SHARP

FOR PC-1250/CE-125

APPLICATION SOFTWARE

PROGRAM EXAMPLES

MODEL PC-1250, OPTIONAL BOARD AND PERIPHERALS LIMITED WARRANTY

Sharp Electronics Corporation warrants each of these products to the original purchaser to be free from defective materials and workmanship. Under this warranty the product will be repaired or replaced, at our option, without charge for parts or labor, with the exception of supplies, such as batteries, ribbons, inked rollers, etc., when returned to a SHARP FACTORY SERVICE CENTER listed in the instruction booklet supplied with your product.

This warranty does not apply to cassette tapes, software programs or appearance items nor to any product whose exterior has been damaged or defaced, nor to any product subjected to misuse, abnormal service or handling, nor to any product altered or repaired by other than a SHARP FACTORY SERVICE CENTER. This warranty does not apply to any product purchased outside the United States, its territories or possessions.

The period of the warranty shall be ninety (90) days on parts and labor from the date of the original purchase.

This warranty entitles the original purchaser to have the warrantied parts and labor rendered at no cost for the period of the warranty described above when the unit is carried or shipped prepaid to a SHARP FACTORY SERVICE CENTER together with proof of purchase.

THIS SHALL BE THE EXCLUSIVE WRITTEN WARRANTY OF THE ORIGINAL PURCHASER AND NEITHER THIS WARRANTY NOR ANY OTHER WARRANTY, EXPRESSED OR IMPLIED SHALL EXTEND BEYOND THE PERIOD OF TIME LISTED ABOVE. IN NO EVENT SHALL SHARP BE LIABLE FOR CONSEQUENTIAL ECONOMIC DAMAGE OR CONSEQUENTIAL DAMAGE TO PROPERTY. SOME STATES DO NOT ALLOW A LIMITATION ON HOW LONG AN IMPLIED WARRANTY LASTS OR AN EXCLUSION OF CONSEQUENTIAL DAMAGE, SO THE ABOVE LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU. IN ADDITION, THIS WARRANTY GIVES SPECIFIC LEGAL RIGHTS AND YOU MAY HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

FOR YOUR RECORDS....

For your assistance in reporting this electronic calculator in case of loss or theft, please record below the model number and serial number which are located on the bottom of the unit.
Please retain this information.

Model Number_____ Serial Number_____

Date of Purchase Place of Purchase

_____ _____

APPLICATION SOFTWARE FOR PC-1250 / CE-125

This program library includes twenty (20) programs in mathematics, statistics, engeneering, business, and game fields.

The programs in this library are recorded in the supplied microcassette tape.

This program library is editted according to the followings, so please read these carefully in prior to your operation.

- **Program Title**
  This is a summary of the program contents.
- **Overview:**
  The brief explanation (concept) of the program is shown.
- **Instructions:**
  Shows the brief explanation of how to use and operate this program according to the "Key Operation Sequence" explained later.
- **Example:**
  For a better understanding of the program execution, an example using the program is provided.
- **Contents/Reference:**
  To let you understand the logics employed in the program such as formulas are explained.
- **Printed Outputs:**
  Outputs through the printer (CE-125) are provided by using the example.
- **Key Operation Sequence:**
  For your program execution, the actual key operation sequence is shown step by step by using the example.
- **Program List:**
  Program list is output through the CE-125.

  How to enter the programs into the machine.

  The Program List shown in this library is basically supposed to be typed in as it is printed.

  However, there are several points you should know in prior to the typing such as;
  1) The colon (:) right after each line number must be omitted.
  2) [ENTER] key must be pressed at the end of each program line.

  For more details, refer to the instruction manual of the PC-1250.

Showing the bytes used in each program itself

The number of bytes used in each program is shown at the end of each program listing. For instance, in the TYPING PRACTICE program, 475 were used bytes.

They way to find this out is as follows:

at RUN position

1) CLEAR [ENTER]

2) 1438 [−] MEM [ENTER] → number of bytes.

- **Memory Contents:**

  Memory contents during the program execution are explained.

* Also make sure that you use these programs after through checks through such as the examples.
* For continuous improvements and additions, these programs are subject to changes without notices.
* To help us improve our programs, we would appreciate any suggestions or comments in writing.
* **Sharp Corporation and/or its subsidiaries assume no responsibilities or obligations to any losses or damages that could arise through the use of this program examples and the relevant micro software tape.**

CONTENTS

Programs with (P) prefixed to the program title need CE-125 in program execution.

FILE NAME & TAPE COUNTER
in the supplied microcassette tape

Note: Some program list contents may differ slightly in between the programs included in the tape and this program library. In program execution, however this difference causes no impediment.

Program Title: NEWTON'S METHOD FOR FINDING ROOTS OF EQUATIONS

OVERVIEW (mathematical)

Finding the roots of equations is usually troublesome, but by using Newton's Method the approximate roots of equations can be found.

When 1 root is found, depending on the interval width, by using Newton's Method the starting point automatically changes.

CONTENTS

$$X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)}$$

If the absolute value of the distance between X_n and X_{n+1} is less than 10^{-8}, X_n is considered a root and is displayed. Here the first derivative is defined in the following way:

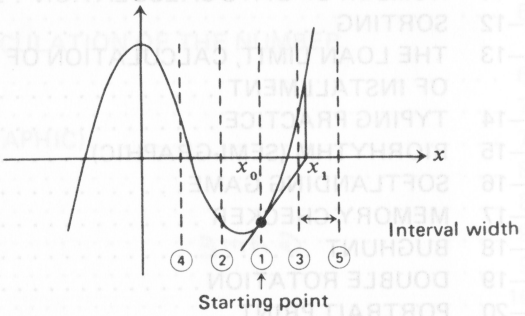
$$f'(X) = \frac{f(X+h) - f(X)}{h} \quad (h \text{ is the minute interval})$$

Change E-8 in line 340 to change the value for 10^{-8}.

INSTRUCTIONS

INPUT

Starting point
Minute interval
Interval



OUTPUTS

Root value (by pressing the ENTER key, the next interval's root is found)

EXAMPLE

$x^3 - 2x^2 - x + 2 = 0$ (the roots are $-1, 1, 2$)

starting point = 0
minute interval = 10^{-4}
interval = 0.5

The above values are used in the calculation.

The functions are to be written into lines after 500 as subroutines.

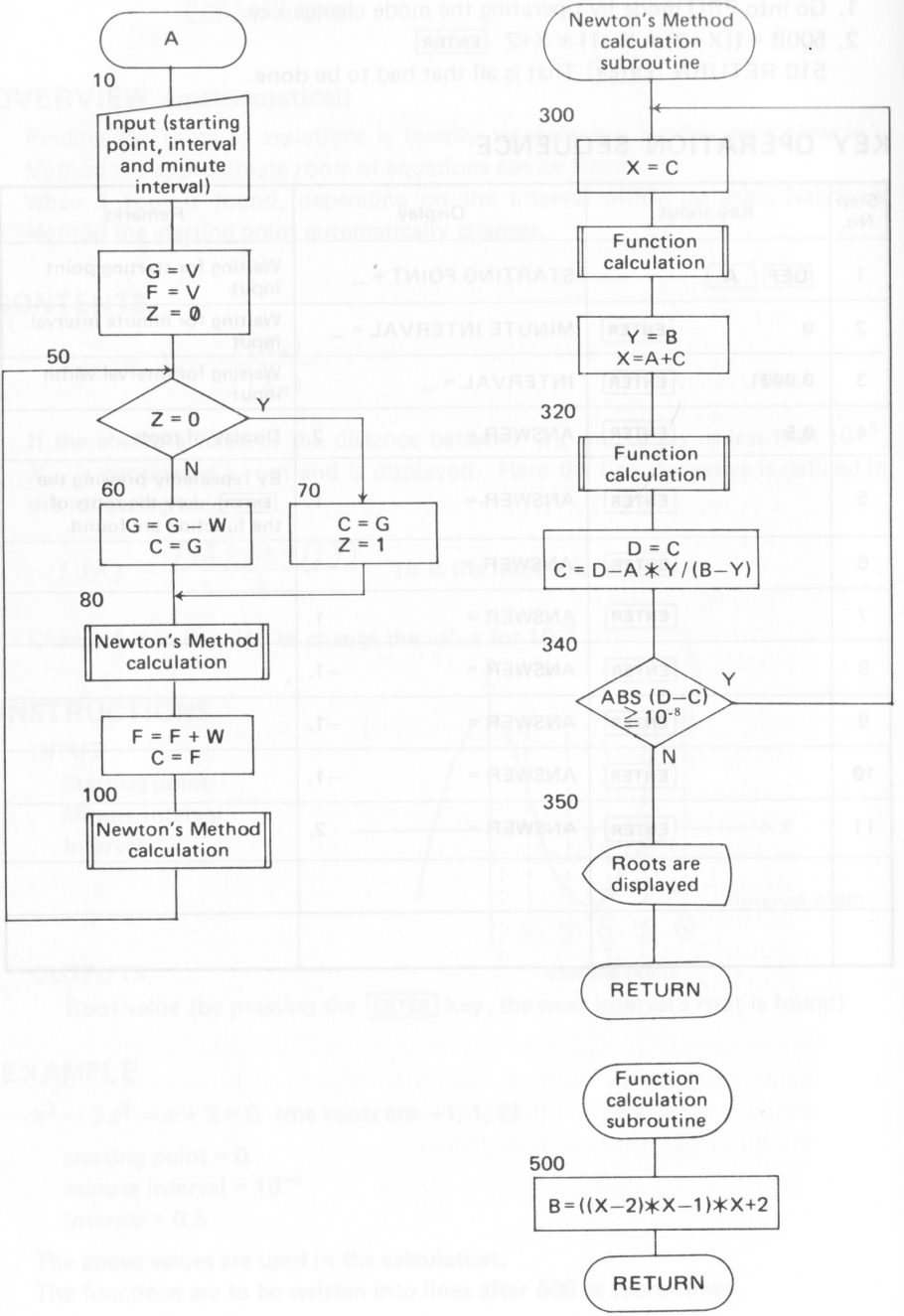
How to type in the example:

1. Go into PRO mode by operating the mode change key.
2. 500B = ((X−2) ∗ X−1) ∗ X+2 [ENTER]
   510 RETURN [ENTER] That is all that had to be done.

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	[DEF] [A]	STARTING POINT = _	Waiting for starting point input
2	0 [ENTER]	MINUTE INTERVAL = _	Waiting for minute interval input
3	0.0001 [ENTER]	INTERVAL = _	Waiting for interval width input
4	0.5 [ENTER]	ANSWER = 2.	Display of roots
5	[ENTER]	ANSWER = 1.	By repeatedly pressing the [ENTER] key the roots of the function are found.
6	[ENTER]	ANSWER = −1.	
7	[ENTER]	ANSWER = 1.	
8	[ENTER]	ANSWER = −1.	
9	[ENTER]	ANSWER = −1.	
10	[ENTER]	ANSWER = −1.	
11	[ENTER]	ANSWER = 2.	
	⋮	⋮	
	⋮	⋮	

FLOWCHART


A
10
Input (starting point, interval and minute interval)
G = V
F = V
Z = 0
50
Z = 0
Y
N
60
G = G – W
C = G
70
C = G
Z = 1
80
Newton's Method calculation
F = F + W
C = F
100
Newton's Method calculation
Newton's Method calculation subroutine
300
X = C
Function calculation
Y = B
X=A+C
320
Function calculation
D = C
C = D–A*Y/(B–Y)
340
ABS (D–C) ≧ 10^-8
Y
N
350
Roots are displayed
RETURN
Function calculation subroutine
500
B=((X–2)*X–1)*X+2
RETURN

PROGRAM LIST

```
 10:"A": INPUT "STARTING
    POINT=";V
 20:INPUT "MINUTE INTERV
    AL=";A
 30:INPUT "INTERVAL=";W
 40:G=V:F=V:Z=0
 50:IF Z=0 GOTO 70
 60:G=G-W:C=G: GOTO 80
 70:C=G:Z=1
 80:GOSUB 300
 90:F=F+W:C=F
100:GOSUB 300
110:GOTO 50
120:END
300:X=C: GOSUB 500
310:Y=B:X=A+C
320:GOSUB 500
330:D=C:C=D-A*Y/(B-Y)
340:IF ABS (D-C)>=E-8
    GOTO 300
350:BEEP 3: PRINT "ANSWE
    R=",C
360:RETURN
500:B=((X-2)*X-1)*X+2
510:RETURN

268
```

MEMORY CONTENTS

A	Minute interval
B	f (x)
C	X_0
D	f (x + h)
E	
F	√
G	√
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	Starting point
W	Interval
X	x
Y	f (x)
Z	Initial flag

Program Title: MATRIX PRODUCT

OVERVIEW

Finding matrix products

The matrix product C (mn matrix) is found for matrices A x B (ml x ln).

INSTRUCTIONS

1. The program is initiated by pressing DEF A .
   The number of rows (m) and columns (l), for an ml matrix A are input.
2. The elements of matrix A are input.
3. The number of columns (n) of matrix B are input.
4. The data is revised by pressing DEF B .
   The location to be revised and the revised value is input.
5. The elements of the resulting matrix C are printed by pressing DEF C .

REFERENCE (the calculation)

$$C_{ij} = \sum_{k=1}^{l} a_{ik} \cdot b_{kj} \qquad \begin{pmatrix} i = 1, 2, \cdots\cdots m \\ j = 1, 2, \cdots\cdots n \end{pmatrix}$$

$$m \left\{ \underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1l} \\ a_{21} & a_{22} & \cdots & a_{2l} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{ml} \end{pmatrix}}_{l} \right. \quad \underbrace{\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ b_{l1} & b_{l2} & \cdots & b_{ln} \end{pmatrix}}_{n}$$

$$= \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix}$$

$(ml + ln + mn < 38)$

EXAMPLE

$m\,l$ matrix A (4, 3)	ln matrix B (3, 2)		mn matrix C (product) (4, 2)
$\begin{pmatrix} 4 & 0 & -1 \\ -3 & 3 & 7 \\ -9 & 2 & 5 \\ 5 & -1 & 3 \end{pmatrix}$	$\begin{pmatrix} -1 & 5 \\ -6 & -6 \\ 1 & 4 \end{pmatrix}$	=	$\begin{pmatrix} -5 & 16 \\ -8 & -5 \\ 2 & -37 \\ 4 & 43 \end{pmatrix}$

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	M = _	Waiting for input of the number of rows in the *ml* matrix A
2	4 ENTER	L = _	Waiting for input of the number of columns in the *ml* matrix A
3	3 ENTER	A (1, 1) =	Waiting for input of the element (1, 1) of matrix A
		?	
4	4 ENTER	A (1, 2) =	
		?	
5	0 ENTER	A(1,3) =	
	⋮	⋮	
	⋮	A (4, 3) =	
		?	
15	3 ENTER	L = 3	
		N = _	Waiting for input of the number of columns in the *ln* matrix B
16	2 ENTER	B (1, 1) =	
		?	
17	−1 ENTER	B(1,2) =	
	⋮	⋮	
	⋮		
22	4 ENTER	>	Finished

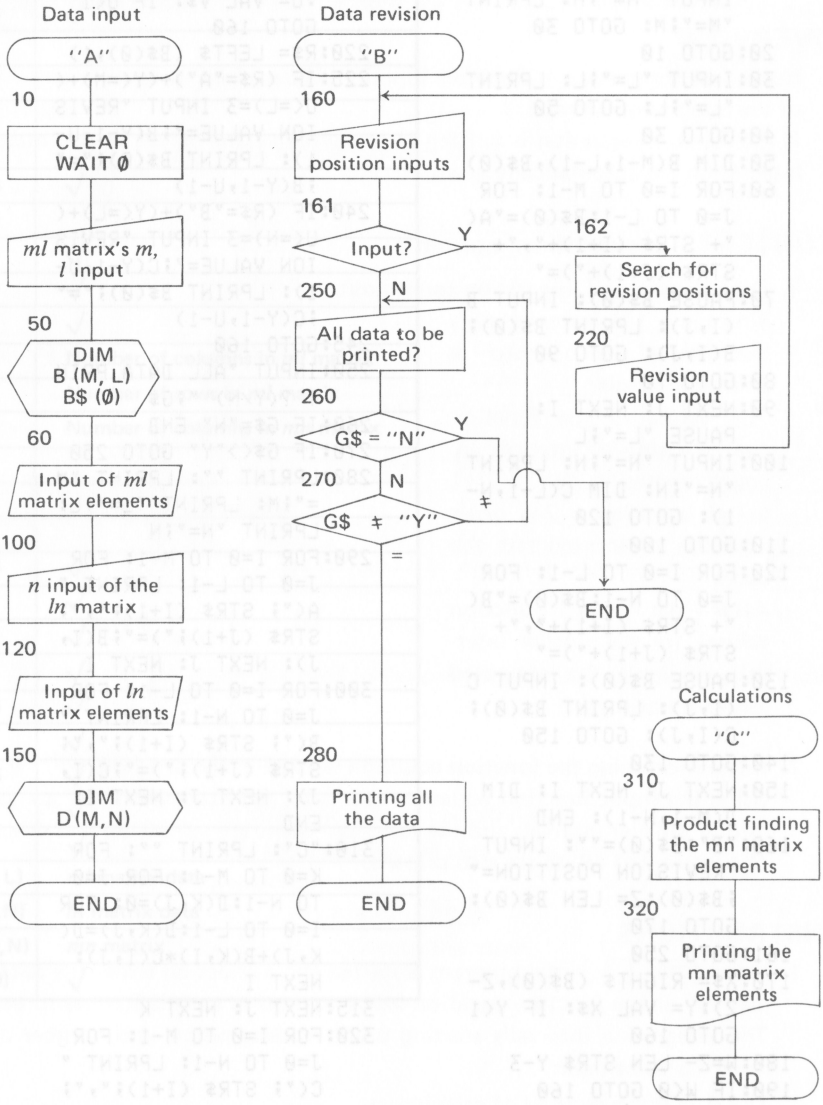
KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF B	REVISION POSITION = _	
2	A (4, 2) ENTER	REVISION VALUE = _	
3	−1 ENTER	REVISION POSITION = _	
4	ENTER	ALL DATA PRINT ? (Y/N) _	
5	Y ENTER		All data is printed
		>	
	N ENTER	>	Processing is ended without data being printed
1	DEF C		Calculation results are printed
		>	Finished

PRINTED OUTPUTS (example of ALL DATA PRINT)

```
M=4.
L=3.
N=2.
A(1,1)=4.
A(1,2)=0.
A(1,3)=-1.
A(2,1)=-3.
A(2,2)=3.
A(2,3)=7.
A(3,1)=-9.
A(3,2)=2.
A(3,3)=5.
A(4,1)=5.
A(4,2)=-1.
A(4,3)=3.
B(1,1)=-1.
B(1,2)=5.
B(2,1)=-6.
B(2,2)=-6.
B(3,1)=1.
B(3,2)=4.
C(1,1)=-5.
C(1,2)=16.
C(2,1)=-8.
C(2,2)=-5.
C(3,1)=2.
C(3,2)=-37.
C(4,1)=4.
C(4,2)=43.
```

FLOWCHART


Data input
"A"
10
CLEAR
WAIT Ø
ml matrix's m,
l input
50
DIM
B (M, L)
B$ (Ø)
60
Input of ml
matrix elements
100
n input of the
ln matrix
120
Input of ln
matrix elements
150
DIM
D(M,N)
END
Data revision
"B"
160
Revision
position inputs
161
Input?
Y
162
Search for
revision positions
220
Revision
value input
N
250
All data to be
printed?
260
G$ = "N"
Y
270
N
G$ ≠ "Y"
≠
=
END
280
Printing all
the data
END
Calculations
"C"
310
Product finding
the mn matrix
elements
320
Printing the
mn matrix
elements
END

PROGRAM LIST

```
 10:"A": CLEAR : WAIT 0:
    INPUT "M=";M: LPRINT
    "M=";M: GOTO 30
 20:GOTO 10
 30:INPUT "L=";L: LPRINT
    "L=";L: GOTO 50
 40:GOTO 30
 50:DIM B(M-1,L-1),B$(0)
 60:FOR I=0 TO M-1: FOR
    J=0 TO L-1:B$(0)="A(
    "+ STR$ (I+1)+","+
    STR$ (J+1)+")="
 70:PAUSE B$(0): INPUT B
    (I,J): LPRINT B$(0);
    B(I,J): GOTO 90
 80:GOTO 70
 90:NEXT J: NEXT I:
    PAUSE "L=";L
100:INPUT "N=";N: LPRINT
    "N=";N: DIM C(L-1,N-
    1): GOTO 120
110:GOTO 100
120:FOR I=0 TO L-1: FOR
    J=0 TO N-1:B$(0)="B(
    "+ STR$ (I+1)+","+
    STR$ (J+1)+")="
130:PAUSE B$(0): INPUT C
    (I,J): LPRINT B$(0);
    C(I,J): GOTO 150
140:GOTO 130
150:NEXT J: NEXT I: DIM
    D(M-1,N-1): END
160:"B":B$(0)="": INPUT
    "REVISION POSITION="
    ;B$(0):Z= LEN B$(0):
    GOTO 170
161:GOTO 250
170:X$= RIGHT$ (B$(0),Z-
    2):Y= VAL X$: IF Y<1
    GOTO 160
180:W=Z- LEN STR$ Y-3
190:IF W<0 GOTO 160
200:IF MID$ (B$(0),Z-W,1
    )<>"," LET W=W-1:
    GOTO 190
210:V$= RIGHT$ (B$(0),W)
    :U= VAL V$: IF U<1
    GOTO 160
220:R$= LEFT$ (B$(0),1)
225:IF (R$="A")+(Y<=M)+(
    U<=L)=3 INPUT "REVIS
    ION VALUE=";B(Y-1,U-
    1): LPRINT B$(0);"="
    ;B(Y-1,U-1)
240:IF (R$="B")+(Y<=L)+(
    U<=N)=3 INPUT "REVIS
    ION VALUE=";C(Y-1,U-
    1): LPRINT B$(0);"="
    ;C(Y-1,U-1)
245:GOTO 160
250:INPUT "ALL DATA PRIN
    T ?(Y/N) ";G$
260:IF G$="N" END
270:IF G$<>"Y" GOTO 250
280:LPRINT "": LPRINT "M
    ="; M: LPRINT "L="; L:
    LPRINT "N="; N
290:FOR I=0 TO M-1: FOR
    J=0 TO L-1: LPRINT "
    A("; STR$ (I+1);",";
    STR$ (J+1);")=";B(I,
    J): NEXT J: NEXT I
300:FOR I=0 TO L-1: FOR
    J=0 TO N-1: LPRINT "
    B("; STR$ (I+1);",";
    STR$ (J+1);")=";C(I,
    J): NEXT J: NEXT I:
    END
310:"C": LPRINT "": FOR
    K=0 TO M-1: FOR J=0
    TO N-1:D(K,J)=0: FOR
    I=0 TO L-1:D(K,J)=D(
    K,J)+B(K,I)*C(I,J):
    NEXT I
315:NEXT J: NEXT K
320:FOR I=0 TO M-1: FOR
    J=0 TO N-1: LPRINT "
    C("; STR$ (I+1);",";
    STR$ (J+1);")=";D(I,
    J): NEXT J: NEXT I:
    END
```

1079

MEMORY CONTENTS

A	
B$	√
C	
D	
E	
F	
G$	√
H	
I	√
J	√
K	√
L	Number of columns in *ml* matrix
M	Number of rows in *ml* matrix
N	Number of columns in *mn* matrix
O	
P	
Q	
R$	√
S	√
T	√
U	√
V$	√
W	√
X$	√
Y	√
Z	√
B(M,L)	*ml* matrix data
C(M,N)	*ln* matrix data
D(M,N)	*mn* matrix
B$(Ø)	√

Program Title: NUMERICAL INTEGRATION USING SIMPSON'S RULE

OVERVIEW

Numerical Integration is done on function values given at equal interval widths of the integration interval.

