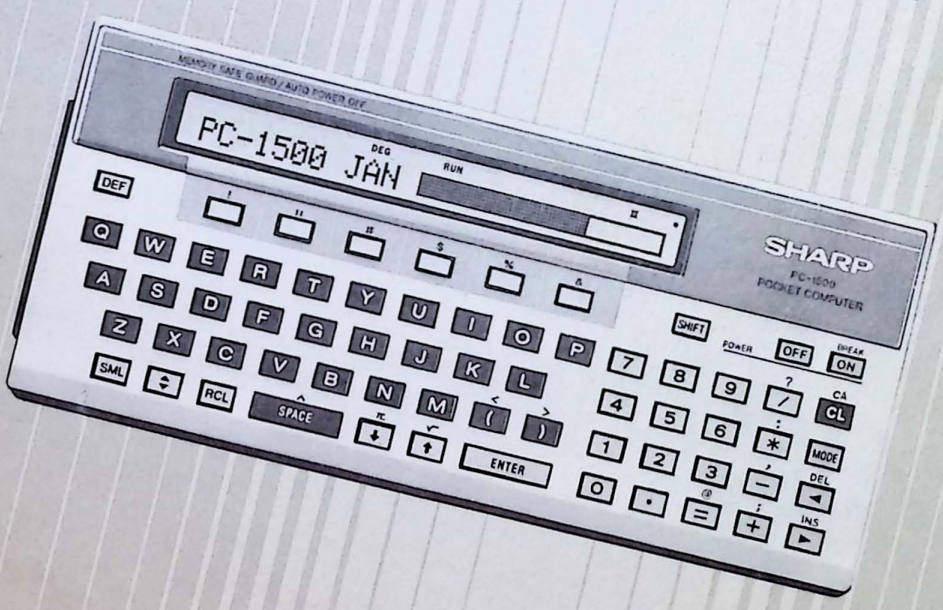
SHARP

POCKET COMPUTER

APPLICATIONS MANUAL

PC-1500



MODEL PC-1500, 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.

SHARP POCKET COMPUTER PC-1500

APPLICATIONS MANUAL

Thank you very much for purchasing the Sharp PC-1500 pocket computer.
This applications manual presents application softwares in various fields. To get the most out of your Sharp PC-1500 pocket computer, please try to make your own softwares that match your needs by referring to this manual.
This manual is editted according to the followings, so please read them carefully before use.

- **PROGRAM TITLE:**
  This is a summary of the program contents.

- **PROGRAM NO.**
  P5 stands for PC-1500, while A, B, C, D, E and F show program fields.
  The program numbers are not always in series. There are numbers skipped.

- **Hardware Configuration:**
  Optional equipments required to execute the program are shown right below the PROGRAM NO., if any.
  CE-150; color graphic printer/cassette interface
  CE-151; memory module
  CTR ; cassette tape recorder

- **Outline:**
  The brief explanation (concept) of the program is shown.

- **Operating Guide:**
  Shows the brief explanation of how to use and operate this program according to the "Key Operation Procedure" explained later.

- **Example:**
  For a better understanding of the program execution, an example using the program is provided.

- **Contents (Formulas):**
  To let you understand the logics employed in the program such as formulas are explained.

- **Printout:**
  Printouts through the optional color graphic printer (CE-150) are provided by using the example. (The character size is 18 char./line.)

- **Key Operation Procedure:**
  For your program execution, the actual key operation procedure is shown step by step by using the example.

- **Program List:**
  Printouts of eighteen characters per line through the CE-150 are listed in full size or reduction.

  How to enter the programs into the machine.

  The Program List shown in this applications manual 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.
  3) The numeral one (1) and the letter I look alike on the program list, so you can not be too careful.

  For more details, refer to page 26on of the instruction manual of the PC-1500.

  * Please make sure that you read through the instruction manual first, then try to type in the programs listed in this applications manual.
  * Also make sure that you use these programs after through checks through such as the examples.
  * **Sharp Corporation and/or its subsidiaries assume no responsibilities or obligations to any losses or damages that could arise through the use of this applications manual.**

- **Memory Contents:**
  Memory contents during the program execution are explained.

* **Constants, such as tax rates, if any, may vary from country to country or district to district.**
  They may also be subject to changes according to the revisions of laws and regulations, or other reasons. So, please be careful when you use these programs as they are listed here.
* For continuous improvements and additions, these programs are subject to change without notices.
* To help us improve our programs, we would appreciate any suggestions or comments in writing.

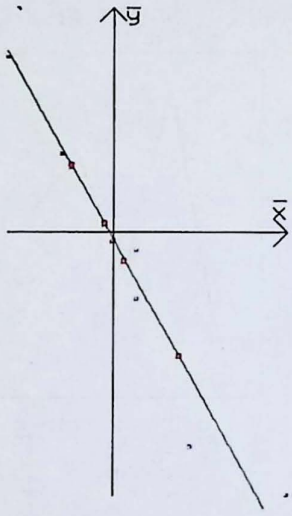
SHARP POCKET COMPUTER PC-1500
APPLICATIONS MANUAL

CONTENTS

COLOR PRINTOUTS

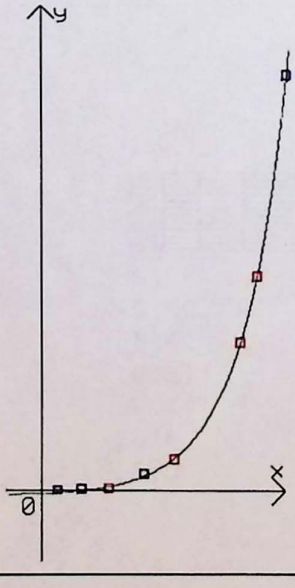
CORRELATION COEFFICIENT, LINEAR REGRESSION AND PLOT

(Refer to page 37.)



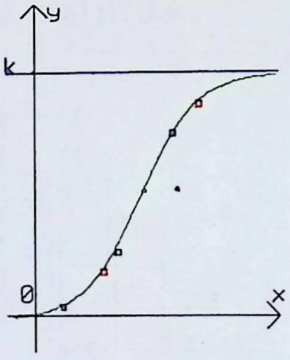
EXPONENTIAL REGRESSION AND PLOT

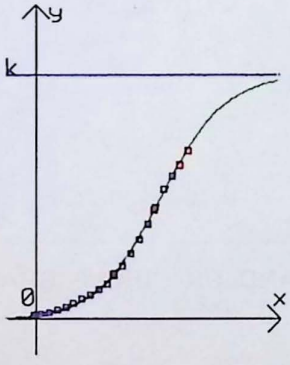
(Refer to page 41.)



