    PRINT "INVALID RESPONSE"
1750    GOTO 1700
1760    PRINT "WHICH FIGHTER THE SKY "S" OR RADER
           "R"
1770    INPUT C$
1780    IF C$ = "T" THEN 1840
1790    IF C$ = "V" THEN 1860
1800    PRINT "WHICH????"
1810    GOTO 1760
1820    LET Q = 5000
1830    GOTO 1760
```

```
1840    LET T1 = T1 - Q
1850    GOTO 1960
1860    LET T2 = T2 - Q
1870    GOTO 1960
1880    PRINT "DO YOU WISH TO FIRE ON GARTH RADER
            (R) OR"
1890    PRINT "ON THE SKY FIGHTER (S)"
1900    INPUT C$
1910    LET Q = 5000*((1 + INT(100*RND))/100)
1920    IF C$ = "S" THEN 1980
1930    IF C$ = "R" THEN 2000
1940    PRINT "WHICH ENEMY????"
1950    GOTO 1880
1960    LET X1 = X1 - 5500
1970    GOTO 2660
1980    LET T1 = T1 - Q
1990    GOTO 2010
2000    LET T2 = T2 - Q
2010    LET X1 = X1 - 5000
2020    GOTO 2660
2030    PRINT "YOU HAVE DECIDED ON USING THE
            LASER"
2040    PRINT "DO YOU WISH COMPUTER ASSISTANCE,
            YES OR NO"
2050    INPUT C$
2060    IF C$ = "YES" THEN 2100
2070    IF C$ = "NO" THEN 2120
2080    PRINT "THE COMPUTER RESPONSES ONLY TO A
            YES OR A NO"
2090    GOTO 2040
2100    LET J = 1
2110    GOTO 2130
2120    LET J = 0
2130    PRINT "WHICH FIGHTER THE SKY (S) OR RADER
            (R)"
2140    PRINT "DO YOU WISH TO FIRE ON"
2150    INPUT C$
2160    IF J = 1 THEN 2190
2170    LET Q = 1000*((1 + INT(100*RND))/100)
2180    GOTO 2200
2190    LET Q = 1000
2200    IF C$ = "S" THEN 2240
2210    IF C$ = "R" THEN 2260
```

```
2220    PRINT "WHICH TARGET?????"
2230    GOTO 2130
2240    LET T1 = T1 − Q
2250    GOTO 2270
2260    LET T2 = T2 − Q
2270    IF J = 1 THEN 2300
2280    LET X1 = X1 − 1000
2290    GOTO 2660
2300    LET X1 = X1 − 1500
2310    GOTO 2660
2320    PRINT "HOW MANY UNITS OF ENERGY DO YOU
          WISH TO FEED TO"
2330    PRINT "THE HYPER-ATOMIC DRIVE UNIT, (1
          UNIT/1KM)"
2340    PRINT "**CAUTION** TOO MUCH ENERGY WILL
          OVER-HEAT"
2350    PRINT "THE REACTOR, INPUT NO MORE THAN
          22,500 UNITS"
2360    PRINT "AT ANY ONE TIME"
2370    INPUT F
2380    PRINT "IN WHICH DIRECTION, AWAY (A) OR TO-
          WARDS (T)"
2390    PRINT "THE DEATH STAR"
2400    INPUT C$
2410    IF F > 2.25E04 THEN 2470
2420    IF C$ = "A" THEN 2500
2430    IF C$ = "T" THEN 2620
2440    PRINT "DON'T YOU KNOW WHICH DIRECTION YOU
          WANT TO GO TO"
2450    GOTO 2330
2460    REM OVERHEATING THE REACTOR
2470    PRINT "YOU HAVE WASTED ";F;" UNITS OF
          ENERGY"
2480    PRINT "THE REACTOR IS CRITICALLY OVER-
          HEATED"
2490    GOTO 2640
2500    LET D = D + F
2510    IF D > = 1.5E05 THEN 2540
2520    GOTO 2640
2530    REM WENT TOO FAR
2540    PRINT "WHERE ARE YOU GOING?? THE BATTLE IS
          IN THE"
2550    PRINT "OPPOSITE DIRECTION"
```

```
2560    GOTO 2640
2570    PRINT "YOU HAVE SMASHED INTO THE DARK
            STAR*******"
2580    PRINT "WHERE DID YOU LEARN TO FLY, GARTH
            RADER"
2590    PRINT "IS LAUGHING AT YOU;; OH!! BY THE
            WAY---"
2600    PRINT "* * * * * YOU HAVE LOST * * * * * *"
2610    GOTO 2690
2620    LET D = D - F
2630    IF D < = 0 THEN 2570
2640    LET X1 = X1 - F
2650    GOTO 2680
2660    PRINT "THE SKY FIGHTER'S ENERGY IS NOW
            ";T1;"UNITS"
2670    PRINT "THE DARK LORD'S ENERGY IS" ;T2;"
            UNITS"
2680    RETURN
2690    PRINT "YOU ARE AN INCOMPETENT GOOD
            KNIGHT"
2700    PRINT "YOU HAVE DISGRACED THE MEMORY OF"
2710    PRINT "OBI- SAN COYOTE! WHOSE SIDE ARE YOU
            ON??"