If the function equation is written in the program the values in the intervals of integration are automatically given.

CONTENTS (calculation contents)

1. **Data is input and integration carried out.**

   Simpson's 1/3 formula splits the interval [a, b] into n smaller intervals.

   The values of the function over the smaller intervals are approximated in 2's ($2i$, $2i$+1 units) by using a 2nd order equation to approximate the curve.

   After the data (function values) for the smaller intervals are input, the integrated values are printed.

$$\int_a^b f(x)dx \fallingdotseq \sum_{i=0}^{N/2-1} \int_{x_{2i}}^{x_{2i+2}} P_2^i(x)\,dx$$

$$= \sum_{i=0}^{N/2-1} I_i$$

$$\fallingdotseq \frac{h}{3}(y_0+4y_1+2y_2+4y_3+\cdots$$

$$\cdots+4y_{-1}+y_n)$$

$$\begin{cases} h = \dfrac{b-a}{n} \\ I_i = \dfrac{h}{3}(y_{2i}+4y_{2i+1}+y_{2i+2}) \end{cases}$$

2. **Integration using the function equation written in the program**

   Using the input function equation as a base, the intervals [a, b] are split up into n smaller intervals and the function values are calculated and printed. The integrated values are also printed.

INSTRUCTIONS

1. Using DEF A , the program is started and a selection either the data input method or the function equation input method of calculation has to be done. The integration interval's starting point, ending point, and number of divisions has to be input.

   The data input method of calculating:

   as the integration intervals are being input the data is printed.

   The function equation input method of calculating:

   the function values are printed according to the function equation.

2. The DEF B corrects the input data as required. Enter revision number and revision value.

3. DEF C The integration values are printed according to the integration interval data (function values).

(Note) The number of divisions is up to 74 and must be a even number.

EXAMPLE

1. The data is input and the calculations are done

interval [0, 5], 40 divisions

$f(x\ 0)$	4	$f(x11)$	−7	$f(x22)$	−2	$f(x33)$	13
$f(x\ 1)$	5.5	$f(x12)$	−8	$f(x23)$	0	$f(x34)$	12.5
$f(x\ 2)$	6	$f(x13)$	−9	$f(x24)$	2	$f(x35)$	12
$f(x\ 3)$	5.7	$f(x14)$	−9.5	$f(x25)$	4	$f(x36)$	10.5
$f(x\ 4)$	5	$f(x15)$	−10	$f(x26)$	6	$f(x37)$	9
$f(x\ 5)$	2	$f(x16)$	−9.5	$f(x27)$	7	$f(x38)$	7.8
$f(x\ 6)$	0	$f(x17)$	−9	$f(x28)$	8	$f(x39)$	6
$f(x\ 7)$	−1.8	$f(x18)$	−8.5	$f(x\ 29)$	9.7	$f(x40)$	4
$f(x\ 8)$	−3	$f(x19)$	−7	$f(x\ 30)$	11		
$f(x\ 9)$	−5	$f(x20)$	−5.5	$f(x\ 31)$	12		
$f(x10)$	−6	$f(x21)$	−4	$f(x\ 32)$	12.5		

2. The function equation is stored in the program and the calculations are done.

$$Y = ((X-2)X-1)X+2$$

interval [0, 1] 20 divisions

The function is stored after line 900 as a subroutine.

How to store into the program (for the case of the example)

(Put in PRO position)

900 Y = ((X−2)∗X−1)∗ X+2 ENTER

910 RETURN ENTER this ends the input

PRINTED OUTPUTS

```
A=0.
B=5.
N=40.
  1 F(0)=4.
  2 F(0.125)=5.5
  3 F(0.25)=6.
  4 F(0.375)=5.7
  5 F(0.5)=5.
  6 F(0.625)=2.
  7 F(0.75)=0.
  8 F(0.875)=-1.8
  9 F(1)=-3.
 10 F(1.125)=-5.
 11 F(1.25)=-6.
 12 F(1.375)=-7.
 13 F(1.5)=-8.
 14 F(1.625)=-9.
 15 F(1.75)=-9.5
 16 F(1.875)=-10.
 17 F(2)=-9.5
 18 F(2.125)=-9.
 19 F(2.25)=-8.5
 20 F(2.375)=-7.
 21 F(2.5)=-5.5
 22 F(2.625)=-4.
 23 F(2.75)=-2.
 24 F(2.875)=0.
 25 F(3)=2.
 26 F(3.125)=4.
 27 F(3.25)=6.
 28 F(3.375)=7.
 29 F(3.5)=8.
 30 F(3.625)=9.7
 31 F(3.75)=11.
 32 F(3.875)=12.
 33 F(4)=12.5
 34 F(4.125)=13.
 35 F(4.25)=12.5
 36 F(4.375)=12.
 37 F(4.5)=10.5
 38 F(4.625)=9.
 39 F(4.75)=7.8
 40 F(4.875)=6.
 41 F(5)=4.


F=8.291666667
```

```
A=0.
B=1.
N=20.

  1 F(0)=2.
  2 F(0.05)=1.945125
  3 F(0.1)=1.881
  4 F(0.15)=1.808375
  5 F(0.2)=1.728
  6 F(0.25)=1.640625
  7 F(0.3)=1.547
  8 F(0.35)=1.447875
  9 F(0.4)=1.344
 10 F(0.45)=1.236125
 11 F(0.5)=1.125
 12 F(0.55)=1.011375
 13 F(0.6)=0.896
 14 F(0.65)=0.779625
 15 F(0.7)=0.663
 16 F(0.75)=0.546875
 17 F(0.8)=0.432
 18 F(0.85)=0.319125
 19 F(0.9)=0.209
 20 F(0.95)=0.102375
 21 F(1)=0.

F=1.083333333
```

KEY OPERATION SEQUENCE

(when the values are input)

Step No.	Key Input	Display	Remarks
1	DEF A	F(X):INPUT=1/CAL.=2?_	Selection of data input or equation input
2	1 ENTER	A = _	Waiting for the starting point of the integration interval
3	Ø ENTER	B = _	Waiting for the ending point of the integration interval
4	5 ENTER	N = _	Waiting for the integration interval division number
5	4Ø ENTER	F(Ø) =	Waiting for the data to be input
		?	
6	4 ENTER	F(Ø.125) =	
		?	
7	5.5 ENTER	F(Ø.25) =	
		?	
	⋮	⋮	
	6 ENTER	F(5) =	
		?	
	4 ENTER	>	

(Data revision)

Step No.	Key Input	Display	Remarks
1	DEF B	REVISION NO. ? = _	Waiting for the revision number input
2	3 ENTER	REVISION VALUE = _	Waiting for the revision value input
3	6 ENTER	REVISION NO. ? = _	
4	ENTER	ALL DATA PRINT ?(Y/N) _	
5	Y ENTER	>	All of the data printed out
	N ENTER	>	Finished without printing

(to be continued to DEF C)

KEY OPERATION SEQUENCE

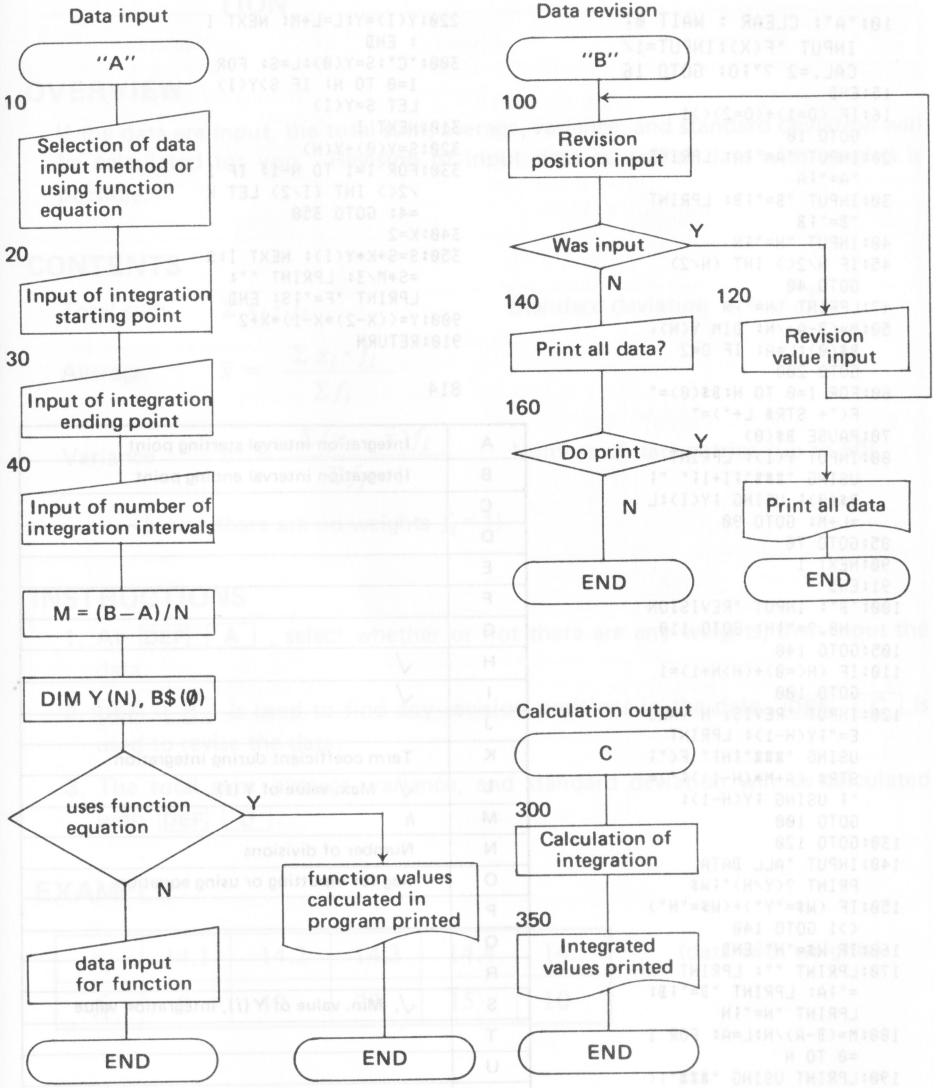
(when the function equation is used)

Step No.	Key Input	Display	Remarks
1	DEF A	F(X):INPUT=1/CAL.=2?_	Waiting for the data input method or the equation input method of calculation
2	2 ENTER	A = _	Waiting for the integration starting point input
3	Ø ENTER	B = _	Waiting for the integration ending point input
4	1 ENTER	N = _	Waiting for the integration interval division number
5	2Ø ENTER		Prints the calculations done to find the function values using the function equation as the base
		>	

(final procedure in both cases)

Step No.	Key Input	Display	Remarks
1	DEF C		Integrated values printed
		>	

FLOWCHART


Data input
"A"
10
Selection of data input method or using function equation
20
Input of integration starting point
30
Input of integration ending point
40
Input of number of integration intervals
M = (B−A)/N
DIM Y(N), B$(Ø)
uses function equation
Y
N
function values calculated in program printed
data input for function
END
END
Data revision
"B"
100
Revision position input
Was input
Y
N
140
120
Print all data?
Revision value input
160
Do print
Y
N
Print all data
END
END
Calculation output
C
300
Calculation of integration
350
Integrated values printed
END

PROGRAM LIST

```
10:"A": CLEAR : WAIT 0:
   INPUT "F(X):INPUT=1/
   CAL.=2 ?";O: GOTO 16
15:END
16:IF (O=1)+(O=2)<>1
   GOTO 10
20:INPUT "A=";A: LPRINT
   "A=";A
30:INPUT "B=";B: LPRINT
   "B=";B
40:INPUT "N=";N
45:IF N/2<> INT (N/2)
   GOTO 40
47:LPRINT "N=";N
50:M=(B-A)/N: DIM Y(N),
   B$(0):L=A: IF O=2
   GOTO 200
60:FOR I=0 TO N:B$(0)="
   F("+ STR$ L+")="
70:PAUSE B$(0)
80:INPUT Y(I): LPRINT
   USING "###";I+1;" ";
   B$(0); USING ;Y(I):L
   =L+M: GOTO 90
85:GOTO 70
90:NEXT I
91:END
100:"B": INPUT "REVISION
    N0.?=";H: GOTO 110
105:GOTO 140
110:IF (H<=0)+(H>N+1)=1
    GOTO 100
120:INPUT "REVISION VALU
    E=";Y(H-1): LPRINT
    USING "###";H;" F(";
    STR$ (A+M*(H-1));")=
    "; USING ;Y(H-1):
    GOTO 100
130:GOTO 120
140:INPUT "ALL DATA
    PRINT ?(Y/N)";W$
150:IF (W$="Y")+(W$="N")
    <>1 GOTO 140
160:IF W$="N" END
170:LPRINT "": LPRINT "A
    =";A: LPRINT "B=";B:
    LPRINT "N=";N
180:M=(B-A)/N:L=A: FOR I
    =0 TO N
190:LPRINT USING "###";(
    I+1);" F("; STR$ L;"
    )="; USING ;Y(I)
195:L=L+M: NEXT I: END
200:LPRINT "": FOR I=0
    TO N:X=L: GOSUB 900
210:LPRINT USING "###";(
    I+1);" F("; STR$ L;"
    )="; USING ;Y
220:Y(I)=Y:L=L+M: NEXT I
    : END
300:"C":S=Y(0):L=S: FOR
    I=0 TO N: IF S>Y(I)
    LET S=Y(I)
310:NEXT I
320:S=Y(0)+Y(N)
330:FOR I=1 TO N-1: IF I
    /2<> INT (I/2) LET K
    =4: GOTO 350
340:K=2
350:S=S+K*Y(I): NEXT I:S
    =S*M/3: LPRINT "":
    LPRINT "F=";S: END
900:Y=((X-2)*X-1)*X+2
910:RETURN
```

814

A	Integration interval starting point
B	Integration interval ending point
C	
D	
E	
F	
G	
H	√
I	√
J	
K	Term coefficient during integration
L	√ Max. value of Y (i)
M	h
N	Number of divisions
O	flag for inputting or using equation
P	
Q	
R	
S	√, Min. value of Y (i), integration value
T	
U	
V	
W$	√
X	√ Function equation X
Y	Function equation Y
Z	
Y (N)	Input data (function value)
B$ (0)	√

Program Title: AVERAGE, VARIANCE AND STANDARD DEVIATION

OVERVIEW

If the data are input, the total sum, average, variance, and standard deviation will be calculated for you. Revision of input data as well as data with weights is possible.

CONTENTS

Total sum $\Sigma x_i \cdot f_i$

Average $\bar{x} = \dfrac{\Sigma x_i \cdot f_i}{\Sigma f_i}$

Variance $\sigma^2 = \dfrac{\Sigma (x_i - \bar{x}) f_i}{\Sigma f_i - 1}$

(when there are no weights $f_i = 1$)

Standard deviation $\sigma = \sqrt{\sigma^2}$

Number of data entries (up to 50)

INSTRUCTIONS

1. At [DEF] [A] , select whether or not there are any weights, then input the data.
2. [DEF] [B] is used to find any revision positions in the data. [DEF] [C] is used to revise the data.
3. The total sum, average, variance, and standard deviation will be calculated with [DEF] [D] .

EXAMPLE

x_i	14.1	14.2	14.3	14.4	14.5
f_i	8	19	23	15	10

(data with weights)

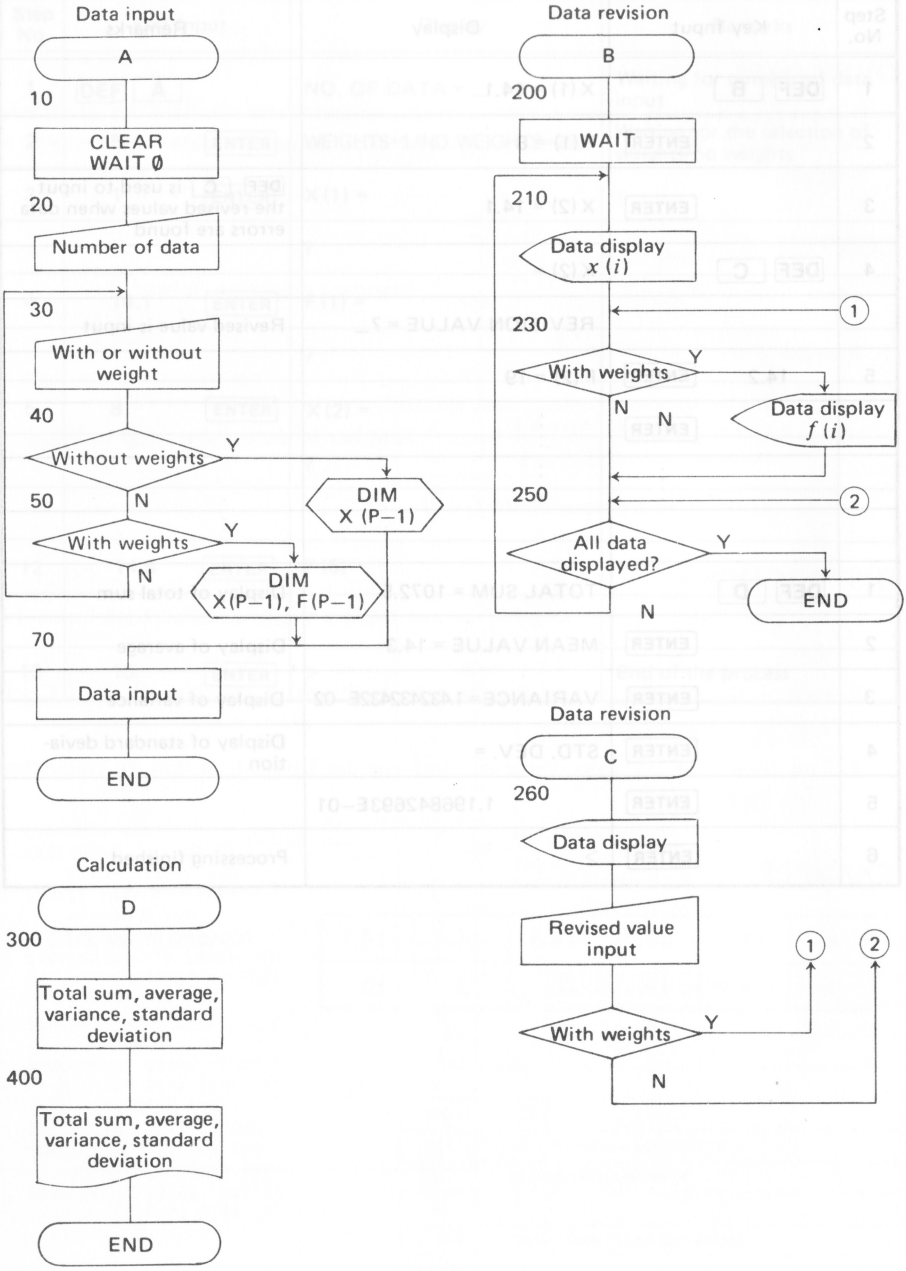
KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	NO. OF DATA = _	Waiting for number of data input
2	5 ENTER	WEIGHTS=1/NO WEIGHTS=2?_	Waiting for the selection of weights/no weights
3	1 ENTER	X(1) =	
		?	
4	14.1 ENTER	F(1) =	
		?	
5	8 ENTER	X(2) =	
		?	
	⋮	⋮	
12	14.5 ENTER	F(5) =	
		?	
13	10 ENTER	>	End of the process

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF B	X(1) = 14.1	
2	ENTER	F(1) = 8	
3	ENTER	X(2) = 14.1	DEF C is used to input the revised values when data errors are found
4	DEF C	X(2) =	
		REVISION VALUE = ?_	Revised value is input
5	14.2 ENTER	F(2) = 19	
	ENTER	⋮	
	⋮	⋮	
1	DEF D	TOTAL SUM = 1072.5	Display of total sum
2	ENTER	MEAN VALUE = 14.3	Display of average
3	ENTER	VARIANCE = 1.432432432E−02	Display of variance
4	ENTER	STD. DEV. =	Display of standard deviation
5	ENTER	1.196842693E−01	
6	ENTER	>	Processing finished

FLOWCHART


Data input
A
10
CLEAR
WAIT Ø
20
Number of data
30
With or without weight
40
Without weights
Y
N
DIM
X (P−1)
50
With weights
Y
N
DIM
X(P−1), F(P−1)
70
Data input
END
Data revision
B
200
WAIT
210
Data display
x (i)
1
230
With weights
Y
N
N
Data display
f (i)
250
2
All data displayed?
Y
N
END
Data revision
C
260
Data display
Revised value input
1
2
With weights
Y
N
Calculation
D
300
Total sum, average, variance, standard deviation
400
Total sum, average, variance, standard deviation
END

PROGRAM LIST

```
10:"A": CLEAR : WAIT 0
20:INPUT "NO. OF DATA="
   ;P
30:INPUT "WEIGHTS=1/NO
   WEIGHTS=2?";A
40:IF A=2 DIM X(P-1):
   GOTO 70
50:IF A=1 DIM X(P-1),F(
   P-1): GOTO 70
60:GOTO 30
70:FOR I=0 TO P-1
80:B$="X("+ STR$ (I+1)+
   ")="
85:PAUSE B$: INPUT X(I)
   : GOTO 100
90:GOTO 85
100:IF A=2 GOTO 150
120:B$="F("+ STR$ (I+1)+
   ")="
130:PAUSE B$: INPUT F(I)
   : GOTO 150
140:GOTO 130
150:NEXT I: END
200:"B": WAIT :I=0
210:B$="X("+ STR$ (I+1)+
   ")=":J=1: PRINT B$;X
   (I)
230:IF A=1 LET B$="F("+
   STR$ (I+1)+")=":
   PRINT B$;F(I):J=2
240:I=I+1
250:IF I=P END
255:GOTO 210
260:"C": PAUSE B$: IF
   LEFT$ (B$,1)="X"
   INPUT "REVISION VALU
   E=";X(I): GOTO 290
270:IF LEFT$ (B$,1)="F"
   INPUT "REVISION VALU
   E=";F(I): GOTO 290
280:GOTO 250
290:IF J=1 GOTO 230
291:GOTO 210
300:"D":N=0:T=0:S=0: FOR
   I=0 TO P-1:X=X(I)
305:F=1: IF A=1 LET F=F(
   I)
310:N=N+F:T=T+F*X:S=S+F*
   X*X: NEXT I
400:WAIT :X=T/N:Q=(S-N*X
   *X)/(N-1):S=√Q:
   PRINT "TOTAL SUM=";T
   : PRINT "MEAN VALUE=
   ";X
410:PRINT "VARIANCE=";Q:
   PRINT "STD. DEV.=":
   PRINT S: END
```

643

MEMORY CONTENTS

A	√
B$	√
C	
D	
E	
F	√
G	
H	
I	√
J	Flag
K	
L	
M	
N	√
O	
P	Data number
Q	Variance
R	
S	Standard deviation
T	Total sum
U	
V	
W	
X	Average
Y	
Z	
X (P−1)	Data
F (P−1)	Data

Program Title: CORRELATION COEFFICIENT AND LINEAR REGRESSION

OVERVIEW (Statistics)

Data is for analysis and testing hypotheses.