LOGISTIC CURVE

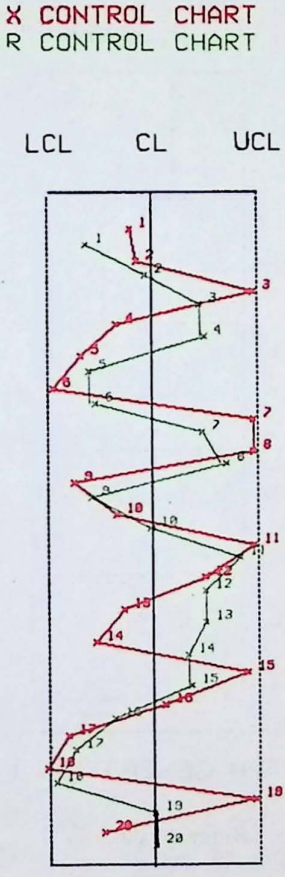
(Refer to page 49.)





$\overline{X}$ – R CONTROL CHART

(Refer to page 75.)

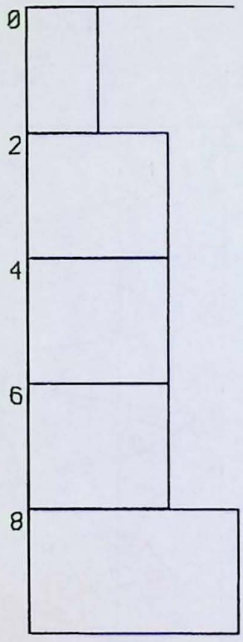


COLOR PRINTOUTS

HISTOGRAM

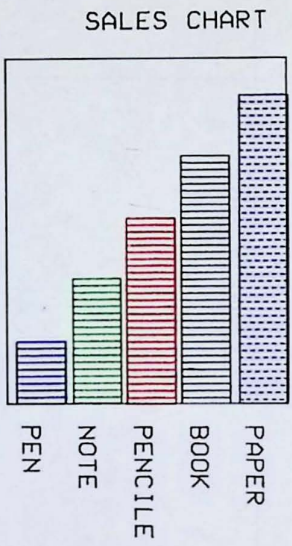
(Refer to page 97.)

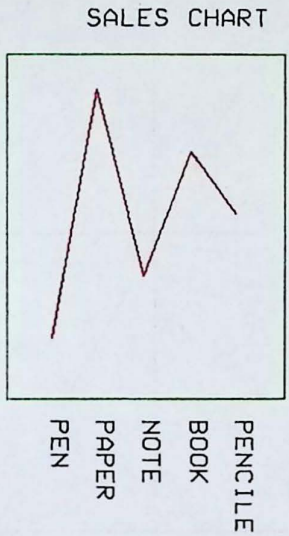
VARIANCE=
6.81
STD. DEV.=
2.60959767



GRAPH GENERATION II (BAR OR BROKEN LINE GRAPH)

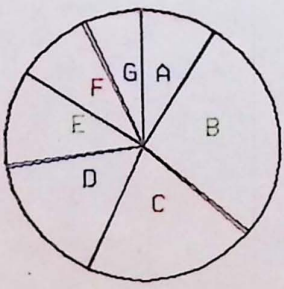
(Refer to page 104.)

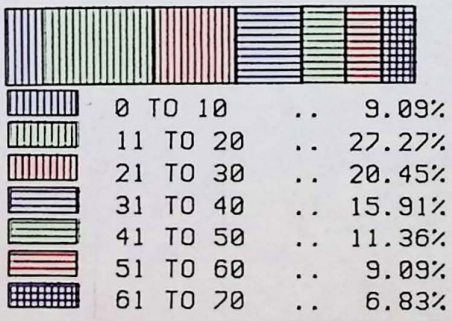




GRAPH GENERATION I (BAND OR CIRCLE GRAPH)

(Refer to page 100.)





INVENTORY CONTROL

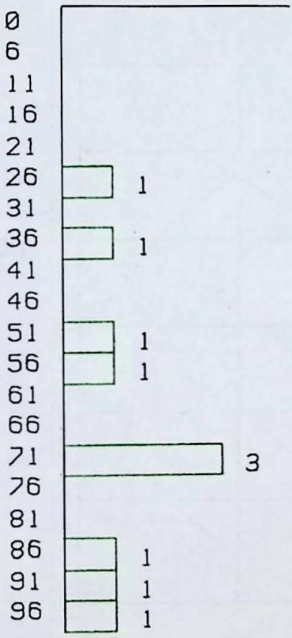
(Refer to page 135.)

```
**  TABLE  **
  1 DESK
        500     250
  2 BED
        100     200
  3 CHAIR
        500     350


PRESENT STOCK LIST
  2 BED
        100     200


**DATA LIST**
  1      50      40
  2      50      10


**MASTER TABLE**
  1 DESK
        500     250
  2 BED
        100     200
  3 CHAIR
        500     350


**  TABLE  **
  1 DESK
        490     250
  2 BICYCLE
         60     200
  3 CHAIR
        500     350
  4 TABLE
        150     100


PRESENT STOCK LIST
  2 BICYCLE
         60     200
```

MANAGEMENT OF STUDENTS' ACHIEVEMENTS

(Refer to page 143.)

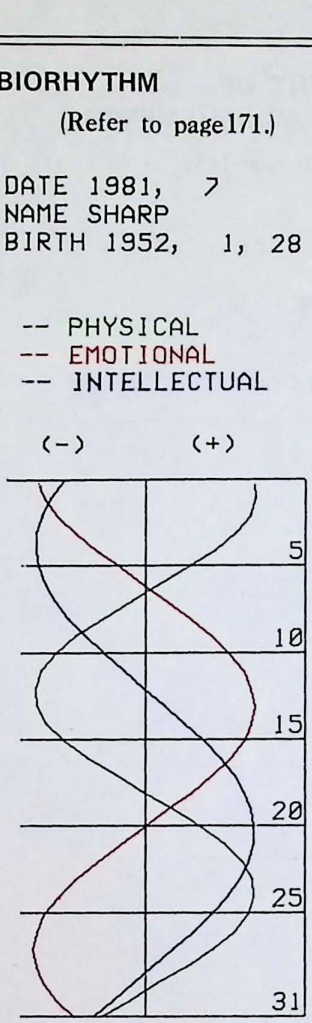
```
AVG. OF ALL= 67

VARIANCE 8

HISTOGRAM
```



COLOR PRINTOUTS

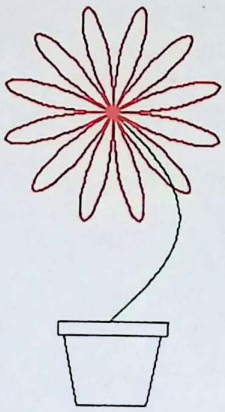
BIORHYTHM

(Refer to page 171.)