2720    PRINT "WHY DON'T YOU PROVE YOUR WORTH
            AND TRY AGAIN"
2730    GOTO 2760
2740    PRINT "OBI- SAN COYOTE WOULD BE PROUD OF
            YOU"
2750    PRINT "YOU ARE INDEED A **GOOD KNIGHT**"
2760    PRINT
2770    PRINT "TO PLAY SPACE WARS AGAIN TYPE GO, IF"
2780    PRINT "NOT TYPE NO"
2790    INPUT L$
2800    IF L$ = "GO" THEN 2840
2810    IF L$ = "NO" THEN 2860
2820    PRINT "DO YOU WANT TO STOP OR PLAY
            AGAIN?????"
2830    GOTO 2770
2840    PRINT
2850    GOTO 50
2860    PRINT "SPACE WARS SAY GOOD-BYE AND MAY
            THE FORCE BE WITH YOU"
2870    END
```

STRAIGHT LINE DEPRECIATION

This program computes the value depreciation of an item by the straight line method.

FORMULA

X_c = last current value, D = depreciation per year,
X_1 = new current value
$$X_c - D = X_1$$

EXAMPLE

ORIGINAL VALUE =
?
10000
LIFETIME IN YEARS =
?
12
YEARLY DEPRECIATION = 833.333

YEAR	VALUE
1	9166.66
2	8333.33
3	7500.
4	6666.66
5	5833.33
6	5000.
7	4166.66
8	3333.33
9	2500.
10	1666.66
11	333.333
12	0

****
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

STRAIGHT LINE DEPRECIATION

```
              ( START )
                  │
                  ▼
        ┌──────────────────┐
        │ PROMPT + INPUT   │
        │ LIFETIME + VALUE │
        └──────────────────┘
                  │
                  ▼
        ┌──────────────────┐
        │    CALCULATE     │
        │     YEARLY       │
        │  DEPRECIATION    │
        └──────────────────┘
                  │
                  ▼
        ┌──────────────────┐
        │     OUTPUT       │
        │      YEAR        │
        │  DEPRECIATION    │
        └──────────────────┘
                  │
                  ▼
        ┌──────────────────┐
        │    CALCULATE     │
        │  VALUE AT END    │
        │   OF YEAR X      │
        └──────────────────┘
                  │
                  ▼
            ╱IS A <0?╲──── YES ───┐
            ╲        ╱            ▼