This program finds the covariance and correlation coefficient for related data sets $(X1, Y1) \cdots (Xn, Yn)$, as well as the linear regression.

The following data input is put into the equation $Y = AX + B$, for output to the printer.

INSTRUCTIONS

1. Data input (X_i, Y_i).
2. Correction of mistaken data.
3. The covariance, correlation coefficient, regression coefficients and means are found and output to the printer.
4. Y is estimated from the X value and output to the printer.
5. The limit for the number of data entries is 100.

CALCULATION

$$S_{xx} = \Sigma xi^2 - n\bar{x}^2$$

$$S_{xy} = \Sigma xiyi - n\bar{x}\bar{y}$$

$$S_{yy} = \Sigma yi^2 - n\bar{y}^2$$

$$C = S_{xy}/(n-1) \quad \cdots\cdots\cdots\cdots \text{ covariance}$$

$$r = S_{xy}/\sqrt{S_{xx}\,S_{yy}} \quad \cdots\cdots\cdots \text{ correlation coefficient}$$

$$\left.\begin{aligned} a &= S_{xy}/S_{xx} \\ b &= \bar{y} - a\bar{x} \end{aligned}\right\} \text{ regression coefficient } (y = ax + b)$$

EXAMPLE

X	6.9	7.6	7.6	9.0	8.1	6.5	6.4	6.9
Y	12	10	9	5	6	15	14	12

Estimated value

X = 7,

X = 8,

X = 7.5,

PRINTED OUTPUTS

```
NO. OF DATA=8.
X(1)=6.9
Y(1)=12.
X(2)=7.6
Y(2)=10.
X(3)=7.6
Y(3)=9.
X(4)=9.
Y(4)=5.
X(5)=8.1
Y(5)=6.
X(6)=6.5
Y(6)=15.
X(7)=6.4
Y(7)=14.
X(8)=6.9
Y(8)=12.
COVARIANCE=
                -3.060714286
CORRELATION COEFFICIENT=
          -9.693968513E-01
*REGRESSION COEFFICIENT*
 A=-3.942042318
 B=39.4475621
*MEAN VALUE*
 X=7.375
 Y=10.375

*ESTIMATION*
 X=7.
 Y=11.85326587
 X=8.
 Y=7.911223556
 X=7.5
 Y=9.882244715
```

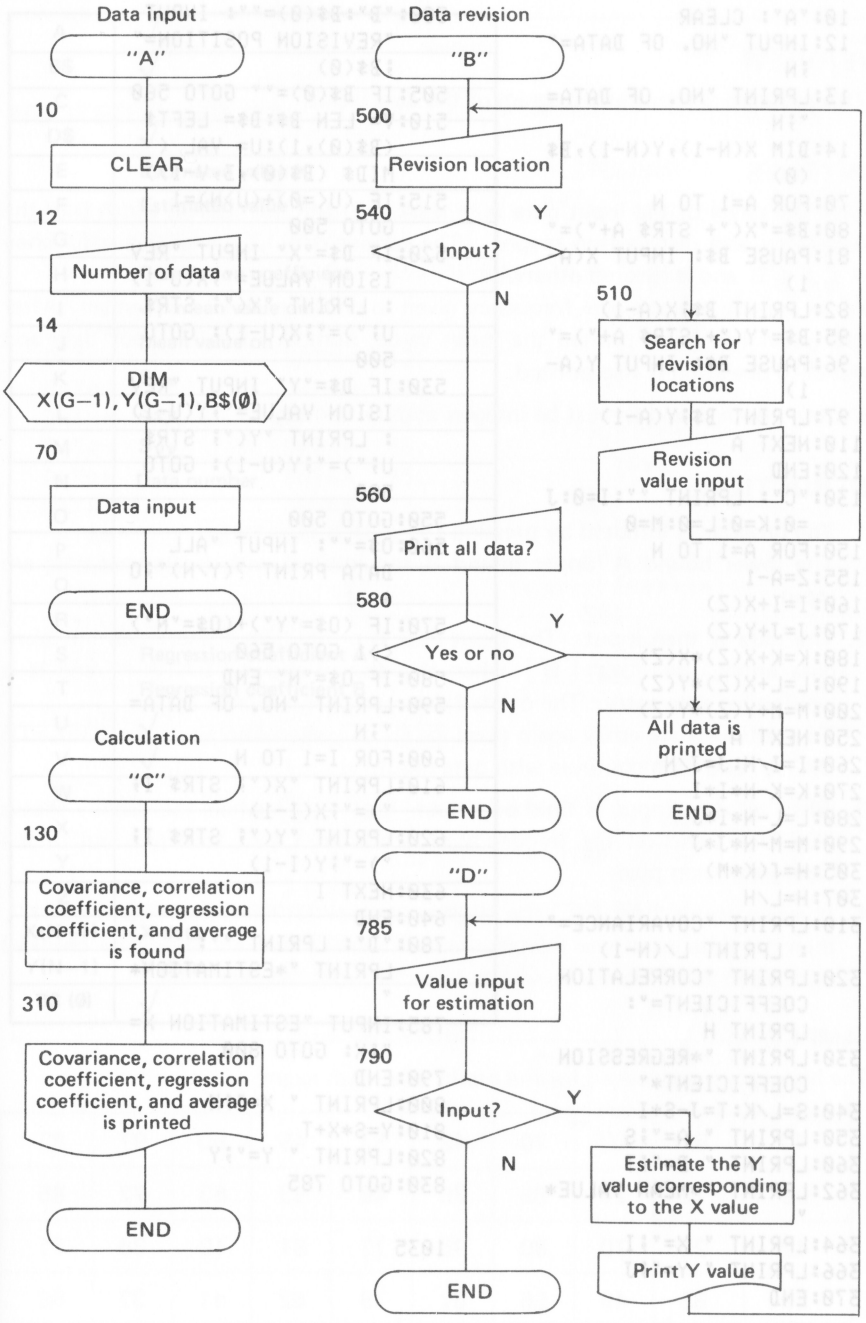
KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	NO. OF DATA = _	
2	8 ENTER	X(1) =	Data input
		?	
3	6.9 ENTER	Y(1) =	
		?	
4	12 ENTER	X(2) =	
		?	
	⋮	⋮	
	⋮	⋮	
		X(8) =	
		?	
17	6.9 ENTER	Y(8) =	
		?	
18	12 ENTER	>	

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF B	REVISION POSITION = _	
2	X(2) ENTER	REVISION VALUE = _	
3	7.6 ENTER	REVISION POSITION = _	
4	ENTER	ALL DATA PRINT ?(Y/N) _	
5	Y ENTER		Output all data to printer
		>	
	N ENTER	>	Finish without having data printed
1	DEF C		Covariance, Correlation coefficient, regression coefficient A, B and mean values X, Y are calculated and printed
		>	
1	DEF D	ESTIMATION X = _	Y estimated from the value X and printed
2	7 ENTER	ESTIMATION X = _	
3	8 ENTER	⋮ Repeat	
⋮	⋮	⋮	
	ENTER	>	If only ENTER is pressed, finished

FLOWCHART


Data input
"A"
10
CLEAR
12
Number of data
14
DIM
X(G−1),Y(G−1),B$(Ø)
70
Data input
END
Data revision
"B"
500
Revision location
540
Input?
Y
N
510
Search for
revision
locations
Revision
value input
560
Print all data?
580
Yes or no
Y
N
All data is
printed
END
END
Calculation
"C"
130
Covariance, correlation
coefficient, regression
coefficient, and average
is found
310
Covariance, correlation
coefficient, regression
coefficient, and average
is printed
END
"D"
785
Value input
for estimation
790
Input?
Y
N
Estimate the Y
value corresponding
to the X value
Print Y value
END

PROGRAM LIST

```
 10:"A": CLEAR
 12:INPUT "NO. OF DATA="
    ;N
 13:LPRINT "NO. OF DATA=
    ";N
 14:DIM X(N-1),Y(N-1),B$
    (0)
 70:FOR A=1 TO N
 80:B$="X("+ STR$ A+")="
 81:PAUSE B$: INPUT X(A-
    1)
 82:LPRINT B$;X(A-1)
 95:B$="Y("+ STR$ A+")="
 96:PAUSE B$: INPUT Y(A-
    1)
 97:LPRINT B$;Y(A-1)
110:NEXT A
120:END
130:"C": LPRINT "":I=0:J
    =0:K=0:L=0:M=0
150:FOR A=1 TO N
155:Z=A-1
160:I=I+X(Z)
170:J=J+Y(Z)
180:K=K+X(Z)*X(Z)
190:L=L+X(Z)*Y(Z)
200:M=M+Y(Z)*Y(Z)
250:NEXT A
260:I=I/N:J=J/N
270:K=K-N*I*I
280:L=L-N*I*J
290:M=M-N*J*J
305:H=√(K*M)
307:H=L/H
310:LPRINT "COVARIANCE="
    : LPRINT L/(N-1)
320:LPRINT "CORRELATION
    COEFFICIENT=":
    LPRINT H
330:LPRINT "*REGRESSION
    COEFFICIENT*"
340:S=L/K:T=J-S*I
350:LPRINT " A=";S
360:LPRINT " B=";T
362:LPRINT "*MEAN VALUE*
    "
364:LPRINT " X=";I
366:LPRINT " Y=";J
370:END
500:"B":B$(0)="": INPUT
    "REVISION POSITION="
    ;B$(0)
505:IF B$(0)="" GOTO 560
510:V= LEN B$:D$= LEFT$
    (B$(0),1):U= VAL (
    MID$ (B$(0),3,V-1))
515:IF (U<=0)+(U>N)=1
    GOTO 500
520:IF D$="X" INPUT "REV
    ISION VALUE=";X(U-1)
    : LPRINT "X("; STR$
    U;")=";X(U-1): GOTO
    500
530:IF D$="Y" INPUT "REV
    ISION VALUE=";Y(U-1)
    : LPRINT "Y("; STR$
    U;")=";Y(U-1): GOTO
    500
550:GOTO 500
560:O$="": INPUT "ALL
    DATA PRINT ?(Y/N)";O
    $
570:IF (O$="Y")+(O$="N")
    <>1 GOTO 560
580:IF O$="N" END
590:LPRINT "NO. OF DATA=
    ";N
600:FOR I=1 TO N
610:LPRINT "X("; STR$ I;
    ")=";X(I-1)
620:LPRINT "Y("; STR$ I;
    ")=";Y(I-1)
630:NEXT I
640:END
780:"D": LPRINT "":
    LPRINT "*ESTIMATION*
    "
785:INPUT "ESTIMATION X=
    ";X: GOTO 800
790:END
800:LPRINT " X=";X
810:Y=S*X+T
820:LPRINT " Y=";Y
830:GOTO 785


1035
```

MEMORY CONTENTS

A	√
B$	√
C	
D$	√
E	Estimated value X
F	Estimated value Y
G	
H	Correlation coefficient
I	√, mean value on X
J	Mean value on Y
K	S_{xx}
L	S_{xy}
M	S_{yy}
N	Data number
O	
P	
Q	
R	
S	Regression coefficient A
T	Regression coefficient B
U	√
V	√
W	
X	
Y	
Z	
X(N−1)	X data
Y(N−1)	Y data
B$ (Ø)	√

Program Title: HISTOGRAM

OVERVIEW

This program graphs the histogram of the data input.

CONTENTS

Give the range of the input data A ~ B (A < B) and the interval width D of the frequency and distribution graphs. The various data X_i are valid only when $A \leqq X \leqq B$ and is ignored otherwise.

When making a graph, the frequency given to one asterisk (*) is first set. After the graph has been printed the mean value, number of valid input data and standard deviation is also printed.

(Note) Both A and B must be integers and up to 4 digits.

INSTRUCTIONS

1. The program is initiated by pressing [R] [U] [N] [ENTER] .
   The lower bound A, upper bound B and interval with (scale unit size) are input.
2. The data is then input. The input data is printed, so if there was mistaken data entry, use [DEF] [B] when the display shows "DATA =" and it is waiting for data entry. The deleted data can be input once again.
   To continue data entry again press [DEF] [A] when display shows "DELETE DATA" and then continue with data input.
3. When all data input is finished, input [DEF] [C] when the display shows "DATA =" and set the frequency of one asterisk (∗), then printout of the histogram takes place.
4. If when using [DEF] [B] the display shows "OVER-DELETED CHECK", then the data input was mistaken and the entire data set should be checked once again and program should be restarted from the beginning.

EXAMPLE

The historgram are desired for the marks of a math exam.

78	92	63	70	42	53	45	60	97	82
98	12	24	85	36	49	53	83	72	85
42	23	70	80	95	77	81	19	36	71
29	63	49	55	67	78	62	41	32	68

Range 0 ~ 100

Interval 10

"*" = 1

PRINTED OUTPUTS

```
INPUT DATA
DATA=  78.
DATA=  92.
DATA=  63.
DATA=  70.
DATA=  42.
DATA=  53.
DATA=  45.
DATA=  60.
DATA=  97.
DATA=  82.
DATA=  98.
DATA=  12.
DATA=  24.
DATA=  85.
DATA=  36.
DATA=  49.
DATA=  49.

DELETED DATA
**  49.
```

```
INPUT DATA
DATA=  53.
DATA=  83.
DATA=  72.
DATA=  85.
DATA=  42.
DATA=  23.
DATA=  70.
DATA=  80.
DATA=  95.
DATA=  77.
DATA=  81.
DATA=  19.
DATA=  36.
DATA=  71.
DATA=  29.
DATA=  63.
DATA=  49.
DATA=  55.
DATA=  67.
DATA=  87.
DATA=  78.
DATA=  62.
DATA=  40.
DATA=  41.
DATA=  32.
DATA=  68.

DELETED DATA
**  87.
**  40.
```

```
HISTOGRAM
*=1.
          0    5   10   15
          +----+----+----+---
        0I
          I
       10I                    2
          I**
       20I                    3
          I***
       30I                    3
          I***
       40I                    6
          I******
       50I                    3
          I***
       60I                    6
          I******
       70I                    7
          I*******
       80I                    6
          I******
       90I                    4
          I****
      100I
          I
      100+----+----+----+---

NO. OF DATA= 40.
MEAN VALUE= 60.425
STD. DEV.= 22.82639645
```

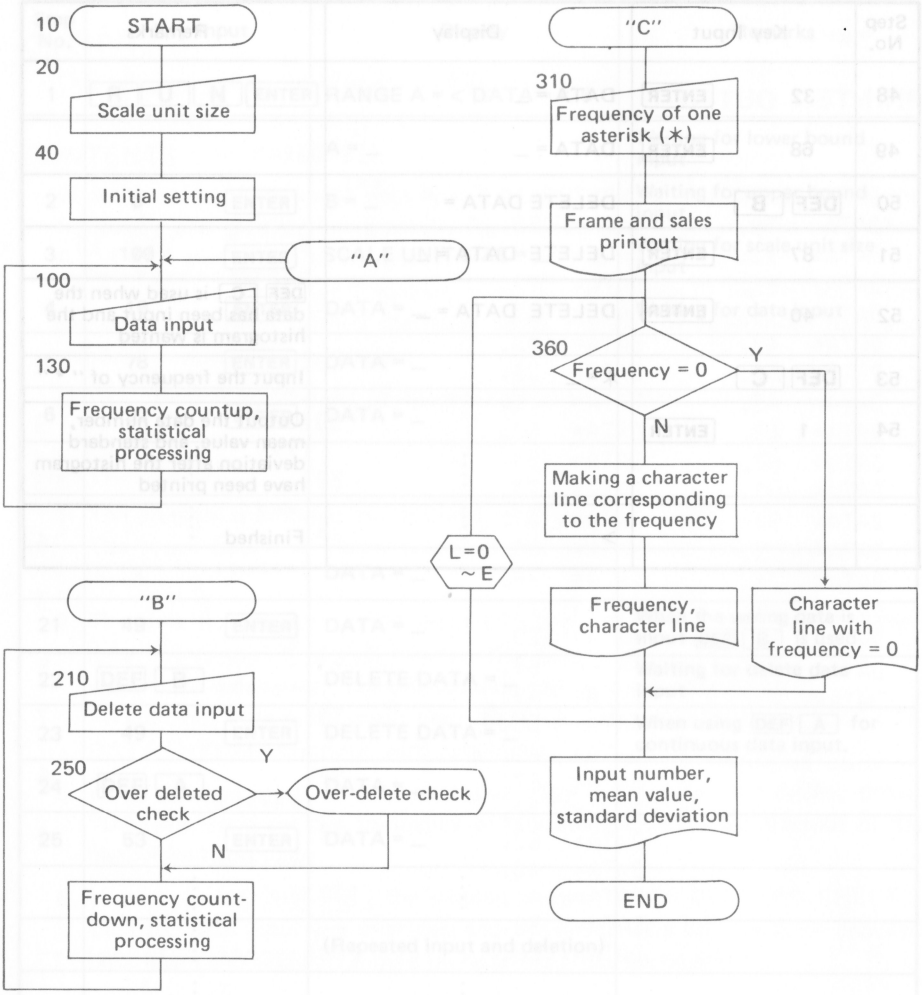
KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	R U N ENTER	RANGE A = < DATA = < B	
		A = _	Waiting for lower bound input
2	0 ENTER	B = _	Waiting for upper bound input
3	100 ENTER	SCALE UNIT SIZE = _	Waiting for scale unit size input
4	10 ENTER	DATA = _	Waiting for data input
5	78 ENTER	DATA = _	
6	92 ENTER	DATA = _	
	⋮	⋮	
	⋮	⋮	
		DATA = _	
21	49 ENTER	DATA = _	When the wrong data is input DEF B is used
22	DEF B	DELETE DATA = _	Waiting for delete data input
23	49 ENTER	DELETE DATA = _	When using DEF A for continuous data input.
24	DEF A	DATA = _	
25	53 ENTER	DATA = _	
	⋮	⋮	
	⋮	(Repeated input and deletion)	
	⋮	⋮	

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
48	32 [ENTER]	DATA = _	
49	68 [ENTER]	DATA = _	
50	[DEF] [B]	DELETE DATA =	
51	87 [ENTER]	DELETE DATA = _	
52	40 [ENTER]	DELETE DATA = _	[DEF] [C] is used when the data has been input and the histogram is wanted
53	[DEF] [C]	✳ = _	Input the frequency of " * "
54	1 [ENTER]		Output the data number, mean value, and standard deviation after the histogram have been printed
		>	Finished

FLOWCHART


10
START
20
Scale unit size
40
Initial setting
"A"
100
Data input
130
Frequency countup, statistical processing
"B"
210
Delete data input
250
Over deleted check
Y
Over-delete check
N
Frequency count-down, statistical processing
"C"
310
Frequency of one asterisk (*)
Frame and sales printout
360
Frequency = 0
Y
N
Making a character line corresponding to the frequency
L=0 ~ E
Frequency, character line
Character line with frequency = 0
Input number, mean value, standard deviation
END

PROGRAM LIST

```
 10:CLEAR : WAIT 100
 20:PRINT "RANGE  A =< D
    ATA =< B": INPUT "A=
    ";A;"B=";B
 30:INPUT "SCALE UNIT SI
    ZE=";D
 40:C=B-A:E= INT (C/D)
 60:DIM D(E),B$(0)*24
 70:LPRINT "INPUT DATA"
 80:K=1: WAIT 0:F=0
 90:"A" USING : IF F>0
    LET F=0: LPRINT "":
    LPRINT "INPUT DATA"
100:X=-E99
110:INPUT "DATA=";X
120:IF (X<A)+(X>B)=1
    GOTO 100
125:LPRINT "DATA=  ";X
130:N= INT ((X-A)/D)
140:D(N)=D(N)+1
150:U=U+X:V=V+X*X
160:K=K+1
170:GOTO 100
200:"B" USING : IF F<>1
    LET F=1: LPRINT "":
    LPRINT "DELETED DATA
    "
210:INPUT "DELETE DATA="
    ;X
230:IF (X<A)+(X>B)=1
    GOTO 210
235:LPRINT "**  ";X
240:N= INT ((X-A)/D)
250:IF D(N)-1<0 BEEP 2:
    WAIT : PRINT "OVER-D
    ELETED CHECK"
260:D(N)=D(N)-1:K=K-1
265:U=U-X:V=V-X*X
270:GOTO 210
300:"C"F=2: WAIT 100
305:LPRINT "": LPRINT ""
    : LPRINT "HISTOGRAM"
310:P=1: INPUT "*= ";P:
    LPRINT "*=";P
320:LPRINT ""
330:LPRINT " "; USING "#
    ####";0;5*P;10*P;15*
    P
340:LPRINT "     +----+-
    ---+----+---"
345:H=A
350:FOR L=0 TO E
360:IF D(L)=0 GOTO 440
370:Q= INT (D(L)/P): IF
    D(L)>Q*P LET Q=Q+1
380:IF Q>18 LET Q=18
390:B$(0)="  I"
400:FOR M=1 TO Q:B$(0)=B
    $(0)+"*": NEXT M
410:LPRINT USING "#####"
    ;H;"I              ";
    D(L)
420:LPRINT B$(0)
430:GOTO 460
440:LPRINT USING "#####"
    ;H;"I"
450:LPRINT "      I"
460:H=H+D: NEXT L
470:LPRINT USING "#####"
    ;B;"+----+----+----+
    ---"
480:LPRINT "": LPRINT ""
500:W=U/(K-1): USING
505:LPRINT "NO. OF DATA=
    ";K-1
510:LPRINT "MEAN VALUE=
    ";W
520:LPRINT "STD. DEV.= "
    ;√(-W*W+V/(K-1))
530:END
```

953

MEMORY CONTENTS

A	Lower bound
B	Upper bound
C	√
D	Interval width
E	Interval number
F	Status flag
G	√
H	√
I	
J	
K	Valid input data number
L	Loop counter
M	Loop counter
N	√
O	
P	Frequency of 1 " * "
Q	√
R	
S	
T	
U	Sum of input data
V	Σx_i^2
W	Mean value
X	Input data (x_i)
Y	
Z	
D (E)	Frequency in each small interval
B$ (Ø) * 21	For graph printout

Program Title: INTERSECTION BETWEEN CIRCLE AND STRAGHT LINE

OVERVIEW

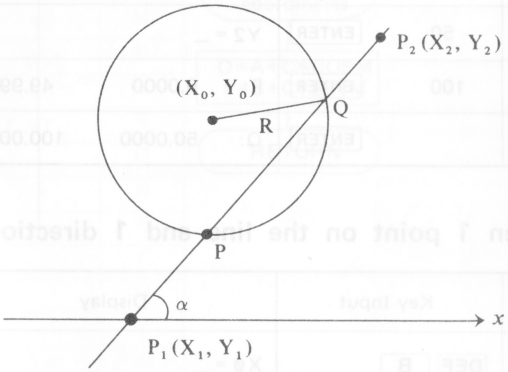
The points of intersection between circles and straight lines in the X–Y plane are found.