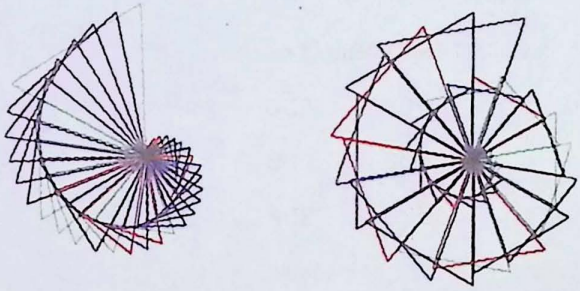
COMPUTER-DESIGNED FLOWER

(Refer to page 198.)



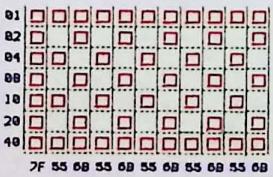
COMPUTER GRAPHICS

(Refer to page 200.)



DOT PATTERN DEVELOPMENT

(Refer to page 207.)



SHARP

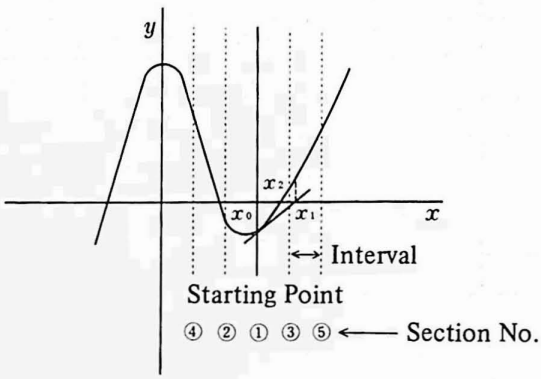
PROGRAM TITLE	ROOT OF AN EQUATION	PROGRAM NO. P5–A–1	1

[Outline] (Mathematics)

Finding the root of an equation is generally a time consuming task. Here is a method of root approximation using Newton's Method.
When a root is found, the starting point automatically varies with the designated interval according to Newton's Method. A quadratic equation has been chosen as an example:

[Operating Guide]

Input: Starting point
Minute value
Interval



Output: Root value (Press the [ENTER] key to find a root for the next interval.)

[Example]

$x^3 - 2x^2 - x + 2 = 0$ (Root = −1, 1, 2)
Calculation is made with the starting point being 0, the minute value being 10^{-4} and the interval being 0.5.
Write a function as a subroutine after line 500.
How to write a subroutine (in the above example):

1. Set the "PRO" mode by pressing the [MODE] key.
2. 500B = ((x − 2) * x − 1) * x + 2 [ENTER]
   510 RETURN [ENTER]

[Contents] (Formula)

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

When the absolute value of the difference between Xn and X_{n+1} becomes less than 10^{-8}, Xn is displayed as a root. The differential f'(x) is defined as follows:

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$ (h: minute value)

To vary 10^{-8}, change 1E–8 of the 340 line.

PROGRAM TITLE	ROOT OF AN EQUATION	PROGRAM NO. P5–A–1	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	STARTING POINT =_	
2	0 ENTER	MINUTE =_	
3	0.0001 ENTER	INTERVAL =_	
4	0.5 ENTER	ANS. = 2	
5	ENTER	ANS. = 1	Repeat ENTER to find next root.
6	ENTER	ANS. = –1	
	ENTER	ANS. = 1	
	ENTER	ANS. = –1	
	ENTER	ANS. = –1	
	ENTER	ANS. = –1	
	ENTER	ANS. = 2	
	⋮	⋮	

PROGRAM TITLE	ROOT OF AN EQUATION	PROGRAM NO. P5–A–1	3

[Program List]

```
 10:"A":INPUT "STA
    RTING POINT=";
    V
 20:INPUT "MINUTE=
    ";A
 30:INPUT "INTERVA
    L=";W
 40:G=V:F=V:Z=0
 50:IF Z=0GOTO 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)>=
    1E-8GOTO 300
350:BEEP 3:PRINT "
    ANS.=";C
360:RETURN
500:B=((X-2)*X-1)*
    X+2
510:RETURN
```

```
STATUS 1
                  300
```

[Memory Contents]

A	Minute Value (input value) = h
B	f(x)
C	x_o
D	f (x+h)
E	
F	√
G	√
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	Starting Point (input value)
W	Interval (input value)
X	x
Y	f (x)
Z	Initial Flag

SHARP

PROGRAM TITLE	MUTUAL CONVERSIONS BETWEEN RECTANGULAR COORDINATES AND POLAR COORDINATES	PROGRAM NO. P5– A–2	1

[Outline]

In this program, mutual conversions in two or three dimensions will be done. The dimension in inputs and outputs is in accordance with the preset.

[Operation Guide]

This program includes four functions as shown below;

- ○ two dimensions
  - Rectangular to Polar
  - Polar to Rectangular
- ○ three dimensions
  - Rectangular to Polar
  - Polar to Rectangular

[Example]

1. Two dimensions

a) Rectangular → Polar

$X = -1$, $Y = \sqrt{3}$ ⇨ $R = 2$, $\theta = 120°$

b) Polar → Rectangular

$R = 2$, $\theta = 120°$ ⇨ $X = -1$, $Y = 1.732$

2. Three dimensions

a) Rectangular → Polar

$X = -1$, $Y = 2$, $Z = -3$ ⇨ $R = 3.741657387$, $\theta = -53.30077479°$, $\varphi = 116.5650512$

b) Polar → Rectangular

$R = 3.741657387$, $\theta = -53.30077479°$, $\varphi = 116.5650512°$ ⇨ $X = -1$, $Y = 2$, $Z = -3$

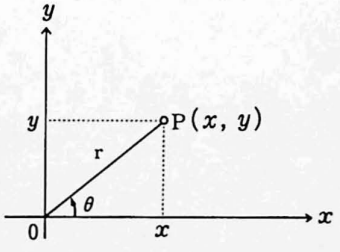
[Contents] (Formulas)

1. Two dimensions

a) Rectangular → Polar

When $x = y = 0$, then $r = 0$
therefore θ can't be defined.

$$\begin{cases} r = \sqrt{x^2 + y^2} \\ \text{When } y \geq 0, \text{ then } \theta = \text{Cos}^{-1}(x/r) \\ \text{When } y < 0, \text{ then } \theta = -\text{Cos}^{-1}(x/r) \end{cases}$$



b) Polar → Rectangular

$$\begin{cases} x = r \text{ Cos } \theta \\ y = r \text{ Sin } \theta \end{cases}$$