                │             ┌────────────┐
                NO            │  .F₁ A + O │
                │             └────────────┘
                ▼
        ┌──────────────────┐
        │     OUTPUT       │
        │   YEAR VALUE     │
        └──────────────────┘
                  │
                  ▼
   YES ──── ╱ IS X <B? ╲
            ╲          ╱
                │
                NO
                ▼
        ┌──────────────────┐
        │ OUTPUT CONTINUE  │
        │   MESSAGE +      │
        │     INPUT        │
        └──────────────────┘
                  │
                  ▼
   YES ──── ╱  IS L = 1? ╲
            ╲            ╱
                │
                NO
                ▼
             ( STOP )
```
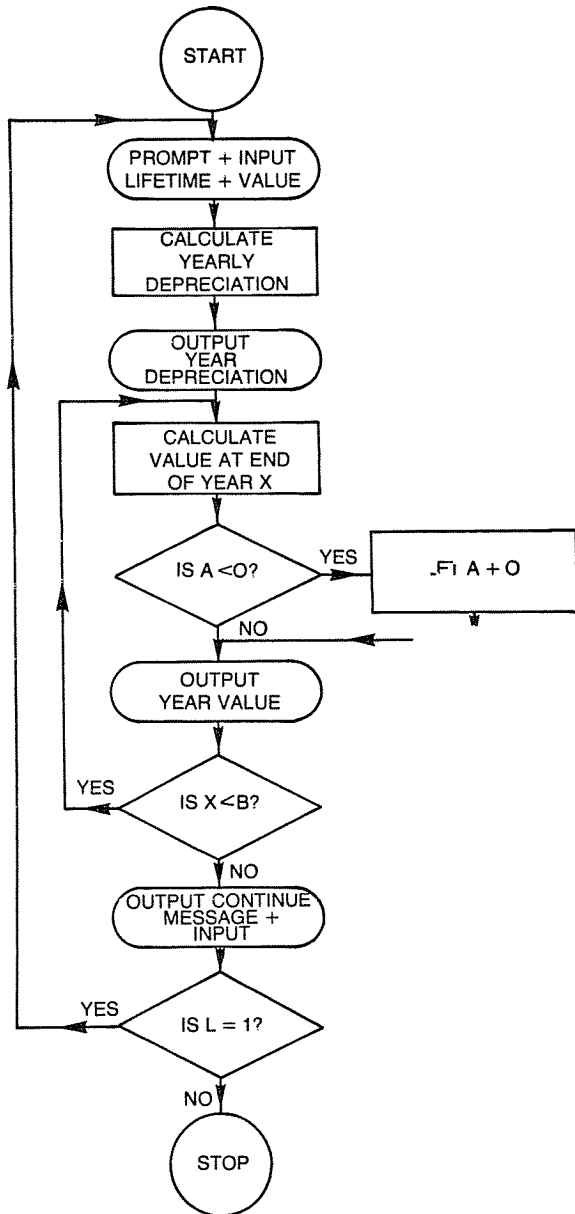
STRAIGHT LINE DEPRECIATION

```
10    REM THIS PROGRAM COMPUTES VALUE DEPRE-
      CIATION
20    REM BY THE STRAIGHT LINE METHOD
30    PRINT "ORIGINAL VALUE = ";
40    INPUT A
50    PRINT "LIFETIME IN YEARS = ";
60    INPUT B
70    LET C = A/B
80    PRINT "YEARLY DEPRECIATION = ";C
90    PRINT
100   PRINT "YEAR", "VALUE"
110   LET X = 0
120   LET X = X + 1
130   LET A = A - C
140   IF A < 0 THEN 160
150   GOTO 170
160   LET A = 0
170   PRINT X,A
180   IF X < B THEN 120
190   PRINT "*********************"
200   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
210   INPUT L
220   IF L = 1 THEN 240
230   STOP
240   PRINT
250   GOTO 30
260   END
```

VECTOR CROSS PRODUCT

If $A(A_1, A_2, A_3)$ and $B(B_1, B_2, B_3)$ are two three-dimensional vectors then the cross product of A and B is denoted by AxB. The program responds with a solution represented by X, Y and Z.

FORMULAE

$$A \times B = \begin{vmatrix} A_2 A_3 \\ B_2 B_3 \end{vmatrix}, \quad - \begin{vmatrix} A_1 A_3 \\ B_1 B_3 \end{vmatrix}, \quad \begin{vmatrix} A_1 A_2 \\ B_1 B_2 \end{vmatrix}$$

$$= (A_2 B_3 - A_3 B_2, \ A_3 B_1 - A_1 B_3, \ A_1 B_2 - A_2 B_1)$$

EXAMPLES

```
ENTER 1ST VECTOR (A1,A2,A3)
?
10,11,10
ENTER 2ND VECTOR (B1,B2,B3)
?