CONTENTS

The 2 points of intersection between a circle and a straight line are P and Q.

(Note) The angles are in degrees, minutes, and seconds and are to be input in the following way:

123.1423 = 123 degrees 14 minutes 23 seconds.

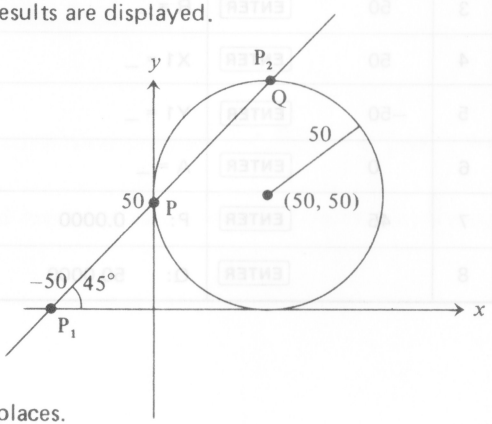


INSTRUCTIONS

1. If the straight line is determined by 2 points, DEF A is used.
   If the line is determined by 1 point and 1 direction angle, DEF B is used.
2. After the data are input, the results are displayed.

EXAMPLE

$X_1 = -50$
$Y_1 = 0$
$X_2 = 50$ $X_P = 0$
$Y_2 = 100$ $Y_P = 50$
$X_0 = 50$ $X_Q = 50$
$Y_0 = 50$ $Y_Q = 100$
$R = 50$
$\alpha = 45°$



(Note) The coordinate values are accurate up to 5 decimal places.

KEY OPERATION SEQUENCE

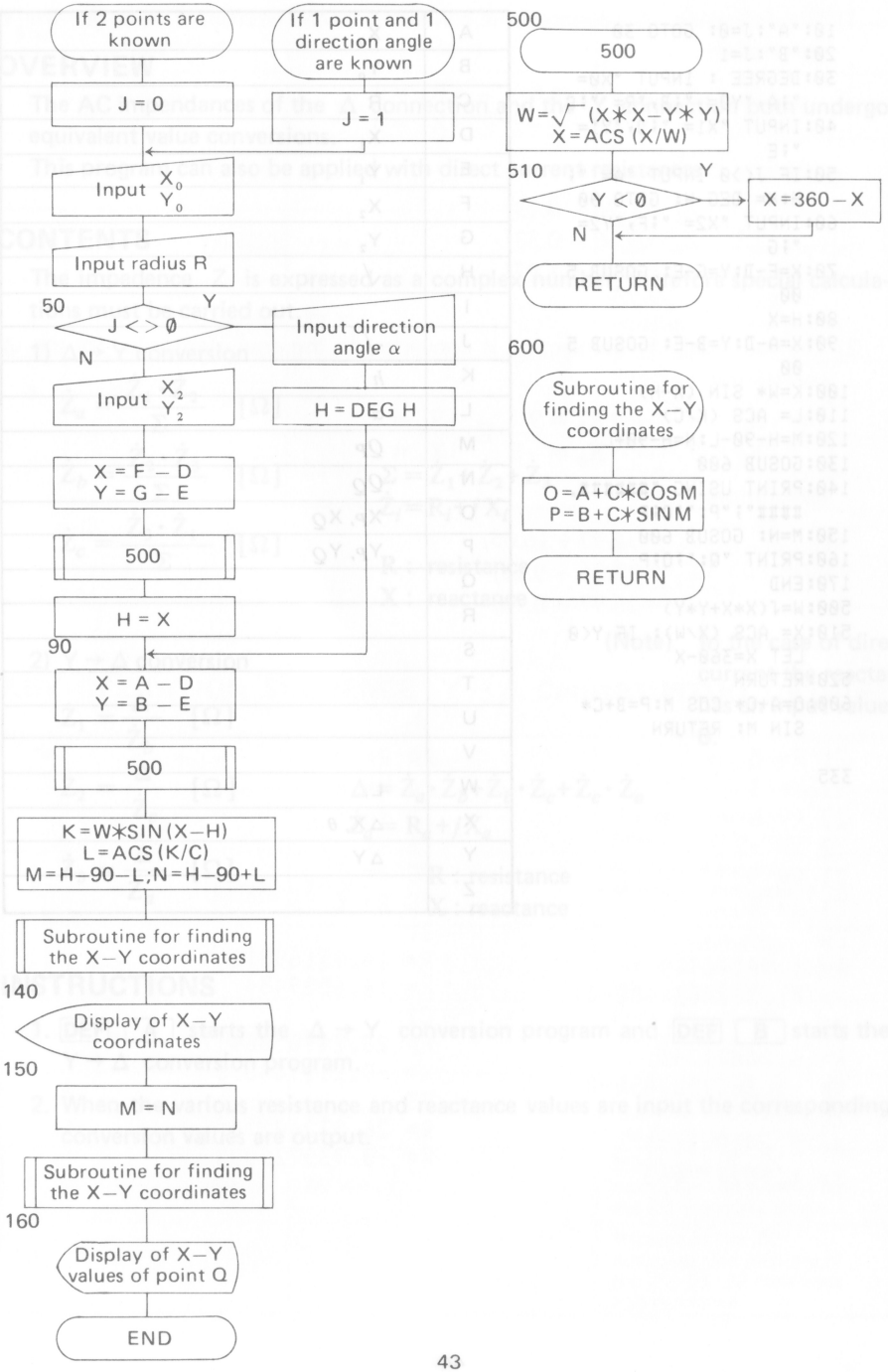
(when 2 points on the line are known)

Step No.	Key Input	Display	Remarks
1	DEF A	X0 = _	
2	50 ENTER	Y0 = _	
3	50 ENTER	R = _	
4	50 ENTER	X1 = _	
5	−50 ENTER	Y1 = _	
6	0 ENTER	X2 = _	
7	50 ENTER	Y2 = _	
8	100 ENTER	P: 0.0000 49.9999	(x_P, y_P)
9	ENTER	Q: 50.0000 100.0000	(x_Q, y_Q)

(when 1 point on the line and 1 direction angle are known)

Step No.	Key Input	Display	Remarks
1	DEF B	X0 = _	
2	50 ENTER	Y0 = _	
3	50 ENTER	R = _	
4	50 ENTER	X1 = _	
5	−50 ENTER	Y1 = _	
6	0 ENTER	A = _	
7	45 ENTER	P: 0.0000 49.9999	(x_p, y_p)
8	ENTER	Q: 50.0000 100.0000	(x_q, y_q)

FLOWCHART


If 2 points are known
If 1 point and 1 direction angle are known
J = 0
J = 1
Input X_0 Y_0
Input radius R
50
J <> Ø
Y
N
Input direction angle α
Input X_2 Y_2
H = DEG H
X = F − D
Y = G − E
500
H = X
90
X = A − D
Y = B − E
500
K=W*SIN(X−H)
L=ACS(K/C)
M=H−90−L;N=H−90+L
Subroutine for finding the X−Y coordinates
140
Display of X−Y coordinates
150
M = N
Subroutine for finding the X−Y coordinates
160
Display of X−Y values of point Q
END
500
500
W=√(X*X−Y*Y)
X=ACS(X/W)
510
Y
Y < Ø
X=360−X
N
RETURN
600
Subroutine for finding the X−Y coordinates
O=A+C*COSM
P=B+C*SINM
RETURN

PROGRAM LIST

```
 10:"A":J=0: GOTO 30
 20:"B":J=1
 30:DEGREE : INPUT "X0=
    ";A,"Y0= ";B,"R= ";C
 40:INPUT "X1= ";D,"Y1=
    ";E
 50:IF J<>0 INPUT "A= ";
    H:H= DEG H: GOTO 90
 60:INPUT "X2= ";F,"Y2=
    ";G
 70:X=F-D:Y=G-E: GOSUB 5
    00
 80:H=X
 90:X=A-D:Y=B-E: GOSUB 5
    00
100:K=W* SIN (X-H)
110:L= ACS (K/C)
120:M=H-90-L:N=H-90+L
130:GOSUB 600
140:PRINT USING "######.
    ####";"P:";O;P
150:M=N: GOSUB 600
160:PRINT "Q:";O;P
170:END
500:W=√(X*X+Y*Y)
510:X= ACS (X/W): IF Y<0
    LET X=360-X
520:RETURN
600:O=A+C* COS M:P=B+C*
    SIN M: RETURN
```

335

MEMORY CONTENTS

A	X_0
B	Y_0
C	R
D	X_1
E	Y_1
F	X_2
G	Y_2
H	$\sqrt{}$
I	
J	$\sqrt{}$
K	h
L	α
M	Q_P
N	Q_Q
O	X_P, X_Q
P	Y_P, Y_Q
Q	
R	
S	
T	
U	
V	
W	L
X	ΔX, θ
Y	ΔY
Z	

Program Title: $\Delta \leftrightarrow$ Y CONVERSIONS

OVERVIEW

The AC impendances of the Δ connection and the Y connection both undergo equivalent value conversions.
This program can also be applied with direct current resistances.

CONTENTS

The impedence Z is expressed as a complex number, therefore special calculations must be carried out.

1) $\Delta \rightarrow$ Y conversion

$$\dot{Z}_a = \frac{\dot{Z}_1 \cdot \dot{Z}_2}{\Sigma} \quad [\Omega]$$

$$\dot{Z}_b = \frac{\dot{Z}_2 \cdot \dot{Z}_3}{\Sigma} \quad [\Omega]$$

$$\dot{Z}_c = \frac{\dot{Z}_3 \cdot \dot{Z}_1}{\Sigma} \quad [\Omega]$$

$$\Sigma = \dot{Z}_1 + \dot{Z}_2 + \dot{Z}_3$$
$$\dot{Z}_i = R_i + jX_i$$

R : resistance
X : reactance

2) Y $\rightarrow \Delta$ conversion

$$\dot{Z}_1 = \frac{\Delta}{\dot{Z}_b} \quad [\Omega]$$

$$\dot{Z}_2 = \frac{\Delta}{\dot{Z}_c} \quad [\Omega]$$

$$\dot{Z}_3 = \frac{\Delta}{\dot{Z}_a} \quad [\Omega]$$

$$\Delta = \dot{Z}_a \cdot \dot{Z}_b + \dot{Z}_b \cdot \dot{Z}_c + \dot{Z}_c \cdot \dot{Z}_a$$
$$\dot{Z}_a = R_a + jX_a$$

R : resistance
X : reactance

(Note) In the case of direct current the reactance has an input value of 0.

INSTRUCTIONS

1. DEF A starts the $\Delta \rightarrow$ Y conversion program and DEF B starts the Y $\rightarrow \Delta$ conversion program.
2. When the various resistance and reactance values are input the corresponding conversion values are output.

EXAMPLE

1) The following Δ connectin is converted to the equivalent value for the Y connection.

$$Z_1\begin{cases}R_1 = 5\\X_1 = 3\end{cases}\quad Z_2\begin{cases}R_2 = 6\\X_2 = -2\end{cases}\quad Z_3\begin{cases}R_3 = 9\\X_3 = 5\end{cases}$$

calculations give

$$\dot{Z}_a = 1.76 - 0.13j$$
$$\dot{Z}_b = 3.10 - 0.33j$$
$$\dot{Z}_c = 2.09 + 1.97j$$

2) The following Y connection is converted to the equivalent value for the Δ connection.

$$Z_a\begin{cases}R_a = 8\\X_a = 3\end{cases}\quad Z_b\begin{cases}R_b = 9\\X_b = -5\end{cases}\quad Z_c\begin{cases}R_c = 7\\X_c = 6\end{cases}$$

calculations give

$$\dot{Z}_1 = 14.97 + 16.65j$$
$$\dot{Z}_2 = 23.25 - 9.21j$$
$$\dot{Z}_3 = 26.97 - 0.74j$$

PRINTED OUTPUTS

```
Z1 R= 5
Z1 X= 3
Z2 R= 6
Z2 X= -2
Z3 R= 9
Z3 X= 5
ZA
R= 1.76146789
X= -1.284403668E-01
ZB
R= 3.10091743
X= -3.302752299E-01
ZC
R= 2.091743119
X= 1.972477063
```

```
ZA R= 8
ZA X= 3
ZB R= 9
ZB X= -5
ZC R= 7
ZC X= 6
Z1
R= 14.97169811
X= 16.6509434
Z2
R= 23.24705884
X= -9.211764695
Z3
R= 26.97260274
X= -0.739726009
```

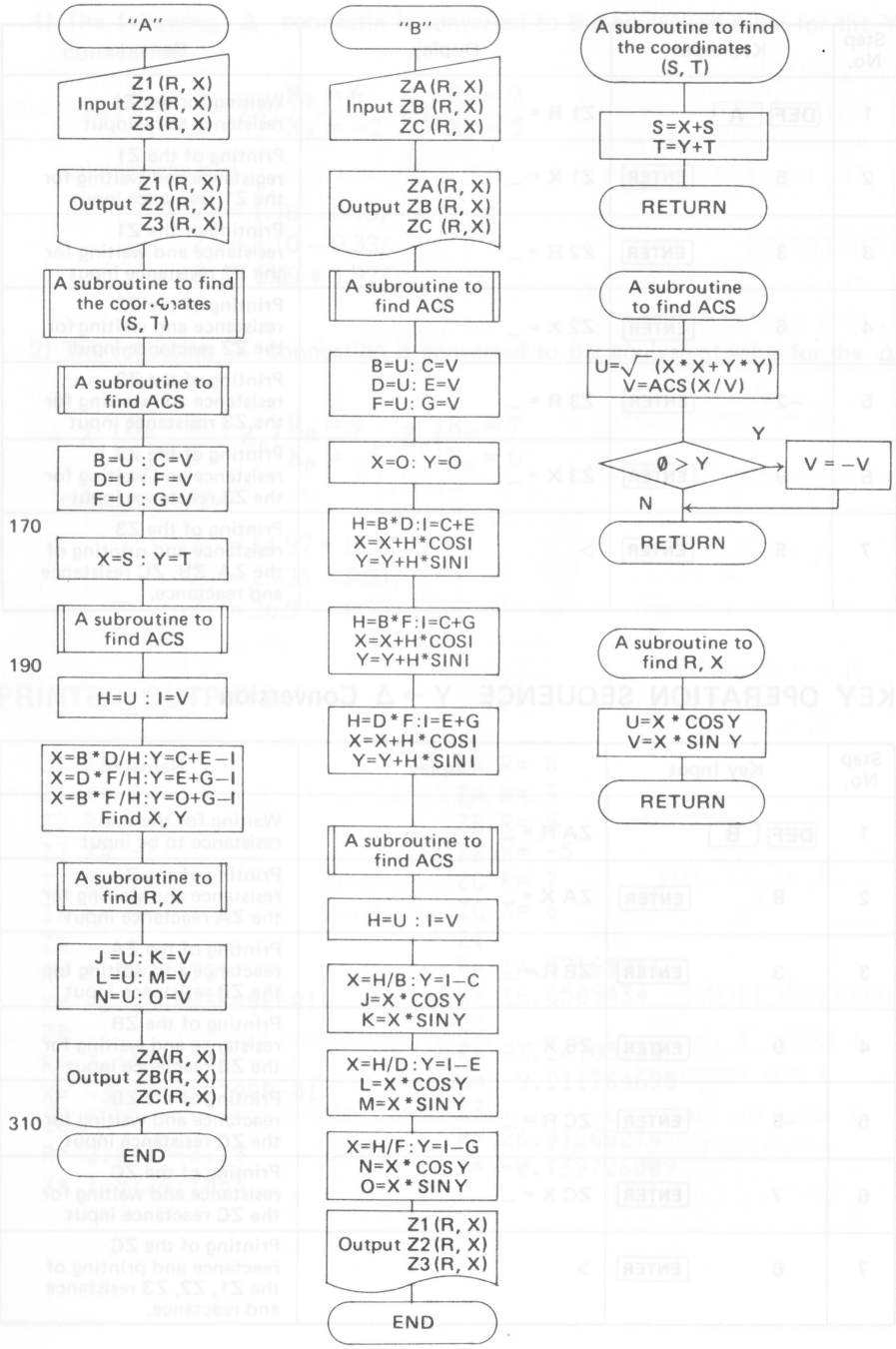
KEY OPERATION SEQUENCE Δ → Y Conversion

Step No.	Key Input	Display	Remarks
1	DEF A	Z1 R = _	Waiting for the Z1 resistance to be input
2	5 ENTER	Z1 X = _	Printing of the Z1 registance and waiting for the Z1 reactance input
3	3 ENTER	Z2 R = _	Printing of the Z1 resistance and waiting for the Z2 resistance input
4	6 ENTER	Z2 X = _	Printing of the Z2 resistance and waiting for the Z2 reactance input
5	−2 ENTER	Z3 R = _	Printing of the Z2 resistance and waiting for the Z3 resistance input
6	9 ENTER	Z3 X = _	Printing of the Z3 resistance and waiting for the Z3 reactance input
7	5 ENTER	>	Printing of the Z3 resistance and printing of the ZA, ZB, ZC resistance and reactance.

KEY OPERATION SEQUENCE Y → Δ Conversion

Step No.	Key Input	Display	Remarks
1	DEF B	ZA R = _	Waiting for the ZA resistance to be input
2	8 ENTER	ZA X = _	Printing of the ZA resistance and waiting for the ZA reactance input
3	3 ENTER	ZB R = _	Printing of the ZA reactance and waiting for the ZB resistance input
4	9 ENTER	ZB X = _	Printing of the ZB resistance and waiting for the ZB reactance input
5	−5 ENTER	ZC R = _	Printing of the ZB reactance and waiting for the ZC resistance input
6	7 ENTER	ZC X = _	Printing of the ZC resistance and waiting for the ZC reactance input
7	6 ENTER	>	Printing of the ZC reactance and printing of the Z1, Z2, Z3 resistance and reactance.

FLOWCHART


"A"
Z1 (R, X)
Input Z2 (R, X)
Z3 (R, X)
Z1 (R, X)
Output Z2 (R, X)
Z3 (R, X)
A subroutine to find the coordinates (S, T)
A subroutine to find ACS
B=U : C=V
D=U : F=V
F=U : G=V
170
X=S : Y=T
A subroutine to find ACS
190
H=U : I=V
X=B * D/H:Y=C+E−I
X=D * F/H:Y=E+G−I
X=B * F/H:Y=O+G−I
Find X, Y
A subroutine to find R, X
J=U: K=V
L=U: M=V
N=U: O=V
ZA(R, X)
Output ZB(R, X)
ZC(R, X)
310
END
"B"
ZA(R, X)
Input ZB (R, X)
ZC (R, X)
ZA(R, X)
Output ZB (R, X)
ZC (R, X)
A subroutine to find ACS
B=U: C=V
D=U: E=V
F=U: G=V
X=O: Y=O
H=B * D:I=C+E
X=X+H * COSI
Y=Y+H * SINI
H=B * F:I=C+G
X=X+H * COSI
Y=Y+H * SINI
H=D * F:I=E+G
X=X+H * COSI
Y=Y+H * SINI
A subroutine to find ACS
H=U : I=V
X=H/B : Y=I−C
J=X * COSY
K=X * SINY
X=H/D : Y=I−E
L=X * COSY
M=X * SINY
X=H/F : Y=I−G
N=X * COSY
O=X * SINY
Z1 (R, X)
Output Z2 (R, X)
Z3 (R, X)
END
A subroutine to find the coordinates (S, T)
S=X+S
T=Y+T
RETURN
A subroutine to find ACS
U=√(X * X+Y * Y)
V=ACS (X/V)
Y
Ø > Y
V = −V
N
RETURN
A subroutine to find R, X
U=X * COSY
V=X * SIN Y
RETURN

PROGRAM LIST

```
 10:"A":T=0:S=0: DEGREE
 20:INPUT "Z1 R=";X:
    LPRINT "Z1 R= ";
    STR$ X
 30:INPUT "Z1 X=";Y:
    LPRINT "Z1 X= ";
    STR$ Y
 40:GOSUB 400
 50:GOSUB 350
 60:B=U:C=V
 70:INPUT "Z2 R=";X:
    LPRINT "Z2 R= ";
    STR$ X
 80:INPUT "Z2 X=";Y:
    LPRINT "Z2 X= ";
    STR$ Y
 90:GOSUB 400
100:GOSUB 350
110:D=U:E=V
120:INPUT "Z3 R=";X:
    LPRINT "Z3 R= ";
    STR$ X
130:INPUT "Z3 X=";Y:
    LPRINT "Z3 X= ";
    STR$ Y
140:GOSUB 400
150:GOSUB 350
160:F=U:G=V
170:X=S:Y=T
180:GOSUB 350
190:H=U:I=V
200:X=B*D/H:Y=C+E-I:
    GOSUB 450
210:J=U:K=V
220:X=D*F/H:Y=E+G-I
230:GOSUB 450
240:L=U:M=V
250:X=B*F/H:Y=C+G-I
260:GOSUB 450
270:N=U:O=V
280:USING : LPRINT "ZA":
    LPRINT "R= "; STR$ J
    : LPRINT "X= "; STR$
    K
290:LPRINT "ZB": LPRINT
    "R= "; STR$ L:
    LPRINT "X= "; STR$ M
300:LPRINT "ZC": LPRINT
    "R= "; STR$ N:
    LPRINT "X= "; STR$ O
310:END
350:U=√(X*X+Y*Y)
360:V= ACS (X/U)
370:IF 0>Y LET V=-V
380:RETURN
400:S=X+S:T=Y+T
410:RETURN
450:U=X* COS Y:V=X* SIN
    Y
460:RETURN
505:"B": CLEAR
510:DEGREE : INPUT "ZA R
    =";X: LPRINT "ZA R=
    "; STR$ X
520:INPUT "ZA X=";Y:
    LPRINT "ZA X= ";
    STR$ Y
530:GOSUB 350
540:B=U:C=V
550:INPUT "ZB R=";X:
    LPRINT "ZB R= ";
    STR$ X
560:INPUT "ZB X=";Y:
    LPRINT "ZB X= ";
    STR$ Y
570:GOSUB 350
580:D=U:E=V
590:INPUT "ZC R=";X:
    LPRINT "ZC R= ";
    STR$ X
600:INPUT "ZC X=";Y:
    LPRINT "ZC X= ";
    STR$ Y
610:GOSUB 350
620:F=U:G=V
630:X=0:Y=0
640:H=B*D:I=C+E
650:X=X+H* COS I
```

```
660:Y=Y+H* SIN I
670:H=B*F:I=C+G
680:X=X+H* COS I
690:Y=Y+H* SIN I
710:H=D*F
720:I=E+G
730:X=X+H* COS I
740:Y=Y+H* SIN I
750:GOSUB 350
760:H=U:I=V
770:X=H/B:Y=I-C
780:J=X* COS Y:K=X* SIN
    Y
790:X=H/D:Y=I-E
800:L=X* COS Y:M=X* SIN
    Y
810:X=H/F:Y=I-G
820:N=X* COS Y:O=X* SIN
    Y
860:LPRINT "Z1": LPRINT
    "R= "; STR$ L:
    LPRINT "X= "; STR$ M
870:LPRINT "Z2": LPRINT
    "R= "; STR$ N:
    LPRINT "X= "; STR$ O
880:LPRINT "Z3": LPRINT
    "R= "; STR$ J:
    LPRINT "X= "; STR$ K
890:END

1076
```

MEMORY CONTENTS

	Δ → Y Conversion	Y → Δ Conversion
A		
B	$R_1 (Z_1)$	$R_a (Z_a)$
C	$X_1 (Z_1)$	$X_a (Z_a)$
D	$R_2 (Z_2)$	$R_b (Z_b)$
E	$X_2 (Z_2)$	$X_b (Z_b)$
F	$R_3 (Z_3)$	$R_c (Z_c)$
G	$X_3 (Z_3)$	$X_c (Z_c)$
H	Σ	Δ
I	Σ	Δ
J	$R_a (Z_a)$	$R_3 (Z_3)$
K	$X_a (Z_a)$	$X_3 (Z_3)$
L	$R_b (Z_b)$	$R_1 (Z_1)$
M	$X_b (Z_b)$	$X_1 (Z_1)$
N	$R_c (Z_c)$	$R_2 (Z_2)$
O	$X_c (Z_c)$	$X_2 (Z_2)$
P		
Q		
R		
S	√	
T	√	
U	√	|Z|
V	√	θ
W		
X	√	√
Y	√	√
Z		

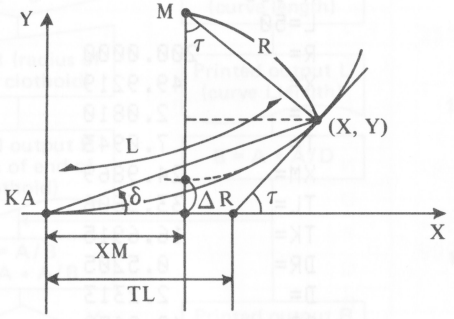
Program Title: CLOTHOID CURVE

OVERVIEW

This program will assist in finding the various elements of clothoid curves used, for instance, in the designing of roads.