PROGRAM TITLE	MUTUAL CONVERSIONS BETWEEN RECTANGULAR COORDINATES AND POLAR COORDINATES	PROGRAM NO. P5– A–2	2

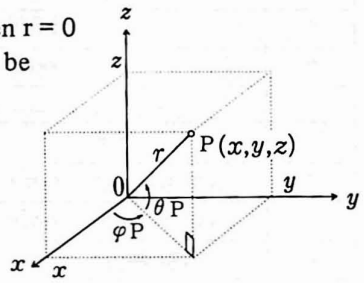
2. Three dimensions

a) Rectangular → Polar

$r = \sqrt{x^2+y^2+z^2}$

$\theta = \mathrm{Sin}^{-1}(z/r)$

When $x = y = z = 0$, then $r = 0$ therefore θ and φ can't be defined.



When $x>0$, then $\varphi = \mathrm{Tan}^{-1}\ (y/x)$
When $x = 0$ and $y \geq 0$, then $\varphi = 90°$.
When $x = 0$ and $y<0$, then $\varphi = -90°$
When $x<0$ and $y \geq 0$, then $\varphi = \mathrm{Tan}^{-1}\ (y/x) + 180°$
When $x<0$ and $y<0$, then $\varphi = \mathrm{Tan}^{-1}\ (y/x) - 180°$

b) Polar → Rectangular

$$\begin{cases} x = r\,\mathrm{Cos}\,\theta \cdot \mathrm{Cos}\,\varphi \\ y = r\,\mathrm{Cos}\,\theta \cdot \mathrm{Sin}\,\varphi \\ z = r\,\mathrm{Sin}\,\theta \end{cases}$$

DEF A ; two dimensional Rec. to Polar
DEF B ; two dimensional Polar to Rec.
DEF C ; three dimensional Rec. to Polar
DEF D ; three dimensional Polar to Rec.

[Key Operation Procedure]

* First, set to the degree mode.

Step No.	Input	Display	Remarks
1	DEF A	X = _	
2	–1 ENTER	Y = _	
3	√3 ENTER	R = 2	
4	ENTER	THETA = 120	
1	DEF B	R = _	
2	2 ENTER	THETA = _	
3	120 ENTER	X = –1.000	
4	ENTER	Y = 1.732	
1	DEF C	X = _	
2	–1 ENTER	Y = _	
3	2 ENTER	Z = _	

PROGRAM TITLE	MUTUAL CONVERSIONS BETWEEN RECTANGULAR COORDINATES AND POLAR COORDINATES	PROGRAM NO. P5– A–2	3

Step No.	Input	Display	Remarks
4	–3 [ENTER]	R = 3.741657387	
5	[ENTER]	THETA = –53.30077479	
6	[ENTER]	PHI = 116.5650512	
1	[DEF] [D]	R =_	
2	3.741657387 [ENTER]	THETA =_	
3	–53.30077479 [ENTER]	PHI = _	
4	116.5650512 [ENTER]	X = –1.000000001	
5	[ENTER]	Y = 2	
6	[ENTER]	Z = –3	

PROGRAM TITLE	MUTUAL CONVERSIONS BETWEEN RECTANGULAR COORDINATES AND POLAR COORDINATES	PROGRAM NO. P5– A–2	4

[Program List]

```
 10:"A":GOSUB 500
 20:R=√(X*X+Y*Y)
 30:IF R=0WAIT :
    USING :PRINT "
    R=0 ANGLE UNDE
    FINED":END
 40:GOSUB 700
 50:C=ACS (X/R)*A
 60:WAIT :USING :
    PRINT "R=";R
 62:PRINT "THETA="
    ;C
 64:END
 70:"B":GOSUB 600
 75:X=R*COS C:Y=R*
    SIN C
 80:USING :PRINT "
    X=";USING "###
    #####.###";X
 83:USING :PRINT "
    Y=";USING "###
    #####.###";Y
 85:END
 90:"C":GOSUB 500
100:INPUT "Z=";Z
110:R=√(X*X+Y*Y+Z*
    Z)
120:IF R=0GOTO 30
130:C=ASN (Z/R)
140:IF X>0LET F=
    ATN (Y/X):GOTO
    180
150:GOSUB 700
160:IF X=0LET F=A*
    ACS 0:GOTO 180
170:F=ATN (Y/X)+A*
    ACS -1
175:WAIT
180:USING :PRINT "
    R=";R
182:PRINT "THETA="
    ;C
184:PRINT "PHI=";F
190:END
200:"D":GOSUB 600
205:GOSUB 610
210:INPUT "PHI=";F
220:X=X*COS F:Y=Y*
    SIN F:Z=R*SIN
    C
230:WAIT :USING :
    PRINT "X=";X
232:PRINT "Y=";Y
234:PRINT "Z=";Z
240:END
500:INPUT "X=";X,"
    Y=";Y
510:USING :RETURN
600:INPUT "R=";R,"
    THETA=";C
605:RETURN
610:USING :X=R*COS
    C:Y=R*COS C:
    RETURN
700:A=(Y=0)+SGN Y:
    RETURN
```

```
STATUS 1
                    655
```

[Memory Contents]

A	√
B	
C	θ
D	
E	
F	φ
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	r
S	
T	
U	
V	
W	
X	x
Y	y
Z	z

SHARP

PROGRAM TITLE	FOURIER SERIES	PROGRAM NO. P5–A–3	1
		CE–150 required	

[Outline]

This program does Fourier expansion of a periodic function f(t) with $f(t+2\pi) = f(t)$.

[Operating Guide]

Input: 1). No. of divisions input

With the display of "N=", key-in the No. of divisions within the period.

2). Function value input

The input range is $[0, 2\pi]$ and with the display of "Y (i) =", key-in the function value $f\left(\frac{2\pi i}{N}\right)$

Output: Fourier coefficient output

The outputs of Fourier coefficient, ai (up to N/2) and bi (up to N/2-1) of function f (t) are possible.

Note that the No. of divisions N for input 1) must be even number and 176 maximum.

[Example]

Function values for n = 1 to 10 when one period $[0, 2\pi]$ of a composite wave forms $f(t) = \text{Cos } 2t + 3 \text{ Sin } t + 7 \text{ Sin } 3t$.

f (1) = 8.729771
f (2) = −2.070344
f (3) = −2.070341
f (4) = 8.729764
f (5) = 1
f (6) = −8.11173
f (7) = 0.45231
f (8) = 0.45231
f (9) = −8.111737
f (10) = 1

The Fourier expansion is thus performed.