4,3,4
VECTOR CROSS PRODUCT
X = 14   Y = 0   Z = - 14
+++++++++++++++++++++++++++++++
TYPE 1 TO CONTINUE, 0 TO STOP
?
1
ENTER 1ST VECTOR (A1,A2,A3)
?
12,23,34
ENTER 2ND VECTOR (B1,B2,B3)
?
23,41,67
VECTOR CROSS PRODUCT
X = 147   Y = - 22   Z = - 37
++++++++++++++++++++++++++++++++++++++
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END
```

$A_1 B_1$

VECTOR CROSS PRODUCT

START

PROMPT + INPUT VECTORS

CALCULATE VECTOR CROSS PRODUCT

OUTPUT CROSS PRODUCT

OUTPUT MESSAGE + CONTINUE INPUT

IS L = 1?
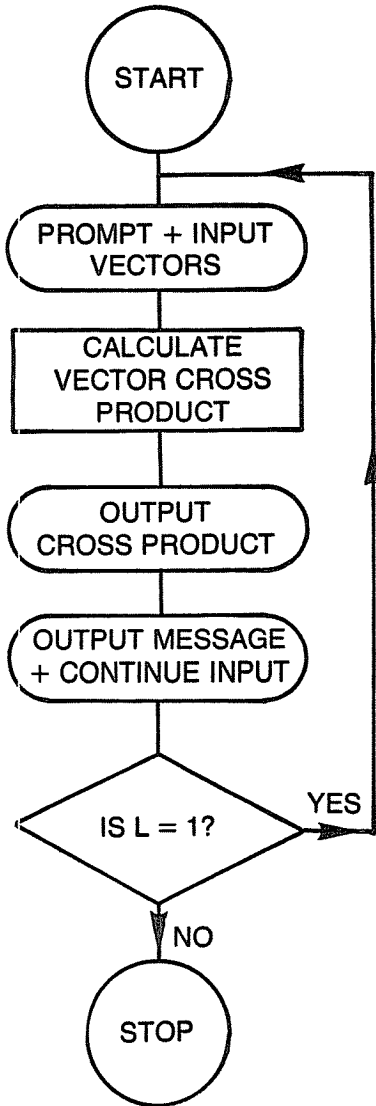
YES

NO

STOP

VECTOR CROSS PRODUCT

```
10    REM THIS PRODUCT COMPUTES THE CROSS PRO-
      DUCT
20    REM OF TWO VECTORS
30    PRINT "ENTER 1ST VECTOR (A1,A2,A3)"
40    INPUT A1,A2,A3
50    PRINT "ENTER 2ND VECTOR (B1,B2,B3)
60    INPUT B1,B2,B3
70    LET X = (A2*B3) – (A3*B2)
80    LET Y = (A3*B1) – (A1*B3)
90    LET Z = (A1*B2) – (A2*B1)
100   PRINT "VECTOR CROSS PRODUCT"
110   PRINT "X = ";X,"Y = ";Y,"Z = ";Z
120   PRINT "++++++++++++++++++++ "
130   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
140   INPUT L
150   IF L = 1 THEN 170
160   STOP
170   PRINT
180   GOTO 30
190   END
```

VECTOR DOT PRODUCT AND NORM

This program computes the vector dot product, also known as the scalar product and the norms of two vectors.

FORMULAE

$$\vec{A} = (A_1, A_2, A_3) \text{ and } \vec{B} \ (B_1, B_2, B_3) \text{ are two vectors}$$

$$NORM \text{ of } \vec{A} \text{ is denoted } |\vec{A}| \text{ and } \vec{B} \text{ is denoted } |\vec{B}|$$

$$|\vec{A}| = \sqrt{A_1^2 + A_2^2 = A_3^2}$$

$$|\vec{B}| = \sqrt{B_1^2 + B_2^2 + B_3^2}$$

$$\vec{A} \bullet \vec{B} = A_1 B_1 + A_2 B_2 + A_3 B_3$$

EXAMPLE

ENTER FIRST VECTOR (A1,A2,A3)
?
2,3,4
ENTER SECOND VECTOR (B1,B2,B3)
?