CONTENTS

For clothoid curves, A, R or A, L are input to find X, Y.



$$L = \frac{A^2}{R}$$

$$l = \frac{A}{R}$$

$$X \fallingdotseq Al\left(1 - \frac{l^4}{40} + \frac{l^8}{3456} - \frac{l^{12}}{599040}\right)$$

$$Y \fallingdotseq \frac{Al^3}{6}\left(1 - \frac{l^4}{56} + \frac{l^8}{7040} - \frac{l^{12}}{1612800}\right)$$

Tangent angle $\tau = \frac{L}{2R} \times \frac{180}{\pi}$

$XM = X - R \sin \tau$ $\quad \delta = \tan^{-1}\frac{Y}{X}$

$TL = X - Y \cot \tau$

$TK = Y \cdot \operatorname{cosec} \tau$ $\quad SC = Y \cdot \operatorname{cosec} \delta$

Amount of movement

$\Delta R = Y + R \cos \tau - R$

INSTRUCTIONS

1. DEF A : when the parameters and the radius of the clothoid curve are known, DEF B : when the parameters of the clothoid curve and the curve length are known.
2. After inputting the data, the elements of the clothoid curve are output on the printer.

EXAMPLE

A = 100	X = 49.9219	XM = 24.9869 m
R = 200 m	Y = 2.081	TL = 33.3606 m
L = 50	t = 7°09'43"	ΔR = 0.5205 m
	SC = 49.9652	δ = 2°23'13"
		TK = 16.6915

(Note) The distance is rounded off at the 5th digit.

PRINTED OUTPUTS

```
A=100
R=200
L=        50.0000
X=        49.9219
Y=         2.0810
T=         7.0943
XM=       24.9869
TL=       33.3606
TK=       16.6915
DR=        0.5205
D=         2.2313
SC=       49.9652
```

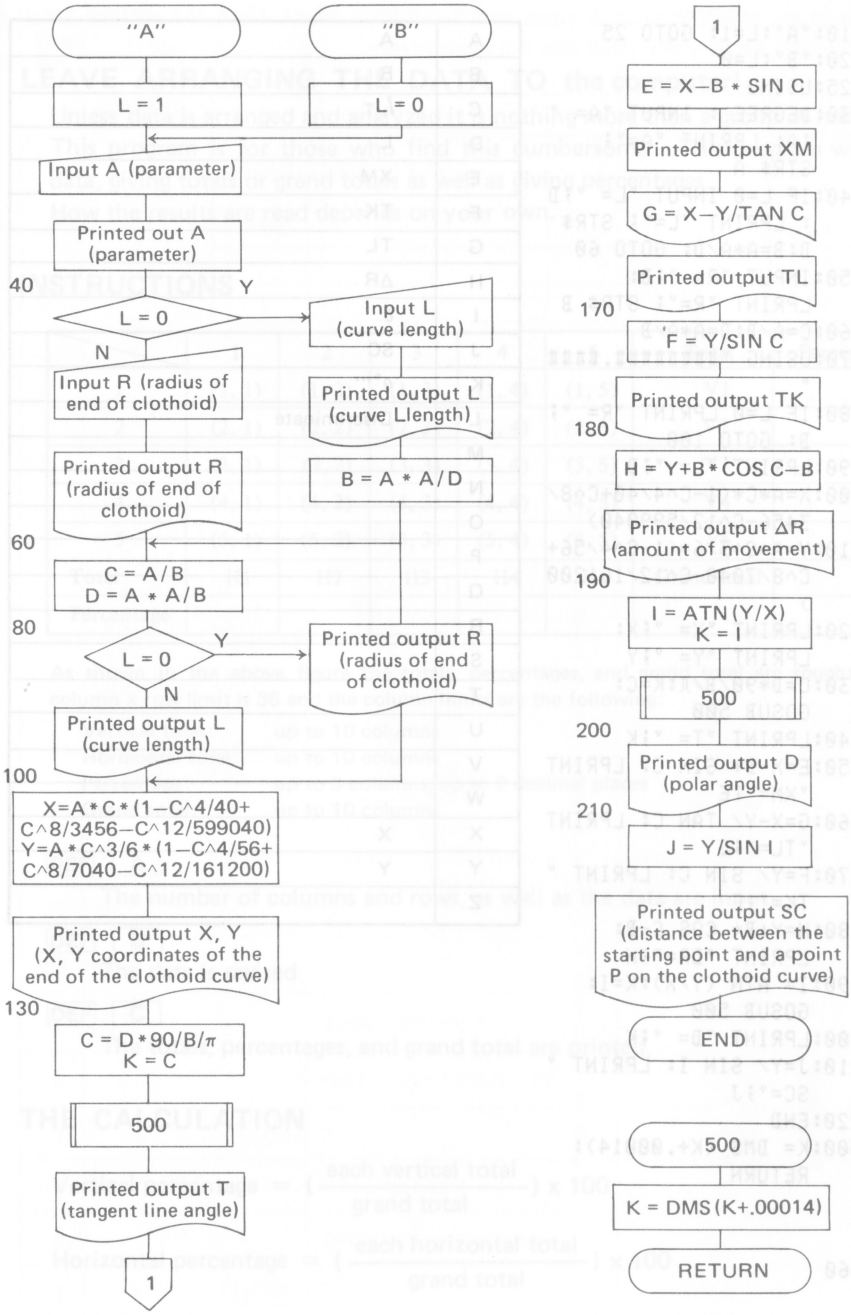
```
A=100
L=50
R=       200.0000
X=        49.9219
Y=         2.0810
T=         7.0943
XM=       24.9869
TL=       33.3606
TK=       16.6915
DR=        0.5205
D=         2.2313
SC=       49.9652
```

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	A = _	Waiting for parameters to be input
2	100 ENTER	R = _	Printing the parameters and waiting for the input of the radius of the end of the clothoid curve.
3	200 ENTER	>	Printing the radius and printing the clothoid elements.

Step No.	Key Input	Display	Remarks
1	DEF B	A = _	Waiting for the parameters to be input
2	100 ENTER	L = _	Printing the parameters and waiting for the curve length to be input
3	50 ENTER	>	Printing the curve length and printing of the clothoid elements

FLOWCHART


"A"
"B"
L = 1
L = 0
Input A (parameter)
Printed out A (parameter)
40
L = 0
Y
Input L (curve length)
N
Input R (radius of end of clothoid)
Printed output L (curve Llength)
Printed output R (radius of end of clothoid)
B = A * A/D
60
C = A/B
D = A * A/B
80
L = 0
Y
Printed output R (radius of end of clothoid)
N
Printed output L (curve length)
100
X=A*C*(1−C^4/40+C^8/3456−C^12/599040)
Y=A*C^3/6*(1−C^4/56+C^8/7040−C^12/161200)
Printed output X, Y (X, Y coordinates of the end of the clothoid curve)
130
C = D*90/B/π
K = C
500
Printed output T (tangent line angle)
1
1
E = X−B * SIN C
Printed output XM
G = X−Y/TAN C
Printed output TL
170
F = Y/SIN C
Printed output TK
180
H = Y+B*COS C−B
Printed output ΔR (amount of movement)
190
I = ATN(Y/X)
K = I
500
200
Printed output D (polar angle)
210
J = Y/SIN I
Printed output SC (distance between the starting point and a point P on the clothoid curve)
END
500
K = DMS(K+.00014)
RETURN

PROGRAM LIST

```
10:"A":L=1: GOTO 25
20:"B":L=0
25:USING
30:DEGREE : INPUT "A= "
   ;A: LPRINT "A=";
   STR$ A
40:IF L=0 INPUT "L= ";D
   : LPRINT "L="; STR$
   D:B=A*A/D: GOTO 60
50:INPUT "R= ";B:
   LPRINT "R="; STR$ B
60:C=A/B:D=A*A/B
70:USING "#########.#####
   "
80:IF L=0 LPRINT "R= ";
   B: GOTO 100
90:LPRINT "L= ";D
100:X=A*C*(1-C^4/40+C^8/
   3456-C^12/599040)
110:Y=A*C^3/6*(1-C^4/56+
   C^8/7040-C^12/161200
   )
120:LPRINT "X= ";X:
   LPRINT "Y= ";Y
130:C=D*90/B/π:K=C:
   GOSUB 500
140:LPRINT "T= ";K
150:E=X-B* SIN C: LPRINT
   "XM=";E
160:G=X-Y/ TAN C: LPRINT
   "TL=";G
170:F=Y/ SIN C: LPRINT "
   TK=";F
180:H=Y+B* COS C-B:
   LPRINT "DR=";H
190:I= ATN (Y/X):K=I:
   GOSUB 500
200:LPRINT "D= ";K
210:J=Y/ SIN I: LPRINT "
   SC=";J
220:END
500:K= DMS (K+.00014):
   RETURN


460
```

MEMORY CONTENTS

A	A
B	R
C	l, τ
D	L
E	XM
F	TK
G	TL
H	ΔR
I	δ
J	SC
K	$\theta^{\circ\prime\prime\prime}$
L	Discriminate
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	X
Y	Y
Z	

Program Title: CROSS-FOOTING

LEAVE ARRANGING THE DATA TO the computer!

Unless data is arranged and analyzed it is nothing more than a group of numbers. This program is for those who find this cumbersome. This program will add data, giving totals or grand totals as well as giving percentages.
How the results are read depends on your own.

INSTRUCTIONS

	1	2	3	4	5	Total	Percentage
1	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	V1	
2	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	V2	
3	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	V3	
4	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	V4	
5	(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	V5	
Total	H1	H2	H3	H4	H5	Grand total	
Percentage							

As shown in the above figure the totals, percentages, and grand total are sought. The column x row limit is 36 and the column limits are the following:

Vertical total:	up to 10 columns
Horizontal total:	up to 10 columns
Percentage:	up to 3 columns, up to 2 decimal places
Grand total:	up to 10 columns

DEF A

The number of columns and rows, as well as the data are input.

DEF B

The data is revised

DEF C

The totals, percentages, and grand total are printed.

THE CALCULATION

$$\text{Vertical percentage} = \left(\frac{\text{each vertical total}}{\text{grand total}}\right) \times 100$$

$$\text{Horizontal percentage} = \left(\frac{\text{each horizontal total}}{\text{grand total}}\right) \times 100$$

percentage is rounded off at the third decimal place.

EXAMPLE

This is an example with 5 rows and 5 columns which finds the vertical total, horizontal total, grand total and the percentages.

	1	2	3	4	5
1	7	3	1	2	1
2	4	9	2	4	3
3	5	8	3	6	5
4	6	6	4	8	7
5	3	7	5	10	9

PRINTED OUTPUTS

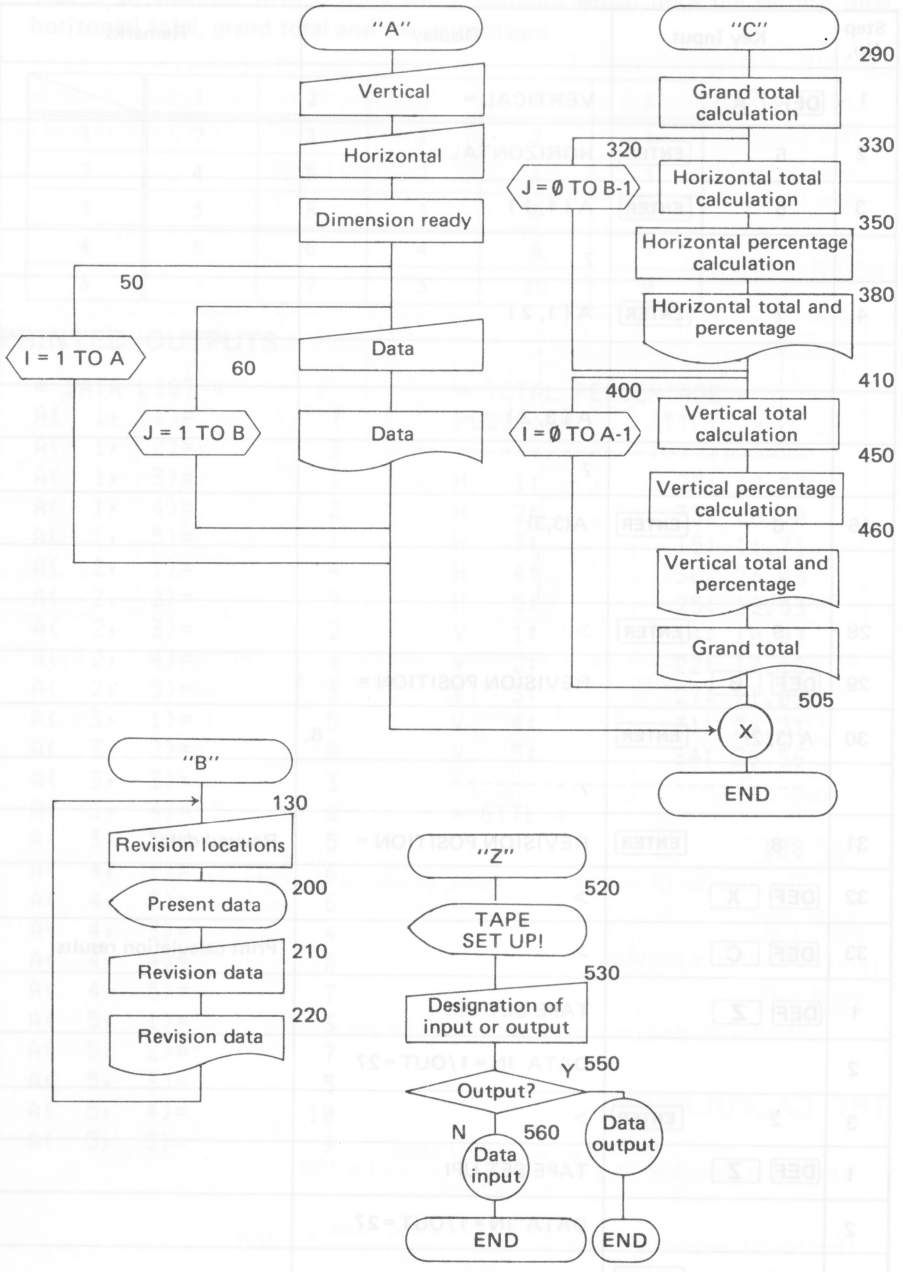
```
* DATA LIST *
A( 1, 1)=          7
A( 1, 2)=          3
A( 1, 3)=          1
A( 1, 4)=          2
A( 1, 5)=          1
A( 2, 1)=          4
A( 2, 2)=          9
A( 2, 3)=          2
A( 2, 4)=          4
A( 2, 5)=          3
A( 3, 1)=          5
A( 3, 2)=          8
A( 3, 3)=          3
A( 3, 4)=          6
A( 3, 5)=          5
A( 4, 1)=          6
A( 4, 2)=          6
A( 4, 3)=          4
A( 4, 4)=          8
A( 4, 5)=          7
A( 5, 1)=          3
A( 5, 2)=          7
A( 5, 3)=          5
A( 5, 4)=         10
A( 5, 5)=          9
```

```
* TOTAL PERCENTAGE *
POSITION          TTL:   %
-----+----------+------
H   1:             25: 19.53
H   2:             33: 25.78
H   3:             15: 11.71
H   4:             30: 23.43
H   5:             25: 19.53
V   1:             14: 10.93
V   2:             22: 17.18
V   3:             27: 21.09
V   4:             31: 24.21
V   5:             34: 26.56
-----------------------
* GTTL *
                    128
```

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	VERTICAL =	
2	5 ENTER	HORIZONTAL =	
3	5 ENTER	A (1, 1)	
		?	
4	7 ENTER	A (1, 2)	
	⋮	⋮	
		A (3, 2)	
		?	
15	6 ENTER	A(3,3)	
	⋮	⋮	
28	9 ENTER	>	
29	DEF B	REVISION POSITION =	
30	A (3, 2) ENTER	6.	
		?	
31	8 ENTER	REVISION POSITION =	Revised data
32	DEF X	>	
33	DEF C	>	Print calculation results
1	DEF Z	TAPE SET UP!	
2		DATA IN = 1 / OUT = 2?	
3	2 ENTER	>	
1	DEF Z	TAPE SET UP!	
2		DATA IN = 1 / OUT = 2?	
3	1 ENTER	>	

FLOWCHART


"A"
Vertical
Horizontal
Dimension ready
50
I = 1 TO A
60
J = 1 TO B
Data
Data
"C"
290
Grand total calculation
320
J = Ø TO B-1
330
Horizontal total calculation
350
Horizontal percentage calculation
380
Horizontal total and percentage
400
I = Ø TO A-1
410
Vertical total calculation
450
Vertical percentage calculation
460
Vertical total and percentage
Grand total
505
X
END
"B"
130
Revision locations
200
Present data
210
Revision data
220
Revision data
"Z"
520
TAPE SET UP!
530
Designation of input or output
Y 550
Output?
N
560
Data input
Data output
END
END

PROGRAM LIST

```
 10:"A": CLEAR : WAIT :
    USING : LPRINT "* DA
    TA LIST *"
 20:INPUT "VERTICAL=";A
 30:INPUT "HORIZONTAL=";
    B
 40:DIM B(A-1,B-1)
 50:FOR I=1 TO A
 60:FOR J=1 TO B
 70:PAUSE "A("; USING "#
    ##";I;",";J;")"
 80:INPUT B(I-1,J-1)
 90:LPRINT "A("; USING "
    ###";I;",";J;")=";
    USING "##########";B
    (I-1,J-1)
100:NEXT J: NEXT I
110:GOTO 505
120:"B": WAIT : USING :
    LPRINT "* REVISED DA
    TA *"
130:INPUT "REVISION POSI
    TION=";C$:Z= LEN C$:
    X$= RIGHT$ (C$,Z-2):
    Y= VAL X$: GOTO 145
140:GOTO 130
145:IF Y<1 GOTO 130
150:W=Z- LEN STR$ Y-3
160:IF W<0 GOTO 130
170:IF MID$ (C$,Z-W,1)<>
    "," LET W=W-1: GOTO
    160
180:V$= RIGHT$ (C$,W):U=
    VAL V$: IF U<1 GOTO
    130
200:PAUSE B(Y-1,U-1)
210:INPUT B(Y-1,U-1)
220:LPRINT "A("; USING "
    ###";Y;",";U;")=";
    USING "##########";B
    (Y-1,U-1)
230:GOTO 130
240:"C": WAIT : USING :
    LPRINT "* TOTAL PERC
    ENTAGE *"
250:LPRINT "POSITION
      TTL:  %"
260:LPRINT "-----+------
    -----+------"
270:D=0
280:FOR I=0 TO A-1
290:FOR J=0 TO B-1
300:D=D+B(I,J)
310:NEXT J: NEXT I
320:FOR J=0 TO B-1
330:E=0
340:FOR I=0 TO A-1
350:E=E+B(I,J)
360:NEXT I
370:F= INT (E/D*E5+0.5)/
    E3
380:LPRINT "H"; USING "#
    ###";J+1;":"; USING
    "##########";E;":";
    USING "###.##";F
390:NEXT J
400:FOR I=0 TO A-1
410:G=0
420:FOR J=0 TO B-1
430:G=G+B(I,J)
440:NEXT J
450:H= INT (G/D*E5+0.5)/
    E3
460:LPRINT "V"; USING "#
    ###";I+1;":"; USING
    "##########";G;":";
    USING "###.##";H
470:NEXT I
480:LPRINT "------------
    ------------"
490:LPRINT "* GTTL *"
500:LPRINT USING "######
    #####";D
505:"X": LPRINT "":
    LPRINT "": LPRINT ""
510:END
520:"Z": PAUSE "TAPE SET
     UP!"
530:INPUT "DATA IN=1/OUT
    =2?";K: IF (K=1)+(K=
    2)=1 GOTO 550
540:BEEP 3: GOTO 530
550:IF K=2 PRINT #"CF";A
    ,B: PRINT #"CF-DATA"
    ;B(*): END
560:CLEAR : INPUT #"CF";
    A,B: DIM B(A-1,B-1):
    INPUT #"CF-DATA";B(*
    )
570:END


1121
```

MEMORY CONTENTS

A	Number of columns
B	Number of rows
C	Revision location
D	Grand total
E	Horizontal total
F	Horizontal percentage
G	Vertical total
H	Vertical percentage
I	√
J	√
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	Horizontal revision location
V	√
W	√
X	√
Y	Vertical revision location
Z	√
B$(N−1)	√
C$(Ø)	√
B(N−1)	√

Program Title: NUMBER OF DAYS CALCULATION

OVERVIEW

How many days has it been since you were born?
This program is convenient for answering such questions. By setting a certain day, this program will output the number of days that have passed since that day.

INSTRUCTIONS

[DEF] [A]

BASE YEAR [ENTER]
MONTH [ENTER]
DAY [ENTER]
TARGET YEAR [ENTER]
MONTH [ENTER]
DAY [ENTER]

To end the program, type in [DEF] [Z] in place of the year.