[Contents] (Formulas)

$$f(t) = \frac{a_0}{2} + \sum_{i=1}^{\infty} (a_i \text{COS}_i t + b_i \text{SIN}_i t)$$

$$a_i = \frac{2}{N} \sum_{n=1}^{N} \cdot y_n \cdot \text{Cos}\left(\frac{2\pi}{N} \times n_j\right)$$

$$b_i = \frac{2}{N} \sum_{n=1}^{N} \cdot y_n \cdot \text{Sin}\left(\frac{2\pi}{N} \times n_j\right)$$

PROGRAM TITLE	FOURIER SERIES	PROGRAM NO. P5–A–3	2

[Printout]

```
A(0)=
          0.0000003
A(1)=
       -0.000000185
A(2)=
    9.999995194E-01
A(3)=
          4.9034E-07
A(4)=
          1.7992E-07
A(5)=
         -0.0000006
B(1)=
        3.000000328
B(2)=
    2.310925336E-06
B(3)=
        6.999998884
B(4)=
    2.219255066E-06
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	N =–	No. of divisions within a period
2	10 ENTER	Y (1) = ?	
3	8.729771 ENTER	Y (2) = ?	
4	–2.070344 ENTER	Y (3) = ?	
5	–2.070341 ENTER	Y (4) = ?	
6	8.729764 ENTER	Y (5) = ?	
7	1 ENTER	Y (6) = ?	
8	–8.11173 ENTER	Y (7) = ?	
9	0.45231 ENTER	Y (8) = ?	
10	0.45231 ENTER	Y (9) = ?	
11	–8.111737 ENTER	Y (10) = ?	
12	1 ENTER	>	Printout

PROGRAM TITLE	FOURIER SERIES	PROGRAM NO. P5-A-3	3

[Program List]

```
 10:"A":CLEAR :
    WAIT 0
 20:CLS :INPUT "N=
    ";N
 30:IF N/2<>INT (N
    /2)THEN 20
 40:DIM Y(N-1)
 50:FOR I=0TO N-1
 60:A$="Y("+STR$ (
    I+1)+")="
 70:PRINT A$;
 80:INPUT Y(I):CLS
 90:NEXT I
 95:RADIAN
100:A=0
110:FOR J=0TO N-1
120:A=A+Y(J):NEXT
    J
130:A=A/N:LPRINT "
    A(0)=",A
140:FOR I=1TO N/2
150:P=2*π*I/N:A=0
160:FOR J=1TO N
170:A=A+Y(J-1)*COS
    (P*J)
180:NEXT J
190:A=A*2/N
200:A$="A("+STR$ I
    +")="
210:LPRINT A$,A
220:NEXT I
230:FOR I=1TO N/2-
    1
240:P=2*π*I/N:B=0
250:FOR J=1TO N
260:B=B+Y(J-1)*SIN
    (P*J)
270:NEXT J
280:B=B*2/N
290:B$="B("+STR$ I
    +")"
300:LPRINT B$,B
310:NEXT I
320:END
```

STATUS 1

443

[Memory Contents]

A	Fourier coefficient (a_o to $a_{n/2}$)
B	Fourier coefficient (b_1 to $b_{n/2-1}$)
C	
D	
E	
F	
G	
H	
I	√
J	√
K	
L	
M	
N	No. of divisions
O	
P	2πI/N
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
A$	Input message
B$	Output message
Y(N)	Input data (Function value)

SHARP

PROGRAM TITLE	LAGRANGE'S INTERPOLATION	PROGRAM NO. P5–A–4	1
		CE-150 required	

[Outline]

This program performs interpolation by using Lagrange's interpolation polynomial to calculate the Yvalue for the X value to be interpolated.

[Operating Guide]

Input
1. Number of coordinates (N) ($N \leq 61$)
2. Coordinates input
   Key-in coordinates X (i) and Y (i). ($1 \leq i \leq N$)
3. After "Z =" has been displayed, key-in the x-coordinte to interpolate.

Output
4. Interpolated value
   "X=": keyed-in x-coordinate to interpolate (=Z)
   "P =": Interpolated value (y-axis)

The above 3 and 4 can be executed repeatedly.

[Example]

Number of coordinates: 4

Coordinates: (5,3)
(8,9)
(12,4)
(6,1)

Values to be interpolated: 7

[Contents] (Formulas)

To make interpolation, using Lagrange's interpolation polynomial, determine the value required for interpolation.

Assuming the number of coordinates is n, determine a polynomial with degree n – 1

$$P_{n-1}(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x^1 + a_0$$

Since $P_{n-1}(x) = y_1 b_1(x) + y_2 b_2(x) + \cdots + y_n b_n(x)$

For $k = 1.2 \cdots n$,

$$b_k(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$$

$$= \prod_{\substack{i=1 \\ j \neq k}}^{n} \frac{(x-x_j)}{(x_k-x_j)}$$

This yields the required interpolation value.

PROGRAM TITLE	LAGRANGE'S INTERPOLATION	PROGRAM NO. P5–A–4	2

[Printout]

```
X= 7
P= 3.821428571
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	N = _	Number of coordinates (MAX. 61)
2	4 ENTER	X (1) = ?	
3	5 ENTER	Y (1) = ?	
4	3 ENTER	X (2) = ?	
5	8 ENTER	Y (2) = ?	
6	9 ENTER	X (3) = ?	
7	12 ENTER	Y (3) = ?	
8	4 ENTER	X (4) = ?	
9	6 ENTER	Y (4) = ?	
10	1 ENTER	Z = _	Execution is completed by pressing only ENTER
11	7 ENTER	Z = _	Key operation returns back to Step 10.
12	ENTER	>	

PROGRAM TITLE	LAGRANGE'S INTERPOLATION	PROGRAM NO. P5–A–4	3

[Program List]

```
 10:"A":CLEAR :
    WAIT 0
 20:INPUT "N=";N
 25:N=N-1:DIM X(N)
    ,Y(N),B(N)
 30:FOR I=0TO N
 35:A$="X("+STR$ (
    I+1)+")="
 36:PRINT A$;
 40:INPUT X(I):
    GOTO 42
 41:N=I:GOTO 55
 42:A$="Y("+STR$ (
    I+1)+")="
 43:CLS
 45:PRINT A$;
 46:INPUT Y(I)
 47:CLS
 50:NEXT I
 55:CLS :INPUT "Z=
    ";Z:GOTO 60
 56:END
 60:P=0:FOR K=0TO
    N
 70:B(K)=1
 80:FOR J=0TO N
 90:IF J=KTHEN 110
100:B(K)=B(K)*(Z-X
    (J))/(X(K)-X(J
    ))
110:NEXT J
120:P=P+B(K)*Y(K)
130:NEXT K
140:LPRINT "X=";Z
150:LPRINT "P=";P
160:GOTO 55

STATUS 1
                362
```