5,6,7
DOT PRODUCT = 56
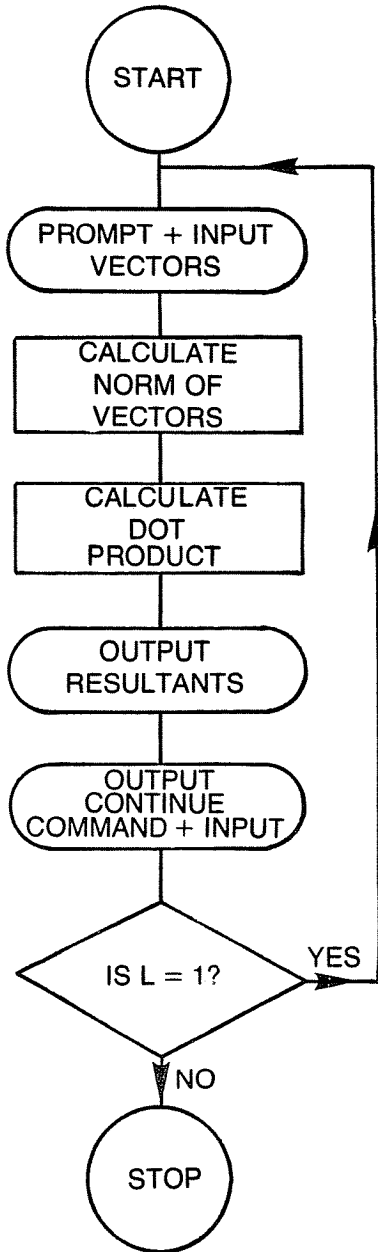NORM OF 1ST VECTOR = 5.38516
NORM OF 2ND VECTOR = 10.4880
****
TYPE 1 TO CONTINUE, 0 TO STOP
?
0
*END

DOT PRODUCT AND NORM

START

PROMPT + INPUT
VECTORS

CALCULATE
NORM OF
VECTORS

CALCULATE
DOT
PRODUCT

OUTPUT
RESULTANTS

OUTPUT
CONTINUE
COMMAND + INPUT

IS L = 1? YES

NO

STOP

VECTOR DOT PRODUCT AND NORM

```
10    REM THIS PROGRAM COMPUTES DOT PRODUCT
         AND
20    REM THE NORMS OF TWO VECTORS
30    PRINT "ENTER 1ST VECTOR (A1,A2,A3)"
40    INPUT A1,A2,A3
50    PRINT "ENTER 2ND VECTOR (B1,B2,B3)"
60    INPUT B1,B2,B3
70    REM CALCULATE NORM OF A VECTOR
80    LET X = SQR((A1↑2) + (A2↑2) + (A3↑2))
90    REM CALCULATE NORM OF B VECTOR
100   LET Y = SQR((B1↑2) + (B2↑2) + (B3↑2))
110   REM CALCULATE DOT PRODUCT
120   LET Z = (A1*B1) + (A2*B2) + (A3*B3)
130   PRINT "DOT PRODUCT = ";Z
140   PRINT "NORM OF 1ST VECTOR = ";X
150   PRINT "NORM OF 2ND VECTOR = ";Y
160   PRINT "************************"
170   PRINT "TYPE 1 TO CONTINUE, 0 TO STOP"
180   INPUT L
190   IF L = 1 THEN 210
200   STOP
210   PRINT
220   GOTO 30
230   END
```

57 PRACTICAL PROGRAMS & GAMES IN BASIC

From arithmetic progression to statistical permutations to one-arm bandits, here are 57 practical, useful and fun programs designed to help you really put your minicomputer to work!

Game programs include blackjack, one-arm bandit, craps, and two space war games. Math and accounting programs include compounding, straight-line depreciation, statistical permutations, instant derivatives, and solutions for integrals—even a whole section of geometric solutions for modern-day Euclids. For history buffs, there is a Day-of-the-Week program for any date back through 1753.

Each program begins with an introductory paragraph describing its capabilities, and continues with a typical program sequence and flowchart. All programs will run on any floating point BASIC.

The author is a veteran computer programmer with extensive experience in developing software in various languages for a wide range of hardware systems.

Radio Shack®