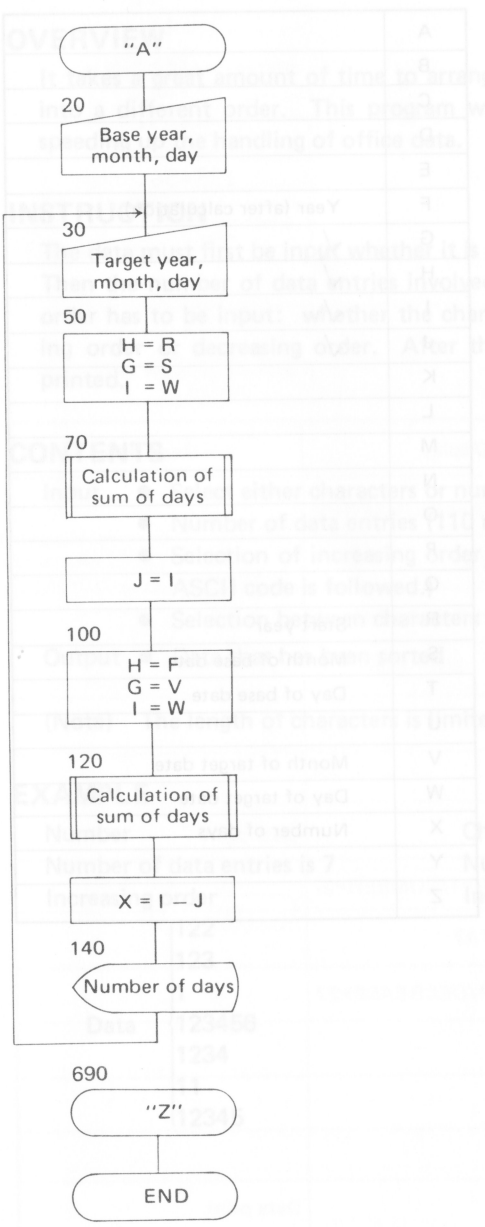
EXAMPLE

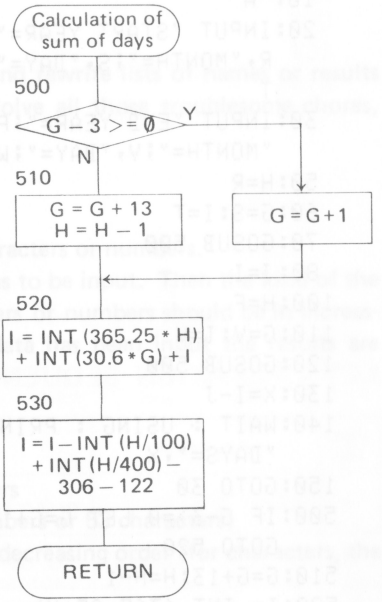
from 1976 year 10 month 5 day
to 1982 year 6 month 4 day : 2068 days
to 1985 year 1 month 1 day : 3010 days

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	START YEAR =	
2	1976 ENTER	MONTH =	Base date 1976 year 10 month 5 day input
3	10 ENTER	DAY =	
4	5 ENTER	END YEAR =	
5	1982 ENTER	MONTH =	Target date 1982 year 6 month 4 day input
6	6 ENTER	DAY =	
7	4 ENTER	DAYS = 2068.	
8	ENTER	END YEAR =	
9	1985 ENTER	MONTH =	Target date 1985 year 1 month 1 day input
10	1 ENTER	DAY =	
11	1 ENTER	DAYS = 3010.	
12	ENTER	END YEAR =	
13	DEF Z	>	

FLOWCHART


"A"
20
Base year, month, day
30
Target year, month, day
50
H = R
G = S
I = W
70
Calculation of sum of days
J = I
100
H = F
G = V
I = W
120
Calculation of sum of days
X = I – J
140
Number of days
690
"Z"
END


Calculation of sum of days
500
G – 3 >= 0
Y
N
510
G = G + 13
H = H – 1
G = G+1
520
I = INT (365.25 * H) + INT (30.6 * G) + I
530
I = I – INT (H/100) + INT (H/400) – 306 – 122
RETURN

PROGRAM LIST

```
 10:"A"
 20:INPUT "START YEAR=";
    R,"MONTH=";S,"DAY=";
    T
 30:INPUT "END YEAR=";F,
    "MONTH=";V,"DAY=";W
 50:H=R
 60:G=S:I=T
 70:GOSUB 500
 80:J=I
100:H=F
110:G=V:I=W
120:GOSUB 500
130:X=I-J
140:WAIT : USING : PRINT
    "DAYS=",X
150:GOTO 30
500:IF G-3>=0 LET G=G+1:
    GOTO 520
510:G=G+13:H=H-1
520:I= INT (365.25*H)+
    INT (30.6*G)+I
530:I=I- INT (H/100)+
    INT (H/400)-306-122:
    RETURN
600:"Z": END

270
```

MEMORY CONTENTS

A	
B	
C	
D	
E	
F	Year (after calculation)
G	√
H	√
I	√
J	√
K	
L	
M	
N	
O	
P	
Q	
R	Start year
S	Month of base date
T	Day of base date
U	
V	Month of target date
W	Day of target date
X	Number of days
Y	
Z	

Program Title: SORTING

OVERVIEW

It takes a great amount of time to arrange and rewrite lists of names or results into a different order. This program will solve all those troublesome chores, speeding up the handling of office data.

INSTRUCTION

The data must first be input whether it is characters or numbers.
Then the number of data entries involved has to be input. Then the kind of the order has to be input: whether the characters or numbers should be in increasing order or decreasing order. After the data has been input the results are printed.

CONTENTS

Input
- Select either characters or numbers
- Number of data entries (110 numbers or 53 characters)
- Selection of increasing order or decreasing order (for characters, the ASCII code is followed.)
- Selection between characters or numbers

Output
- Data that has been sorted

(Note) The length of characters is limited to 16.

EXAMPLE

Number
Number of data entries is 7
Increasing order

Data
122
123
1
123456
1234
11
12345

Character
Number of data entries is 5
Increasing order

Data
CCR
ABCDE
RGHI
KTV
CCK

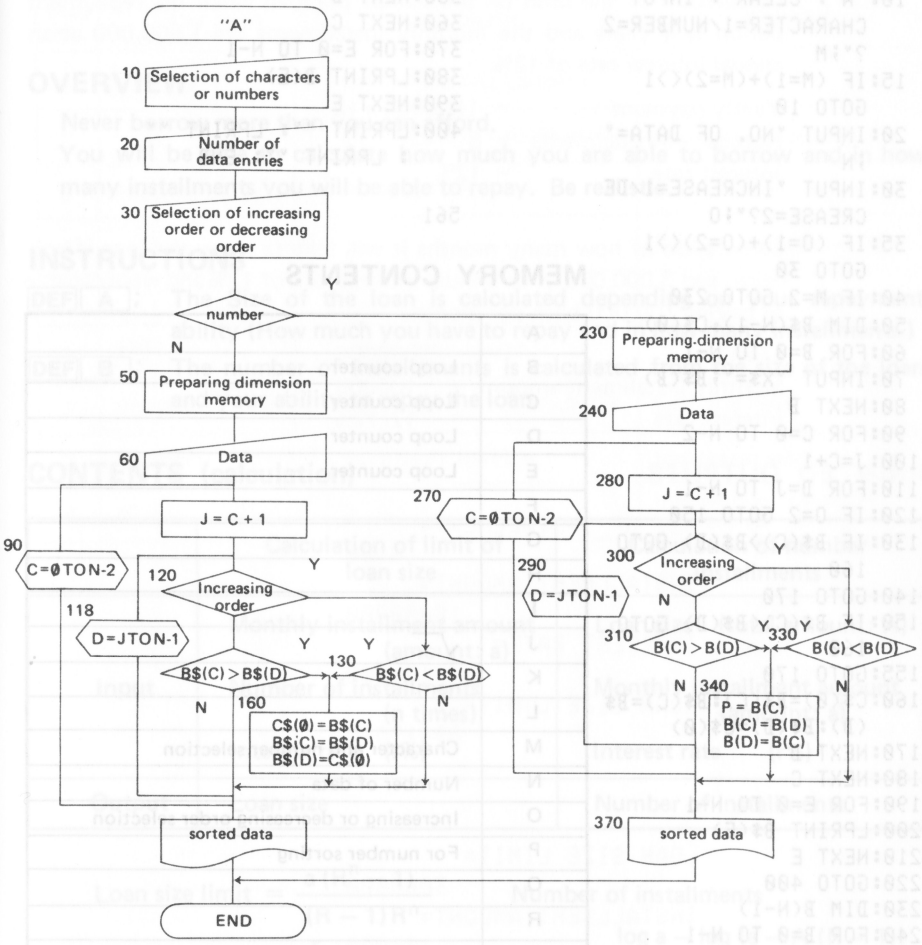
PRINTED OUTPUTS

```
ABCDE                    1.
CCK                     11.
CCR                    122.
KTV                    123.
RGHI                  1234.
                     12345.
                    123456.
```

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	CHARACTER=1/NUMBER=2?	
2	2 ENTER	NO. OF DATA	
3	7 ENTER	INCREASE=1/DECREASE=2?	
4	1 ENTER	X =	
5	122 ENTER	X =	
	⋮	⋮	
12	12345 ENTER	>	Data print
1	DEF A	CHARACTER=1/NUMBER=2?	
2	1 ENTER	NO. OF DATA?	
3	5 ENTER	INCREASE=1/DECREASE=2?	
4	1 ENTER	X$ =	
5	CCR ENTER	X$ =	
	⋮	⋮	
9	CCK ENTER	>	Data print

FLOWCHART


"A"
10 Selection of characters or numbers
20 Number of data entries
30 Selection of increasing order or decreasing order
number
Y
N
50 Preparing dimension memory
60 Data
J = C + 1
90 C=ØTON-2
120 Increasing order
Y
118 D=JTON-1
B$(C) > B$(D)
B$(C) < B$(D)
130
160 C$(Ø) = B$(C)
B$(C) = B$(D)
B$(D) = C$(Ø)
sorted data
END
230 Preparing dimension memory
240 Data
280 J = C + 1
270 C=ØTON-2
300 Increasing order
290 D=JTON-1
310 B(C) > B(D)
330
B(C) < B(D)
340 P = B(C)
B(C) = B(D)
B(D) = B(C)
370 sorted data

PROGRAM LIST

```
 10:"A": CLEAR : INPUT "
    CHARACTER=1/NUMBER=2
    ?";M
 15:IF (M=1)+(M=2)<>1
    GOTO 10
 20:INPUT "NO. OF DATA="
    ;N
 30:INPUT "INCREASE=1/DE
    CREASE=2?";O
 35:IF (O=1)+(O=2)<>1
    GOTO 30
 40:IF M=2 GOTO 230
 50:DIM B$(N-1),C$(0)
 60:FOR B=0 TO N-1
 70:INPUT "X$=";B$(B)
 80:NEXT B
 90:FOR C=0 TO N-2
100:J=C+1
110:FOR D=J TO N-1
120:IF O=2 GOTO 150
130:IF B$(C)>B$(D) GOTO
    160
140:GOTO 170
150:IF B$(C)<B$(D) GOTO
    160
155:GOTO 170
160:C$(0)=B$(C):B$(C)=B$
    (D):B$(D)=C$(0)
170:NEXT D
180:NEXT C
190:FOR E=0 TO N-1
200:LPRINT B$(E)
210:NEXT E
220:GOTO 400
230:DIM B(N-1)
240:FOR B=0 TO N-1
250:INPUT "X=";B(B)
260:NEXT B
270:FOR C=0 TO N-2
280:J=C+1
290:FOR D=J TO N-1
300:IF O=2 GOTO 330
310:IF B(C)>B(D) GOTO 34
    0
320:GOTO 350
330:IF B(C)<B(D) GOTO 34
    0
335:GOTO 350
340:P=B(C):B(C)=B(D):B(D
    )=P
350:NEXT D
360:NEXT C
370:FOR E=0 TO N-1
380:LPRINT B(E)
390:NEXT E
400:LPRINT "": LPRINT ""
    : LPRINT "": END


561
```

MEMORY CONTENTS

A	
B	Loop counter
C	Loop counter
D	Loop counter
E	Loop counter
F	
G	
H	
I	
J	√
K	
L	
M	Character and number selection
N	Number of data
O	Increasing or decreasing order selection
P	For number sorting
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
B$(N−1)	Character data
C$(0)	For character sorting
B(N−1)	Number value data

Program Title: THE LOAN LIMIT, CALCULATION OF THE NUMBER OF INSTALLMENTS

OVERVIEW

Never borrow more than you can afford.
You will be able to calculate how much you are able to borrow and in how many installments you will be able to repay. Be realistic.

INSTRUCTIONS

DEF A ; The Size of the loan is calculated depending on your repayment ability (How much you have to repay and in how many installments.)

DEF B ; The number of installements is calculated from the size of the loan and your ability to repay the loan.

CONTENTS (calculation)

	Calculation of limit of loan size	Calculation of number installments
Input	Monthly installment amount (amount: a) Number of installments (n times) Interest rate (r%)	Loan size (total amount: A) Monthly installment amount (amount: a) Interest rate (r%)
Output	Loan size	Number of installments

$$\text{Loan size limit} = \frac{a(R^n - 1)}{(R-1)R^n}$$

$$\left(\text{where } R = 1 + \frac{r}{100}\right)$$

$$\text{Number of installments} = \frac{\log a - \log(a - A(R-1))}{\log R}$$

EXAMPLE

1. Calculation of the limit on the size of the loan when the repayment period is 8 years and the monthly installments are 1,500,000 at an annual interest rate of 12%.

Yearly payment	= 1500000 x 12	input
Number of installments	= 8	
Interest rate	= 12%	

2. Calculation of how many months it will take to repay the loan if you borrowed 3,000,000 at an annual interest rate of 12% and were capable of paying 100,000 back per month.

Loan size	= 3000000	input
Monthly installment amount	= 100000	
Interest rate	= 12/12%	

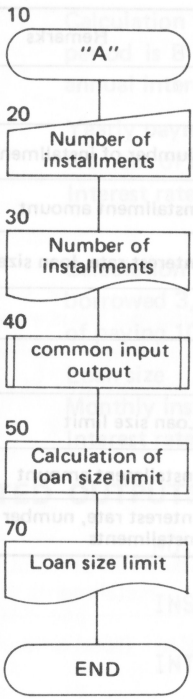
PRINTED OUTPUTS

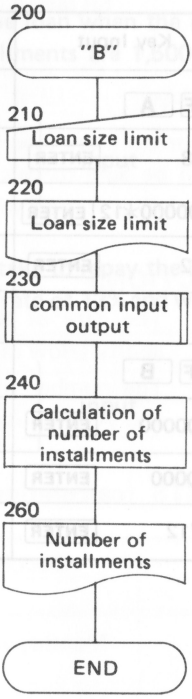
```
NO. OF INSTALLMENT=
                    8.00
INSTALLMENT AMOUNT=
              18000000
INTEREST RATE=
                12.000
LOAN SIZE LIMIT=
              89417515


LOAN SIZE LIMIT=
               3000000
INSTALLMENT AMOUNT=
                100000
INTEREST RATE=
                  1.000
NO. OF INSTALLMENT=
                 35.84
```

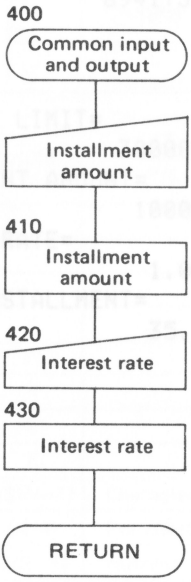
KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	NO. OF INSTALLMENT=	
2	8 ENTER	INSTALLMENT AMOUNT =	Number of installments
3	1500000*12 ENTER	INTEREST RATE =	Installment amount
4	12 ENTER	>	Interest rate, loan size limit
1	DEF B	LOAN SIZE LIMIT =	
2	3000000 ENTER	INSTALLMENT AMOUNT =	Loan size limit
3	100000 ENTER	INTEREST RATE =	Installment amount
4	12/12 ENTER	>	Interest rate, number of installments

FLOWCHART


10
"A"
20
Number of installments
30
Number of installments
40
common input output
50
Calculation of loan size limit
70
Loan size limit
END


200
"B"
210
Loan size limit
220
Loan size limit
230
common input output
240
Calculation of number of installments
260
Number of installments
END


400
Common input and output
Installment amount
410
Installment amount
420
Interest rate
430
Interest rate
RETURN

PROGRAM LIST

```
 10:"A": CLEAR
 20:INPUT "NO. OF INSTAL
    LMENT=";A
 30:LPRINT "NO. OF INSTA
    LLMENT=": LPRINT
    USING "###.##";A
 40:GOSUB 400
 50:J=(1+C/100)^A
 60:D= INT ((J-1)*B/(J*C
    /100))
 70:LPRINT "LOAN SIZE LI
    MIT=": LPRINT USING
    "##########";D
 80:LPRINT "": LPRINT ""
    : LPRINT "": END
200:"B": CLEAR
210:INPUT "LOAN SIZE LIM
    IT=";D
220:LPRINT "LOAN SIZE LI
    MIT=": LPRINT USING
    "##########";D
230:GOSUB 400
240:K=B/(B-D*C/100)
250:A= LOG K/ LOG (1+C/1
    00)
260:LPRINT "NO. OF INSTA
    LLMENT=": LPRINT
    USING "###.##";A
270:LPRINT "": LPRINT ""
    : LPRINT "": END
400:INPUT "INSTALLMENT A
    MOUNT=";B
410:LPRINT "INSTALLMENT
    AMOUNT=": LPRINT
    USING "##########";B
420:INPUT "INTEREST RATE
    =";C
430:LPRINT "INTEREST RAT
    E=": LPRINT USING "#
    ###.###";C
440:RETURN
450:END
```

MEMORY CONTENTS

A	Number of installments
B	Installment size
C	Interest rate
D	Loan size limit
E	
F	
G	
H	
I	
J	√
K	√
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

478

Program Title: TYPING PRACTICE

OVERVIEW

Quick key operation!

How fast and accurate is your typing?

If you practice with this program, it will make programming much easier for you. Improve your skill!

CONTENTS (such as calculation contents)

The number of characters (4 ~ 6) is randomly chosen.

The character arrangement (A ~ Z) is done randomly.

The allotted time depends on the number of characters and the grade level.

3 is the shortest time allotment while 1 is the longest.

INSTRUCTIONS

After the buzzer sounds 4 to 6 characters will be displayed. You are to type in the same characters within the allotted time.

If they are all correct, you get 10 coints.

If more than half are correct, you get 5 points.

After the allotted time is over, the next problem is displayed. The allotted time depends on the grade, which has three levels (1, 2, 3).

3 is the shortest time allotment while 1 is the longest.

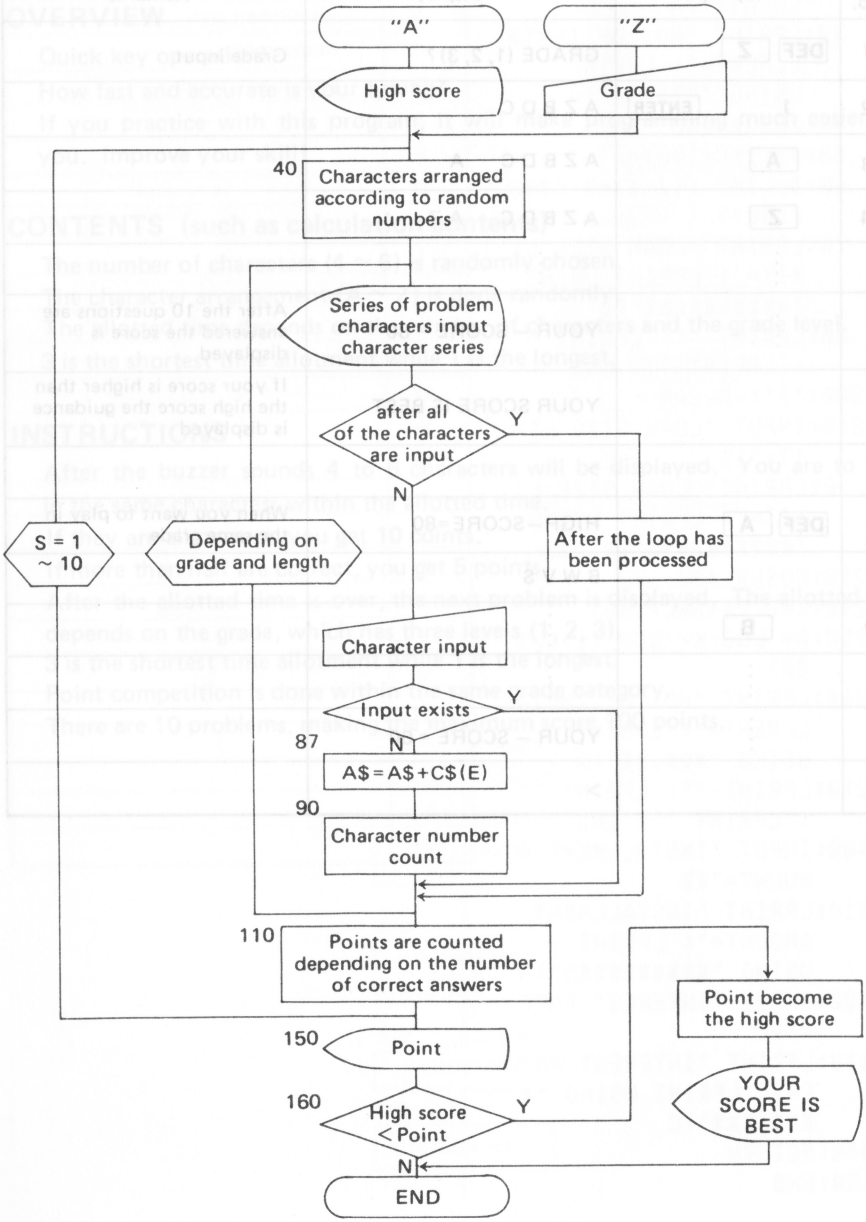
Point competition is done within the same grade category.

There are 10 problems, making the maximum score 100 points.