[Memory Contents]

A	
B	
C	
D	
E	
F	
G	
H	
I	√
J	√
K	√
L	
M	
N	Number of data
O	
P	Value to be determined by interpolating Z.
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	Interpolated value
A$	Input message
B(N)	Operation area for the interpolation
X(N)	Input data to X-axis
Y(N)	Input data to Y-axis

SHARP

PROGRAM TITLE	QUADRATIC AND CUBIC EQUATIONS	PROGRAM NO. P5– A–6	1

CE–150 required

[Outline]

This program determines the roots of quadratic and cubic equations. Selecting a quadratic or cubic equation, and keying-in the factors of the equation, allows you to find its root.

[Operating Guide]

Input: 1. Choosing the equation

[DEF] [A] for the root of a quadratic equation ..... (A)

[DEF] [B] for the root of a cubic equation ..... (B)

2. Coefficients input

For (A), Coefficients a, b and c inputs

For (B), Coefficients a, b, c and d inputs

Output: Root value – "REAL", "X_1" and "X_2" will be printed out for 2 real roots.

"DOUBLE" and "X_1" will be printed out for a double root.

"*** REAL ***", IMAGINARY, real part and imaginary part will be printed out for an imaginary root.

[Example]

1. Root of a quadratic equation

$4X^2 - X - 1 = 0$

$5X^2 + 4X + 1 = 0$

2. Root of a cubic equation

$X^3 + X^2 - 2X - 2 = 0$

[Contents] (Formulas)

I) Root of a quadratic equation:

$ax^2 + bx + c = 0 \quad (a \neq 0)$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(1) Real root with $b^2 - 4ac > 0$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(2) Real root with $b^2 - 4ac = 0$

$$x = -\frac{b}{2a}$$

(3) Imaginary root with $b^2 - 4ac < 0$

Real part: $\frac{-b}{2a}$

Imaginary part: $\frac{\sqrt{4ac-b^2}}{2a}$

II) Root of a cubic equation:

Cardano's method is used for a solution.

$AX^3 + BX^2 + CX + D = 0$ ($A \neq 0$ and all factors are real numbers.)

The following is obtained by dividing factors by A:

$x^3 + ax^2 + bx + c = 0$

Through conversion of $y = x - \frac{a}{3}$,

$y^3 + 3py + q = 0$

$p = \frac{b}{3} - \frac{a^2}{9} \qquad q = c - \frac{ab}{3} + \frac{2a^3}{27}$

Let $Y = u + v$, then the following is obtained:

$u^3 + v^3 + 3uv(u+v) + 3P(u+v) + q = 0$

Let $u^3 + p^3 = -q$, then $uv = -p$.

$v^3 + v^3 = -q$

$u^3 v^3 = -P^3$

This shows that u^3 and v^3 are the roots of a quadratic equation of $t^2 + qt - p^3 = 0$. That is to say,

$u^3 = \frac{1}{2}(-q + \sqrt{q^2 + 4p^3})$

$v^3 = \frac{1}{2}(-q - \sqrt{q^2 + 4p^3})$

From this, the roots α, β and γ of $y^3 + 3py + q = 0$ become as follows:

$\alpha = u + v$

$\beta = -\frac{1}{2}(u+v) + \frac{\sqrt{3}}{2} i (u - v)$

$\gamma = -\frac{1}{2}(u+v) - \frac{\sqrt{3}}{2} i (u - v)$

The above are to be divided into the real and imaginary parts.

(1) When $q^2 + 4p^3 > 0$, u^3 and v^3 are real numbers.
Therefore, u and v are the real cubic root of u^3 and v^3, and the above formula can be used as it is. This is the case for one real root and two imaginary roots.

(2) When $q^2 + 4p^3 < 0$, u^3 and v^3 are imaginary roots.
Let $u^3 = re^{i\theta}$ then $v^3 = re^{-i\theta}$,
$r = -P^3$

$\theta = \tan^{-1} \dfrac{\sqrt{-q^2-4p^3}}{-q}$ therefore,

$u = \sqrt{-P}\left(\cos\dfrac{\theta}{3} + i\sin\dfrac{\theta}{3}\right)$

$v = \sqrt{-P}\left(\cos\dfrac{\theta}{3} + i\sin\dfrac{\theta}{3}\right)$

Through this, roots α, β and γ of $Y^3 + 3PY + q = 0$

are as follows: $\alpha = -2\sqrt{P}\sin\left(\dfrac{\pi}{2} - \dfrac{\theta}{3}\right)$

$\beta = -2\sqrt{-P}\sin\left(\dfrac{\pi}{6} + \dfrac{\theta}{3}\right)$

$\gamma = -2\sqrt{-P}\sin\left(\dfrac{\pi}{6} - \dfrac{\theta}{3}\right)$

This is the case of three different real roots.

(3) When $p^2 + 4p^3 = 0$ and $p \neq 0$, $u^3 = v^3 = -\dfrac{q}{2}$ is obtained.

Therefore, from $U = V = \sqrt[3]{-\dfrac{q}{2}}$

$\alpha = 2u$
$\beta = \gamma = -u$
This is the case of a double root and a another root.

(4) If $q^2 + 4p^3 = 0$ and $p = 0$, $q = 0$ is obtained. Therefore,
since $u = v = 0$, the result is;
$\alpha = \beta = \gamma = 0$

This is a triple root.
Adding $\dfrac{a}{3}$ to α, β and γ finds solutions to the equation.

Be noted that when 2 or 3 different roots are very close to each other, they may be regarded as a double or triple root and vice versa.