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF Z	GRADE (1, 2, 3)?	Grade input
2	1 ENTER	A Z B D C	
3	A	A Z B D C A	
4	Z	A Z B D C A Z	
	⋮	⋮	
	⋮	YOUR – SCORE = 80	After the 10 questions are answered the score is displayed
		YOUR SCORE IS BEST	If your score is higher than the high score the guidance is displayed
		>	
1	DEF A	HIGH–SCORE=80	When you want to play in the same grade
		B W V S	
2	B	⋮	
	⋮	⋮	
	⋮	YOUR – SCORE = 60	
		>	

FLOWCHART


"A"
"Z"
High score
Grade
40
Characters arranged according to random numbers
Series of problem characters input character series
after all of the characters are input
Y
N
S = 1 ~ 10
Depending on grade and length
After the loop has been processed
Character input
Input exists
Y
87
N
A$=A$+C$(E)
90
Character number count
110
Points are counted depending on the number of correct answers
Point become the high score
150
Point
160
High score < Point
Y
YOUR SCORE IS BEST
N
END

PROGRAM LIST

```
10:"Z": CLEAR : DIM B$(
   5),C$(5): RANDOM
15:INPUT "GRADE(1,2,3)?
   ";L: WAIT 0
17:IF (L=1)+(L=2)+(L=3)
   <>1 THEN 15
18:GOTO 30
20:"A": WAIT 0:P=0:
   PAUSE "HIGH-SCORE=";
   X
30:FOR S=1 TO 10
40:B= RND 4+2:Y$="":R=
   INT (B/2)
50:FOR C=0 TO B-1:C$(C)
   =""
60:D= RND 26:B$(C)=
   CHR$ (D+&40):Y$=Y$+
   CHR$ (D+&40): NEXT C
   :A$=""
70:BEEP 3:E=0: WAIT 30:
   USING "&&&&&&&&"
80:FOR W=1 TO B*10/L:
   PRINT Y$;"   ";A$:
   IF E=B LET W=B*20/L:
   GOTO 100
85:C$(E)= INKEY$ : IF C
   $(E)="" THEN 100
87:A$=A$+C$(E)
90:E=E+1
100:NEXT W:Q=0
110:FOR W=0 TO B-1: IF B
   $(W)=C$(W) LET Q=Q+1
120:NEXT W: IF Q<=R THEN
   150
130:IF Q=B LET P=P+10:
   GOTO 150
140:P=P+5
150:NEXT S: USING : BEEP
   3: PAUSE "YOUR-SCORE
   =";P
160:IF P>X LET X=P: WAIT
   100: PRINT "YOUR SCO
   RE IS BEST"
170:END
```

475

MEMORY CONTENTS

A$	√
B	√
C	Loop counter
D	√
E	√
F	
G	
H	
I	
J	
K	
L	Grade
M	
N	
O	
P	Score
Q	√
R	√
S	Loop counter
T	
U	
V	
W	Loop counter
X	High score
Y$	
Z	
B$ (5)	√
C$ (5)	√

Program Title: BIORHYTHM (SEMI-GRAPHIC)

OVERVIEW

You input your name and date of birth, the biorhythm for any specified month will be printed. Watch out for those dangerous days.

CONTENTS

Input: name
date of birth
desired year, month
The date of birth and the year, month to be analyzed are input in the western calender.

Output: The biorhythm curve of the month to be analyzed is printed out using the characteres "S", "K", and "C".

From the date of birth the biorhythm cycle for the physical is 23 days, for emotions 28 days, and for intellect 33 days.

INSTRUCTIONS

[DEF] [A] initiates the program. Input the information in the following order: name, date of birth, year and month to be analyzed. After the above informations are input the biorhytm is printed out using "S", "K", and "C".

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	[DEF] [A]	NAME = _	
	SHARP [ENTER]	DATE OF BIRTH: YEAR = _	
	1952 [ENTER]	MONTH = _	
	1 [ENTER]	DAY = _	
	28 [ENTER]	DESIRED: YEAR = _	
	1982 [ENTER]	MONTH = _	
	9 [ENTER]	>	

PRINTED OUTPUT

```
NAME:SHARP
DATE OF BIRTH:
    1952,1,28
DESIRED YEAR,MONTH:
    1982,9
```

```
      -     0     +
I---------+---------I
I  S  C    +  K       I  1
I    CS    +    K     I  2
I    C S   +       K  I  3
I  C     S +        K  I  4
I--C------S-------K--I  5
I  C        + S      K I  6
I  C        +    S  K  I  7
I  C        +     S K  I  8
I    C      +       KS  I  9
I----C-----+---K--S--I 10
I       C   +  K    S  I 11
I         C +K       S  I 12
I          CK        S  I 13
I          KC      S    I 14
I------K--+C-S------I 15
I      K   +SC         I 16
I    K     S+    C     I 17
I  K    S   +     C    I 18
I  K S      +       C  I 19
I-K-S-----+------C--I 20
I  K        +        C  I 21
I  K        +        C  I 22
I  SK       +        C  I 23
I  S  K     +        C  I 24
I----S-K--+-----C---I 25
I      S  K+       C   I 26
I        S K     C      I 27
I          SK C         I 28
I          + CK         I 29
I---------C---K-----I 30
I---------+---------I
```

(Remark)

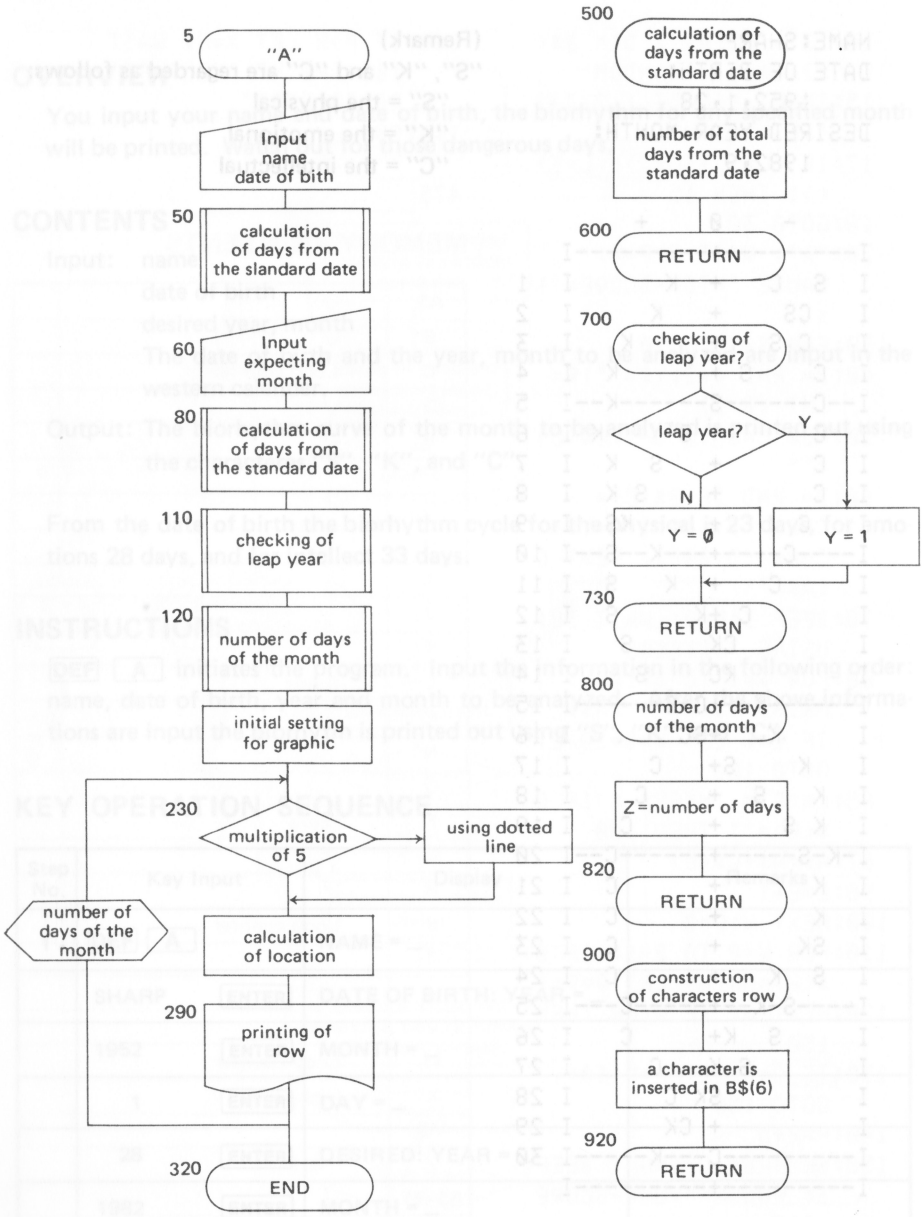
"S", "K" and "C" are regarded as follows:

"S" = the physical

"K" = the emotional

"C" = the intellectual

FLOWCHART


5
"A"
Input
name
date of bith
50
calculation
of days from
the slandard date
60
Input
expecting
month
80
calculation
of days from
the standard date
110
checking of
leap year
120
number of days
of the month
initial setting
for graphic
230
multiplication
of 5
using dotted
line
number of
days of the
month
calculation
of location
290
printing of
row
320
END
500
calculation of
days from the
standard date
number of total
days from the
standard date
600
RETURN
700
checking of
leap year?
leap year?
Y
N
Y = Ø
Y = 1
730
RETURN
800
number of days
of the month
Z=number of days
820
RETURN
900
construction
of characters row
a character is
inserted in B$(6)
920
RETURN

PROGRAM LIST

```
  5:"A": CLEAR : DEGREE
    : LPRINT ""
 10:INPUT "NAME=";D$
 20:LPRINT "NAME:";D$
 30:INPUT "DATE OF BIRTH
    :YEAR=";U$,"MONTH=";
    V,"DAY=";W
 35:F$= STR$ V:G$= STR$
    W
 40:LPRINT "DATE OF BIRT
    H:": LPRINT "    ";U
    $;",";F$;",";G$
 50:GOSUB 500:B=X
 60:INPUT "DESIRED:YEAR=
    ";U$,"MONTH=";V:W=0
 65:F$= STR$ V
 70:LPRINT "DESIRED YEAR
    ,MONTH:": LPRINT "
      ";U$;",";F$
 75:LPRINT ""
 80:GOSUB 500:A=X
100:C=A-B
110:GOSUB 700
120:GOSUB 800
130:D=C- INT (C/23)*23
140:E=C- INT (C/28)*28
150:F=C- INT (C/33)*33
160:DIM B$(2)*21
170:B$(1)="I----------+--
    -------I"
180:B$(2)="I          +
          I"
200:LPRINT "     -    0
      +"
210:LPRINT B$(1)
215:FOR K=1 TO Z
220:L=0: IF K=5* INT (K/
    5) LET L=1
230:IF L=1 LET B$(0)=B$(
    1): GOTO 250
240:B$(0)=B$(2)
250:G= SIN ((K+D)/23*360
    ):P$="S": GOSUB 900
260:G= SIN ((K+E)/28*360
    ):P$="K": GOSUB 900
270:G= SIN ((K+F)/33*360
    ):P$="C": GOSUB 900
290:LPRINT B$(0); USING
    "###";K
300:NEXT K
310:LPRINT B$(1)
320:END
500:T= VAL U$
560:IF V>=3 LET Q=T:R=V+
    1:S=W: GOTO 580
570:Q=T-1:R=13+V:S=W
580:X= INT (365.25*Q)+
    INT (30.6*R)+S-122
590:X=X- INT (Q/100)+
    INT (Q/400)
600:RETURN
700:IF T<>4* INT (T/4)
    LET Y=0: RETURN
710:IF T<>100* INT (T/10
    0) LET Y=1: RETURN
720:IF T<>400* INT (T/40
    0) LET Y=0: RETURN
730:Y=1: RETURN
800:IF (V=4)+(V=6)+(V=9)
    +(V=11)=1 LET Z=30:
    RETURN
810:IF V=2 LET Z=28+Y:
    RETURN
820:Z=31: RETURN
900:N=11+ INT (8* ABS G)
    * SGN G
910:B$(0)= LEFT$ (B$(0),
    N-1)+P$+ RIGHT$ (B$(
    0),21-N)
920:RETURN
```

957

MEMORY CONTENTS

A	√
B	√
C	Total number of days
D	remaining days from a period, name
E	Emotions
F, F$	Intellect
G, G$	Biorhythm curve, √
H	
I	
J	
K	Loop counter
L	√
M	
N	√
O	
P$	√
Q	√
R	
S	
T	√
U$	√
V	Month
W	Date of birth
X	√
Y	√
Z, Z$	Relerant number of days, √
B$(2)*21	√

Program Title: SOFTLANDING GAME

OVERVIEW

This game involves landing a rocket, with only a limited amount of fuel, as softly as possible. The rocket is in free fall. The engine is used to slow down the free falling rocket. If ignition takes place too soon or too much fuel is used, then the rocket is thrust back out into space and becomes dust around the planet.
If all the fuel is burned up, the rocket hits the planet and blows up.
The aim is to land the rocket as softly as possible by controlling the engines while watching how much fuel is burned.

CONTENTS

Gravity is set to be 5 m/(unit time)^2.
If 5 units of fuel per a unit time are burnt, then gravity is offset.

Equations

$$H = H_0 + V_0 t + \frac{1}{2} a t^2$$

$$V = V_0 + a t$$

$$V^2 = V_0^2 + 2aH$$

$$H_0 = 500, \quad V_0 = -50, \quad F_0 = 200$$

H : height
V : speed
a : gravitational acceleration
t : time
V_0 : initial speed

H_0 : initial height
V_0 : initial speed
F_0 : initial fuel
F : fuel burned

The initial height, initial fuel level, and the wait time is stored in line 30 as data. By changing these values the above variables can be changed.

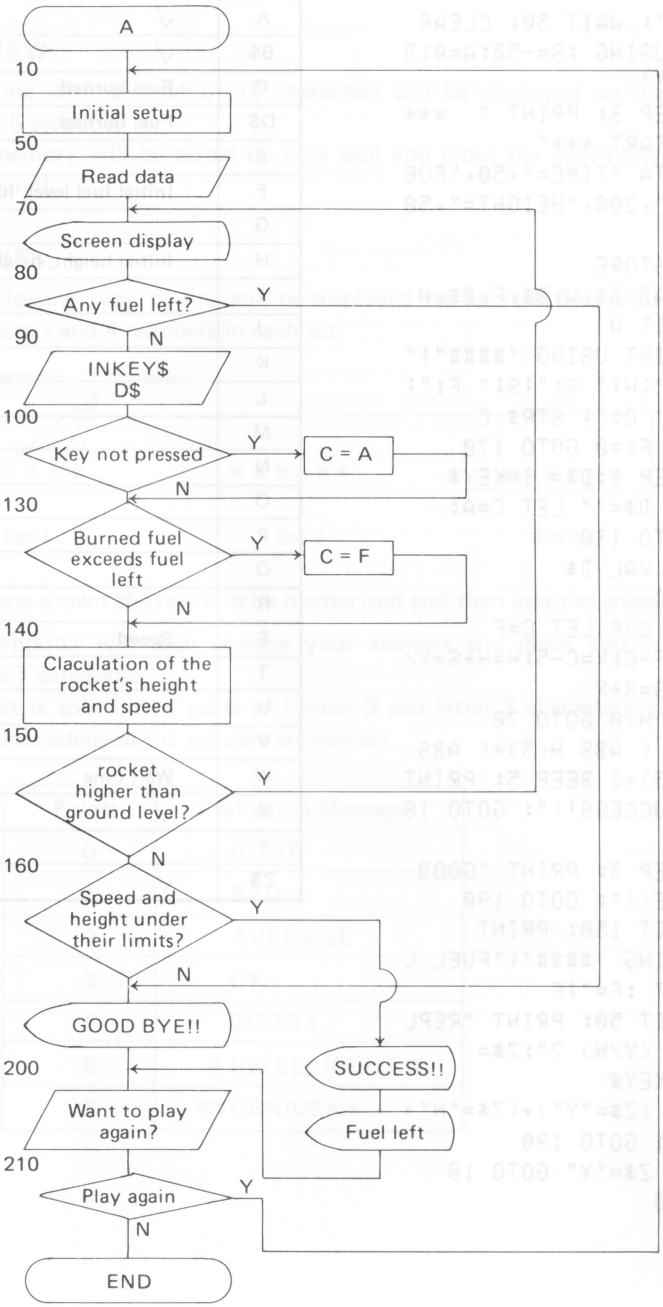
INSTRUCTIONS

1. It is started by pressing [DEF] [A] . Press [0] ~ [9] keys to adjust the amount of fuel used to land the rocket.

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	***START***	
2	Keys 0 ~ 9 designate fuel burned in unit time	H: 500 S: −50 F: 200 C:0	
	9	H: 452 S: −46 F: 191 C:9	
	⋮		
	⋮	Repeat	
	⋮		
	(If successful)	SUCCESS !!	
		FUEL LEFT: F = 15	
	(If failed)	GOOD BYE!!	
		REPLAY (Y/N)?	Wait for input on whether you wish to play again
	Y		Play again
	N	>	End

FLOWCHART


A
10
Initial setup
50
Read data
70
Screen display
80
Any fuel left?
Y
90
N
INKEY$
D$
100
Key not pressed
Y
C = A
N
130
Burned fuel exceeds fuel left
Y
C = F
N
140
Claculation of the rocket's height and speed
150
rocket higher than ground level?
Y
N
160
Speed and height under their limits?
Y
N
GOOD BYE!!
200
Want to play again?
210
Play again
Y
N
END
SUCCESS!!
Fuel left

PROGRAM LIST

```
 10:"A": WAIT 50: CLEAR
    : USING :S=-50:A=0:D
    $=""
 20:BEEP 3: PRINT "  ***
    START ***"
 30:DATA "TIME=",50,"FUE
    L=",200,"HEIGHT=",50
    0
 40:RESTORE
 50:READ B$,W,B$,F,B$,H
 60:WAIT W
 70:PRINT USING "####";"
    H:";H;" S:";S;" F:";
    F;" C:"; STR$ C
 80:IF F<=0 GOTO 170
 90:BEEP 1:D$= INKEY$
100:IF D$="" LET C=A:
    GOTO 130
110:C= VAL D$
120:A=C
130:IF C>F LET C=F
140:F=F-C:X=C-5:H=H+S+X/
    2:S=S+X
150:IF H>0 GOTO 70
160:IF ( ABS H<5)+( ABS
    S<5)=2 BEEP 5: PRINT
    "SUCCESS!!": GOTO 18
    0
170:BEEP 3: PRINT "GOOD
    BYE!!": GOTO 190
180:WAIT 150: PRINT
    USING "####";"FUEL L
    EFT :F=";F
190:WAIT 50: PRINT "REPL
    AY (Y/N) ?":Z$=
    INKEY$
200:IF (Z$="Y")+(Z$="N")
    <>1 GOTO 190
210:IF Z$="Y" GOTO 10
220:END

438
```

MEMORY CONTENTS

A	√
B$	√
C	Fuel burned
D$	Fuel burned
E	
F	Initial fuel level, fuel left
G	
H	Initial height, height
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	Speed
T	
U	
V	
W	Wait time
X	√
Y	
Z$	√

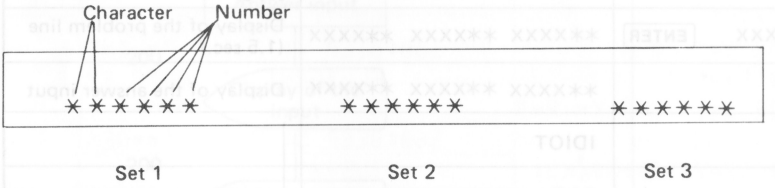
Program Title: MEMORY CHECKER

OVERVIEW

Three line with a total of 18 characters will be displayed on the screen for approx. 5 seconds.
Your memory will be tested by how well you input the above line after it has disappeared.

CONTENTS

The following type of line will be displayed for approx. 5 seconds. There are 2 characters and 4 numbers in each set.



The 3 sets shown above are to be memorized and then input as answers.

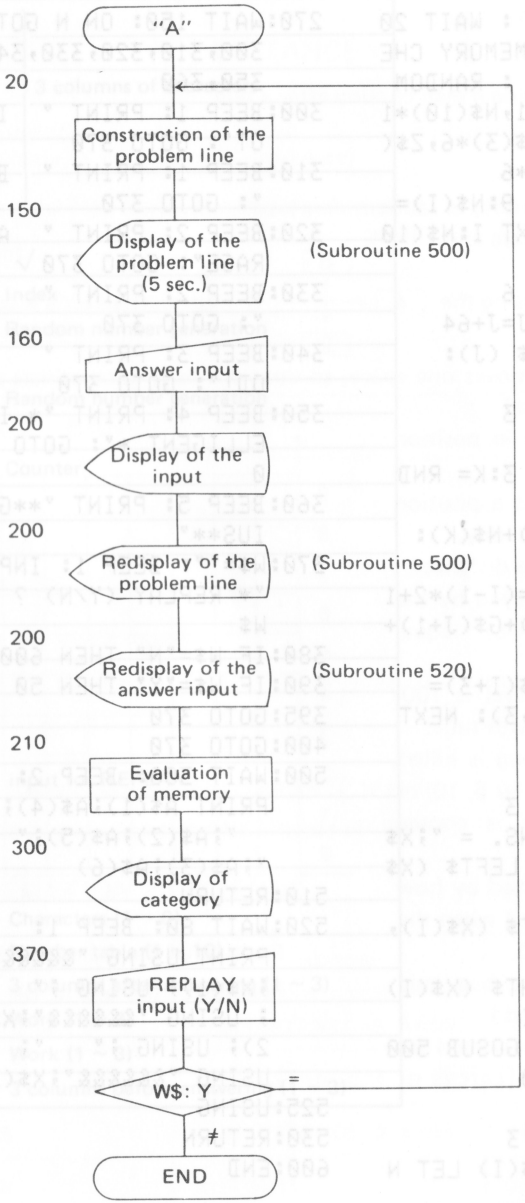
The computer will then analyze your answers and place you in one of the possible 7 categories.
Each set is split into 2 parts of former 3 and latter 3 characters, giving a total of 6 points when all the answers are correct.

Points	Evaluation Message
0	IDIOT
1	BAD
2	AVERAGE
3	OK
4	GOOD !
5	✳ INTELLIGENT ✳
6	✳✳GENIUS✳✳

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	MEMORY CHECK	Title
2		✳✳XXXX ✳✳XXXX ✳✳XXXX	Display of problem line (5 sec.) ✳ . . . character X . . . number
3		ANS. = _	Waiting for the input of set 1
4	(Example) AB1234 ENTER	ANS. = _	Waiting for the input of set 2
5	✳✳XXXX ENTER	ANS. = _	Waiting for the input of set 3
6	✳✳XXXX ENTER	✳✳XXXX ✳✳XXXX ✳✳XXXX	Display of the problem line (1.5 sec.)
7		✳✳XXXX ✳✳XXXX ✳✳XXXX	Display of the answer input
8		IDIOT	display of category
		BAD	
		AVERAGE	
		OK	
		GOOD !	
		✳ INTELLIGENT ✳	
		✳✳GENIUS✳✳	
9		✳REPLAY (Y/N) ?	Player input request
10	Y or N ENTER		If Y, go to step 2
		>	If N, END

FLOWCHART


"A"
20
Construction of the problem line
150
Display of the problem line (5 sec.)
(Subroutine 500)
160
Answer input
200
Display of the input
200
Redisplay of the problem line
(Subroutine 500)
200
Redisplay of the answer input
(Subroutine 520)
210
Evaluation of memory
300
Display of category
370
REPLAY input (Y/N)
W$: Y
=
≠
END

PROGRAM LIST

```
 10:"A": USING : WAIT 20
    0: PRINT "MEMORY CHE
    CK": CLEAR : RANDOM
 20:DIM G$(6)*1,N$(10)*1
    ,V$(3)*3,X$(3)*6,Z$(
    3)*3,Y$(3)*6
 30:FOR I=1 TO 9:N$(I)=
    STR$ I: NEXT I:N$(10
    )="0"
 50:FOR I=1 TO 6
 60:J= RND 26:J=J+64
 70:G$(I)= CHR$ (J):
    NEXT I
 80:FOR I=1 TO 3
 90:Y$(I)=" "
100:FOR J=1 TO 3:K= RND
    9
110:Y$(I)=Y$(I)+N$(K):
    NEXT J
120:L= RND 9:J=(I-1)*2+1
130:A$(I)=G$(J)+G$(J+1)+
    N$(L)
140:H$=Y$(I):A$(I+3)=
    RIGHT$ (H$,3): NEXT
    I
150:GOSUB 500
160:FOR I=1 TO 3
170:INPUT "  ANS. = ";X$
    (I):X$(I)= LEFT$ (X$
    (I),6)
180:Z$(I)= LEFT$ (X$(I),
    3)
190:V$(I)= RIGHT$ (X$(I)
    ,3): NEXT I
200:GOSUB 520: GOSUB 500
    : GOSUB 520
210:N=0
220:FOR I=1 TO 3
230:IF A$(I)=Z$(I) LET N
    =N+1
240:IF A$(I+3)=V$(I) LET
    N=N+1
250:NEXT I
260:N=N+1
270:WAIT 150: ON N GOTO
    300,310,320,330,340,
    350,360
300:BEEP 1: PRINT "  IDI
    OT": GOTO 370
310:BEEP 1: PRINT "  BAD
    ": GOTO 370
320:BEEP 2: PRINT "  AVE
    RAGE": GOTO 370
330:BEEP 2: PRINT "   OK
    ": GOTO 370
340:BEEP 3: PRINT "   GO
    OD!": GOTO 370
350:BEEP 4: PRINT "* INT
    ELLIGENT *": GOTO 37
    0
360:BEEP 5: PRINT "**GEN
    IUS**"
370:W$="": BEEP 1: INPUT
    "* REPLAY (Y/N) ? ";
    W$
380:IF W$="N" THEN 600
390:IF W$="Y" THEN 50
395:GOTO 370
400:GOTO 370
500:WAIT 300: BEEP 2:
    PRINT A$(1);A$(4);"
      ";A$(2);A$(5);"
    ";A$(3);A$(6)
510:RETURN
520:WAIT 80: BEEP 1:
    PRINT USING "&&&&&&"
    ;X$(1); USING ;"   "
    ; USING "&&&&&&";X$(
    2); USING ;"   ";
    USING "&&&&&&";X$(3)
525:USING
530:RETURN
600:END

 891
```

MEMORY CONTENTS

A$	3 columns of characters
B$	
C$	
D$	
E$	
F$	
G	
H$	√
I	Index
J	Random number generation
K	
L	Random number generation
M	
N	Counter
O	
P	
Q	
R	
S	
T	
U	
V	
W$	input for REPLAY
X	
Y	
Z	
G$(6)*1	Characters (1 ~ 6)
N$(10)*1	Number table (1 ~ 10)
V$(3)*3	3 columns after answering (1 ~ 3)
X$(3)*6	Work (1 ~ 3)
Y$(3)*6	Work (1 ~ 3)
Z$(3)*3	3 columns before answering (1 ~ 3)

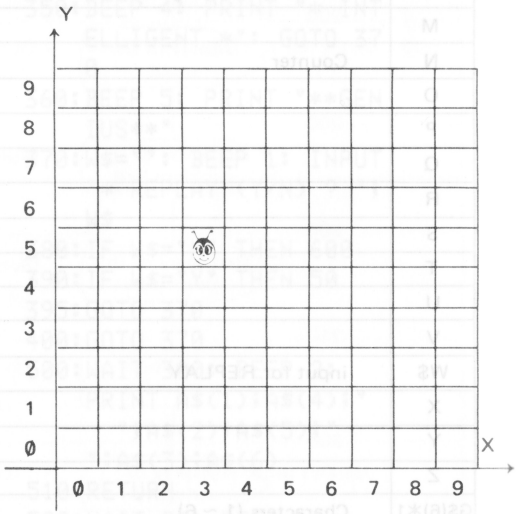
Program Title: **BUGHUNT**

OVERVIEW

This is a game involving a man chasing after a bug.

CONTENTS

The bug moves according to random numbers.
The man chases the bug and kills it.
The man moves by using the [8] [4] [6] [2] keys. (INKEY$ is used)
Each time the man moves one space, so does the bug. (Sometimes the bug will stay in the same place)
Initially the man is in position (Ø, Ø).
The bug is placed at a position that was chosen randomly.
Hints are displayed as distance.
The distance is displayed by the ABS(X−a)+ABS(Y−b) equation.
The initial energy level is 100. This decreases by 1 with time.
Each time that a bug is killed, the energy increases by 5, 10, or 15. (The amount is chosen randomly.)
The score is determined by how many bugs were killed when the energy level reaches 0.
(The position of the bug may "warp" when cornered.)
The program can be started by either pressing RUN [ENTER] or [DEF] [A] .



Position of the man (X, Y)
Position of the bug (a, b)

Concerning the display

(Small characters are actual values)

```
(x, y)          DISTANCE = ℓ      E = e
```

Present position (X coordinate, Y coordinate) — Hint (distance) — Remaining energy

- Each time the man moves the display changes

Bug is caught

```
HIT!    HIT!
```

```
BANG!  BANG!
```

```
SCORE  t   ENERGY  e
```

Concerning the BEEP sound

★ Hint: When the distance is 1 the BEEP goes off 3 times
2 " 2
3 " 1

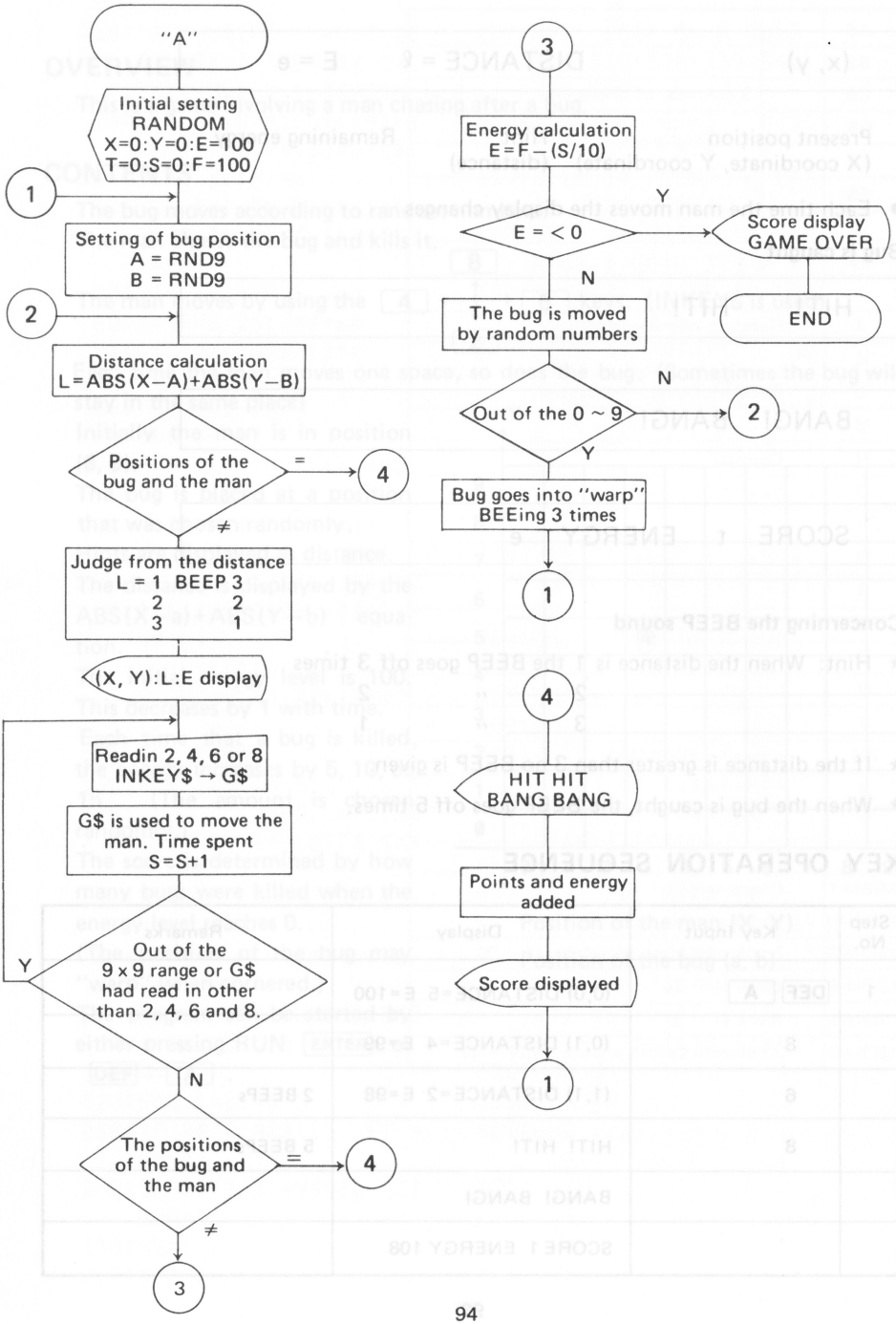
★ If the distance is greater than 3 no BEEP is given.

★ When the bug is caught, the BEEP goes off 5 times.

KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	(0,0) DISTANCE=5 E=100	
	8	(0,1) DISTANCE=4 E=99	
	6	(1,1) DISTANCE=2 E=98	2 BEEPs
	8	HIT! HIT!	5 BEEPs
		BANG! BANG!	
		SCORE 1 ENERGY 108	

FLOWCHART


"A"
Initial setting
RANDOM
X=0:Y=0:E=100
T=0:S=0:F=100
1
Setting of bug position
A = RND9
B = RND9
2
Distance calculation
L=ABS(X−A)+ABS(Y−B)
Positions of the bug and the man
=
4
≠
Judge from the distance
L = 1 BEEP 3
2 2
3 1
(X, Y):L:E display
Readin 2, 4, 6 or 8
INKEY$ → G$
G$ is used to move the man. Time spent
S=S+1
Out of the 9 x 9 range or G$ had read in other than 2, 4, 6 and 8.
Y
N
The positions of the bug and the man
=
4
≠
3
3
Energy calculation
E=F−(S/10)
E = < 0
Y
Score display
GAME OVER
END
N
The bug is moved by random numbers
Out of the 0 ~ 9
N
2
Y
Bug goes into "warp"
BEEing 3 times
1
4
HIT HIT
BANG BANG
Points and energy added
Score displayed
1

PROGRAM LIST

```
 10:"A": RANDOM : WAIT 2
    50: PRINT "** BUGHUN
    T GAME **": BEEP 3
 20:X=0:Y=0:E=100:F=100:
    T=0:S=0
 30:A= RND 9:B= RND 9
 40:L= ABS (X-A)+ ABS (Y
    -B)
 50:IF X=A AND Y=B GOTO
    400
100:IF L=1 BEEP 3
110:IF L=2 BEEP 2
120:IF L=3 BEEP 1
130:WAIT 50: PRINT "(";
    STR$ (X);",";  STR$ (
    Y);") DISTANCE=";
    STR$ (L);" E="; STR$
    (E)
150:S=S+1:E=F- INT (S/2)
153:IF E<=0 THEN 500
155:G$= INKEY$ : IF G$="
    " GOTO 130
157:BEEP 1
160:IF G$="2" LET Y=Y-1:
    GOTO 210
170:IF G$="4" LET X=X-1:
    GOTO 210
180:IF G$="6" LET X=X+1:
    GOTO 210
190:IF G$="8" LET Y=Y+1:
    GOTO 210
200:GOTO 150
210:IF X<0 LET X=0: GOTO
    150
220:IF Y<0 LET Y=0: GOTO
    150
230:IF X>9 LET X=9: GOTO
    150
240:IF Y>9 LET Y=9: GOTO
    150
250:IF X=A AND Y=B GOTO
    400
260:E=F- INT (S/2)
270:IF E<=0 GOTO 500
280:R= RND 5
290:IF R=1 LET B=B-1:
    GOTO 340
300:IF R=2 LET A=A-1:
    GOTO 340
310:IF R=3 LET A=A+1:
    GOTO 340
320:IF R=4 LET B=B+1:
    GOTO 340
340:IF A<0 OR A>9 GOTO 3
    70
350:IF B<0 OR B>9 GOTO 3
    70
360:GOTO 40
370:BEEP 4: PAUSE "*** W
    ARP ***": GOTO 30
400:PAUSE "HIT! HIT!"
410:BEEP 5
420:PAUSE "BANG!  BANG!"
430:T=T+1:C= RND 3*5:F=F
    +C
435:E=F- INT (S/2)
440:WAIT 100: PRINT "SCO
    RE ";T;" ENERGY ";E
450:GOTO 30
500:WAIT : PRINT "SCORE
    "; STR$ (T);" *GAME
    OVER!!*"
510:END

732
```

MEMORY CONTENTS

A	Position of bug X coordinate
B	Position of bug Y coordinate
C	Amount of energy added
D	
E	Remaining energy
F	Energy level
G$	Key read in
H	
I	
J	
K	
L	Distance between bug and man
M	
N	
O	
P	
Q	
R	Size of bug movement
S	Time spent
T	Score
U	
V	
W	
X	Man position X coordinate
Y	Man position Y coordinate
Z	

Program Title: DOUBLE ROTATION

OVERVIEW

Quickly put in order A, B, C · · · · ·

This is a game that arranges randomly placed characters (A – J) in alphabetical order. When the letters are arranged in the right order, a score is displayed. The trick is to attack from the best place.

The sooner the characters are arranged, the better.

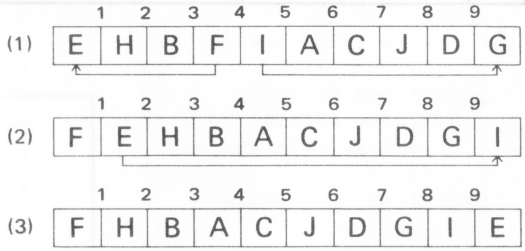
It is fun to race with 2 or 3 of your friends.

INSTRUCTIONS

1. After the program is initiated, by pressing DEF A , "DOUBLE ROTATION" is displayed. A random sequence of characters (A – J) is then displayed.
2. The space in between the characters is taken as the break points (1 – 9) where the numbers are placed. Inputing a break number causes the characters on each side of the breakpoint to be rotated by moving them to the far ends of the row.
3. After the characters have been placed in order, the number of moves required is displayed as the score. The lower the score the better.

EXAMPLE

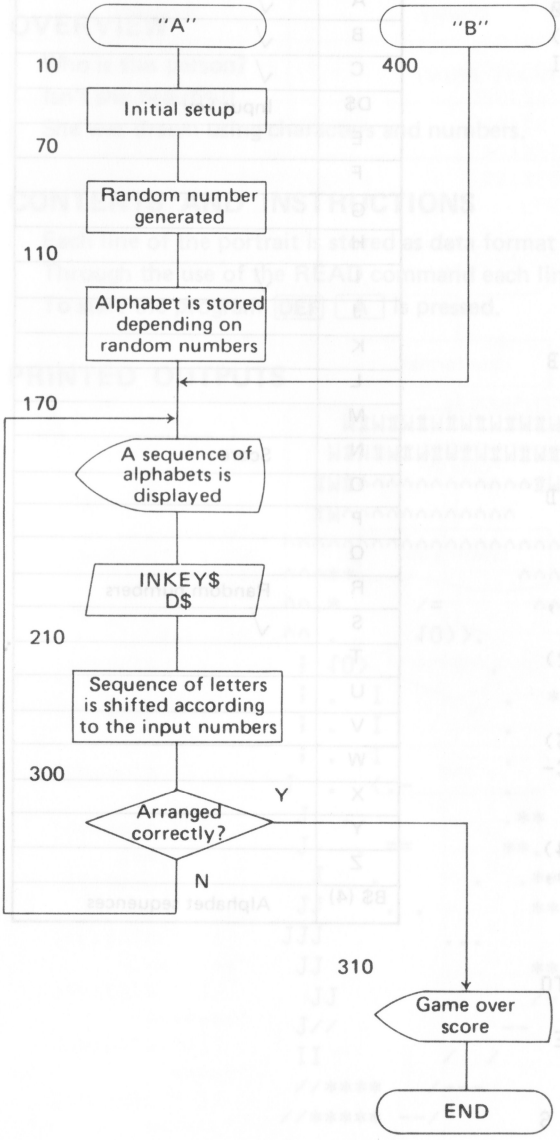
In (1) 4 is input, "F" and "I" move to each side changing the configuration to (2). If 1 is now input, the "E" moves to the far right but "F" stays in its place because it is already in the far left position, becoming configuration (3).



KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	DOUBLE ROTATION	
		A ~ J	Random requence display
2	1 ~ 9	⋮	Numbers between 1 and 9 are selected and input
		Repeated input	
		⋮	
		ABCDEFGHIJ	
		GAME END	
		YOUR SCORE 35	
		>	
			Does player want to play using the same beginning random alphabets?
1	DEF B	A ~ J	
		Same as DEF A in succession	

FLOWCHART


"A"
"B"
10
400
Initial setup
70
Random number generated
110
Alphabet is stored depending on random numbers
170
A sequence of alphabets is displayed
INKEY$
D$
210
Sequence of letters is shifted according to the input numbers
300
Arranged correctly?
Y
N
310
Game over score
END

PROGRAM LIST

```
 10:"A": CLEAR : WAIT 50
    : RANDOM : DIM B$(4)
 20:PAUSE "DOUBLE ROTATI
    ON"
 30:B$(0)="ABCDEFGHIJ"
 40:B$(1)=""
 50:A=0
 60:FOR I=1 TO 10
 70:R= RND 10
 80:S=2^(R-1)
 85:B=S AND A
 90:IF B<>0 GOTO 70
100:A=A OR S
110:B$(1)=B$(1)+ MID$ (B
    $(0),R,1): NEXT I
120:B$(2)=B$(1)
130:N=0
150:BEEP 1
170:D$="": PRINT B$(2):D
    $= INKEY$
180:C= VAL D$
190:IF C=0 GOTO 170
210:B$(3)= LEFT$ (B$(2),
    C)
220:B$(4)= RIGHT$ (B$(2)
    ,10-C)
240:IF C=1 GOTO 260
250:B$(3)= RIGHT$ (B$(3)
    ,1)+ LEFT$ (B$(3),C-
    1)
260:IF C=9 GOTO 280
270:B$(4)= RIGHT$ (B$(4)
    ,9-C)+ LEFT$ (B$(4),
    1)
280:B$(2)=B$(3)+B$(4)
290:N=N+1
300:IF B$(2)<>B$(0) GOTO
    150
310:BEEP 5: PAUSE "GAME
    END"
320:WAIT 200: PRINT
    USING "####";"YOUR S
    CORE";N
330:END
400:"B": WAIT 50: GOTO 1
    20

471
```

MEMORY CONTENTS

A	√
B	√
C	√
D$	Input key
E	
F	
G	
H	
I	√
J	
K	
L	
M	
N	Score
O	
P	
Q	
R	Random numbers
S	√
T	
U	
V	
W	
X	
Y	
Z	
B$ (4)	Alphabet sequences

Program Title: **PORTRAIT PRINT**

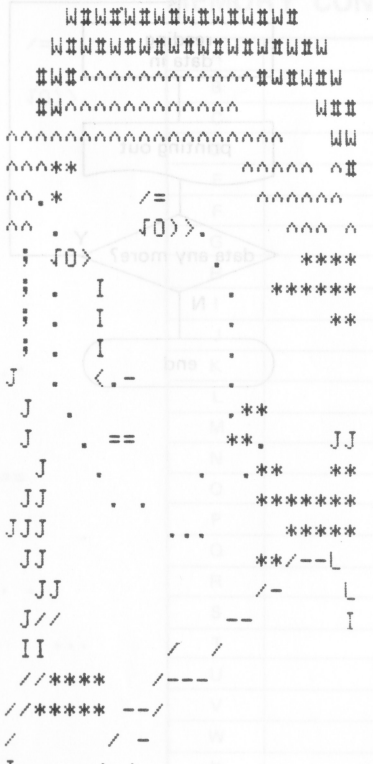
OVERVIEW

Who is this person?
Isn't she beautiful!
She was drawn using characters and numbers.

CONTENTS AND INSTRUCTIONS

Each line of the portrait is stored as data format in BASIC.
Through the use of the READ command each line is read-in and printed.
To start the program DEF A is pressed.

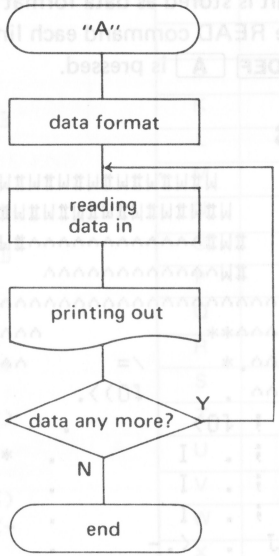
PRINTED OUTPUTS



KEY OPERATION SEQUENCE

Step No.	Key Input	Display	Remarks
1	DEF A	PORTRAIT PRINT	
		>	

FLOWCHART



PROGRAM LIST

```
10:"A": CLEAR : WAIT 0:
   DIM B$(0)*24
20:RESTORE
25:BEEP 3: PAUSE "PORTR
   AIT PRINT"
30:DATA "    W#W#W#W#W#W
   W#W#W#W#"
40:DATA "   W#W#W#W#W#W#W
   #W#W#W#W#W"
50:DATA "  #W#^^^^^^^^^^
   ^^^#W#W#W#W"
60:DATA "  #W^^^^^^^^^^^
   ^^     W##"
70:DATA "^^^^^^^^^^^^^^^
   ^^^^^   WW"
80:DATA "^^^**
     ^^^^^ ^#"
90:DATA "^^.*      /=
     ^^^^^^"
100:DATA "^^ .      J0)>.
       ^^^ ^"
110:DATA " ; J0>
   .     ****"
120:DATA " ; .  I
    .  ******"
130:DATA " ; .  I
   .      **"
140:DATA " ; .  I
    ."
150:DATA "J  .  <.-
    ."
160:DATA " J  .
    .**"
170:DATA " J   . ==
    **.    JJ"
180:DATA "  J   .      .
    .**   **"
190:DATA " JJ    . .
      *******"
200:DATA "JJJ        ...
       *****"
210:DATA " JJ
      **/--L"
220:DATA "  JJ
      /-    L"
230:DATA " J//
    --      I" .
240:DATA " II       /
   /"
250:DATA " //****   /---
    "
260:DATA "//***** --/"
270:DATA "/      / -"
280:DATA "I     /-/"
290:DATA "      /"
295:DATA ""
300:BEEP 1: READ B$(0):
    LPRINT B$(0)
310:IF B$(0)="" END
320:GOTO 300
```

794

MEMORY CONTENTS

A	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
B$(0)*24	FOR READING DATA

SHARP ELECTRONICS CORPORATION

CORPORATE HEADQUARTERS AND EXECUTIVE OFFICES:
10 Sharp Plaza, Paramus, New Jersey 07652. Phone: (201) 265-5600
REGIONAL SALES OFFICES AND DISTRIBUTION CENTERS:
Eastern: 10 Sharp Plaza, Paramus, New Jersey 07652. Phone: (201) 265-5600
Midwest: 430 East Plainfield Road, Country Side, Illinois 60525. Phone: (312) 242-0870
Western: Sharp Plaza 20600 South Alameda St., Carson, California 90810. Phone: (213) 637-9488

Printed in Jap
2M3.4T(TMANE1041CCZ