[Printout]

```
 REAL
    6.403882032E-01
 REAL
   -3.903882032E-01
 ***REAL***
                -0.4
 IMAGINARY
                 0.2
 REAL ROOT
       1.414213562
      -1.414213562
  -9.999999995E-01
```

PROGRAM TITLE	QUADRATIC AND CUBIC EQUATIONS	PROGRAM NO. P5– A–6	4

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	A = _	
2	4 ENTER	B = _	
3	–1 ENTER	C = _	
4	–1 ENTER	>	Printout

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	A = _	
2	5 ENTER	B = _	
3	4 ENTER	C = _	
4	1 ENTER	>	Printout

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF B	A = _	
2	1 ENTER	B = _	
3	1 ENTER	C = _	
4	–2 ENTER	D = _	
5	–2 ENTER	>	Printout

PROGRAM TITLE	QUADRATIC AND CUBIC EQUATIONS	PROGRAM NO. P5– A–6	5

[Program List]

```
 10:"A":INPUT "A="
    ;A,"B=";B,"C="
    ;C
 20:B=-B/2/A:D=B*B
    -C/A
 30:IF D=0GOTO 90
 40:IF D>0GOTO 110
 50:Y=√(-D)
 60:LPRINT "***REA
    L***",B
 70:LPRINT "IMAGIN
    ARY",Y
 80:END
 90:LPRINT "DOUBLE
    ",B
100:END
110:LPRINT "REAL",
    B+√D
120:LPRINT "REAL",
    B-√D
130:END
210:"B":INPUT "A="
    ;D,"B=";F,"C="
    ;G,"D=";H
220:F=F/D:G=G/D:H=
    H/D
240:F=F/3
250:D=G/3-F*F
260:E=H-F*G+2*F*F*
    F
270:C=4*D*D*D+E*E
280:IF 10^(-8)>ABS
    CGOTO 470
290:IF C>0GOTO 400
300:A=2*√(-D)
310:B=ACS (E/(2*D*
    √(-D)))/3
320:D=ASN 1:E=ASN
    .5
330:G=A*SIN (D-B):
    H=-A*SIN (E+B)
340:I=-A*SIN (E-B)
350:G=G-F:H=H-F:I=
    I-F
370:BEEP 3:LPRINT
    "REAL ROOT",G
380:LPRINT H,I
390:END
400:C=√C:A=.5*(C-E
    ):B=-.5*(C+E):
    C=1/3
410:A=ABS A^C*SGN
    A
420:B=ABS B^C*SGN
    B:C=.5*√3
430:BEEP 3:LPRINT
    "*REAL ROOT*",
    A+B-F
440:LPRINT "*REAL*
    ",-.5*(A+B)-F
450:LPRINT "IMAGIN
    ARY",C*ABS (A-
    B)
460:END
470:BEEP 3:IF 10^(
    -8)>ABS D
    LPRINT "TRIPLE
     ROOT",-F:END
480:A=-ABS (.5*E)^
    (1/3)*SGN E
490:LPRINT "REAL R
    OOT",2*A-F
495:LPRINT "DOUBLE
     ROOT",-A-F
500:END
```

```
STATUS 1
                  790
```

[Memory Contents]

A	a	√
B	b − b/(2a)	√
C	c	√
D	d	a
E		√
F		b
G		c
H		d
I		
J		
K		
L		
M		
N		
O		
P		
Q		
R		
S		
T		
U		
V		
W		
X		
Y	√	
Z		

SHARP

PROGRAM TITLE	FIRST ORDER DIFFERENTIAL EQUATION	PROGRAM NO. P5–A–7	1

CE–150 required

[Outline]

This program solves a first order differential equation by using the Runge-Kutta-Gill method.

[Operating Guide]

< Input >		< Output >	< Key Operation >
Initial conditions	x_0	x_0	ENTER key is used for the x value progression.
	y_0	y_0	
		h	
x value increment	h	$x = x_1, x_2, ---$	
Solution interval	T	y value for x	

Write the equation as a subroutine on line 500.
In PRO mode, modify the 500 line equation as required.

Note: Except for $x = nh+x_0$ (n=0, 1, 2, ---) a proportional allocation is made for the y value between $x_0+(n-1)h$ and x_0+nh.

[Example]

1. Equation $y' = -xy$ is solved under the initial condition of $x_0 = 0$, provided $y_0 = 10$.
   However, assuming $h = 0.01$, $T = 0.03$, y is obtained with x = 0.03, 0.06 and so on.

[Contents] (Formulas)

Assume that the equation is $y' = f(x, y)$, with its initial condition of (x_0, y_0). With the x value taken in h increments, sequentially determine y_n of the y value in $x_n = x_0 + nh$ (n=1, 2, ---).
The formulas for determining x_{n+1} and y_{n+1} from x_n and y_n are written as follows, according to the Runge-Kutta-Gill method.

$$k_0 = hf(x_n, y_n) \qquad r_1 = (1/2)(k_0 - 2q_0)$$

$$y^{(1)} = y_n + r_1, \qquad q_1 = q_0 + 3r_1 - (1/2)k_0, \qquad k_1 = hf(x_n + h/2, y^{(1)})$$

$$r_2 = (1 - \sqrt{1/2})(k_1 - q_1),$$

$$y^{(2)} = y^{(1)} + r_2, \qquad q_2 = q_1 + 3r_2 - (1 - \sqrt{1/2})k_1, \quad k_2 = hf(x_n + h/2, y^{(2)})$$

$$r_3 = (1 + \sqrt{1/2})(k_2 - q_2)$$

$$y^{(3)} = y^{(2)} + r_3, \qquad q_3 = q_2 + 3r_3 - (1 + \sqrt{1/2})k_2, \quad k_3 = hf(x_n + 1, y^{(3)})$$

$$r_4 = (1/6)(k_3 - 2q_3)$$

$$y_{n+1} = y^{(3)} + r_4, \qquad q_4 = q_3 + 3r_4 - (1/2)k_3$$

Thus y_{n+1} has been determined from y_n. Here, n = 0, 1, 2, -----
The value of q_0 is 0 (zero) at the start point x_0, and q_4 is thereafter taken as a new q_0.

PROGRAM TITLE	FIRST ORDER DIFFERENTIAL EQUATION	PROGRAM NO. P5– A–7	2

[Printout]

```
X= 0.03
Y= 9.995501013
X= 0.06
Y= 9.982016191
X= 0.09
Y= 9.959581904
X= 0.12
Y= 9.928258582
X= 0.15
Y= 9.888130449
X= 0.18
Y= 9.839305144
X= 0.21
Y= 9.781913245
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	X0 =–	
2	0 ENTER	Y0 =–	
3	10 ENTER	H =–	
4	0.01 ENTER	T =–	
5	0.03 ENTER	0.03 9.995501013	
6	ENTER	0.06 9.982016191	
⋮	⋮	⋮	

PROGRAM TITLE	FIRST ORDER DIFFERENTIAL EQUATION	PROGRAM NO. P5–A–7	3

[Program List]

```
 10:"A":INPUT "X0=
    ";X,"Y0=";Y,"H
    =";H,"T=";T
 20:A=1+√.5:B=1-√.
    5:USING :Q=0
 25:Z=X+T:S=X
 30:GOSUB 500
 40:K=H*F:R=(K-2*Q
    )/2:Y=Y+R
 50:Q=Q+3*R-K/2
 60:X=X+H/2:GOSUB
    500
 70:K=H*F:R=B*(K-Q
    ):Y=Y+R
 80:Q=Q+3*R-B*K
 90:GOSUB 500
100:K=H*F:P=A*(K-Q
    ):Y=Y+R
110:Q=Q+3*R-A*K
120:X=X+H/2:GOSUB
    500
130:K=H*F:R=(K-2*Q
    )/6:Y=Y+R
140:Q=Q+3*R-K/2
150:IF X<ZLET S=X:
    Y1=Y:GOTO 30
160:IF X=ZGOTO 200
170:Y2=(Z-S)*(Y-Y1
    )/H+Y1
180:BEEP 3:LPRINT
    "X=";Z
190:LPRINT "Y=";Y2
    :GOTO 210
200:BEEP 3:LPRINT
    "X=";X
205:LPRINT "Y=";Y
210:Z=Z+T:S=X:Y1=Y
215:WAIT :PRINT X,
    Y
220:GOTO 30
500:F=-X*Y
510:RETURN
```

STATUS 1

474

[Memory Contents]

A	$1+\sqrt{1/2}$
B	$1-\sqrt{1/2}$
C	
D	
E	
F	$f(x, y)$
G	
H	h
I	
J	
K	√
L	
M	
N	
O	
P	
Q	q_n
R	r_n
S	x_{n-1}
T	Interval of solutions
U	
V	
W	
X	x_n
Y	y_n
Z	√
Y1	y_{n-1}
Y2	y_{nT}

SHARP

PROGRAM TITLE	DETERMINANT	PROGRAM NO. P5– A–10	1

CE–150 required

[Outline]

Based on the sweeping-out method, this program calculates the determinant of a matrix with n orders.

Processing includes:

1. Data input
2. Data verification and correction
3. Output of input data and calculation results after program execution
4. Output of calculation results only, after program execution

[Operating Guide]

Processing selection

[DEF] [A] : Data input of elements of the matrix.
[DEF] [B] : Data verification and correction of data.
[DEF] [C] : Output of input data and calculation results (Execution of the determinant.)
[DEF] [D] : Output of calculation results only.

The [DEF] [C] prints out the input data. The order is possible up to 12.

[Example]

$$\begin{bmatrix} 4 & 7 & 1 & 8 \\ 5 & -1 & 2 & -4 \\ 3 & 12 & -5 & 6 \\ 1 & 4 & 7 & 2 \end{bmatrix} = -3276$$

[Contents] (Formulas)

This program converts the matrix into a triangular matrix by using the sweeping-out method, then gets the answer.

Assume that a matrix is $\{a_{ij}\}$ $(i, j = 1 \sim n)$

$P = a_{mm}$ $(m = 2 \sim n)$

$q = a_{im}/P$ $(i = 1 \sim m-1)$

$a_{ij} = a_{ij} - q \cdot a_{mj}$ $(j = 1 \sim m)$

On calculation, the following is obtained:

$a_{ij} = 0$ *for* $i < j$

This results in: $\det = a_{11} \cdot a_{22} \cdot a_{33} \cdot \cdots \cdot a_{nn}$

With P = 0, however, during computation, resulting in an error, since it's impossible to calculate.

PROGRAM TITLE	DETERMINANT	PROGRAM NO. P5–A–10	2

[Printout]

```
A(1,1)= 4
A(1,2)= 7
A(1,3)= 1
A(1,4)= 8
A(2,1)= 5
A(2,2)=-1
A(2,3)= 2
A(2,4)=-4
A(3,1)= 3
A(3,2)= 12
A(3,3)=-5
A(3,4)= 6
A(4,1)= 1
A(4,2)= 4
A(4,3)= 7
A(4,4)= 2
det=-3276
```

[Key Operation Procedure] : Data input

Step No.	Input	Display	Remarks
1	DEF A	N = _	No. of the order input
2	4 ENTER	A(1, 1) = ?	
3	4 ENTER	A(1, 2) = ?	
4	7 ENTER	A(1, 3) = ?	
5	1 ENTER	A(1, 4) = ?	
6	8 ENTER	A(2, 1) = ?	
7	5 ENTER	A(2, 2) = ?	
8	–1 ENTER	A(2, 3) = ?	
9	2 ENTER	A(2, 4) = ?	
10	–4 ENTER	A(3, 1) = ?	
11	3 ENTER	A(3, 2) = ?	
12	2 ENTER	A(3, 3) = ?	Incorrect data input
13	–5 ENTER	A(3, 4) = ?	
14	6 ENTER	A(4, 1) = ?	
15	1 ENTER	A(4, 2) = ?	
16	4 ENTER	A(4, 3) = ?	
17	7 ENTER	A(4, 4) = ?	
18	2 ENTER	>	

PROGRAM TITLE	DETERMINANT	PROGRAM NO. P5–A–10	3

[Key Operation Procedure] : Data confirmation and correction

Step No.	Input	Display	Remarks
19	DEF B	A(1, 1) = 4 ?	
20	ENTER	A(1, 2) = 7 ?	
21	ENTER	A(1, 3) = 1 ?	
22	ENTER	A(1, 4) = 8 ?	
23	ENTER	A(2, 1) = 5 ?	
24	ENTER	A(2, 2) = –1 ?	
25	ENTER	A(2, 3) = 2 ?	
26	ENTER	A(2, 4) = –4 ?	
27	ENTER	A(3, 1) = 3 ?	
28	ENTER	A(3, 2) = 2 ?	
29	12 ENTER	A(3, 3) = –5 ?	Correct data input
30	ENTER	A(3, 4) = 6 ?	
31	ENTER	A(4, 1) = 1 ?	
32	ENTER	A(4, 2) = 4 ?	
33	ENTER	A(4, 3) = 7 ?	
34	ENTER	A(4, 4) = 2 ?	
35	ENTER	>	
36	DEF C	>	Output of input data and result.

PROGRAM TITLE	DETERMINANT	PROGRAM NO. P5– A–10	4

[Program List]

```
 10:"A":CLEAR :
    WAIT 0
 20:CLS :INPUT "N=
    ";N:N=N-1
 30:DIM A(N,N)
 40:FOR I=0TO N
 50:FOR J=0TO N
 60:A$="A("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
 65:PRINT A$;
 70:INPUT A(I,J):
    CLS
 80:NEXT J
120:NEXT I:END
130:"B":FOR I=0TO
    N
140:FOR J=0TO N
150:A$="A("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
160:CLS :PRINT A$;
    A(I,J);
165:CURSOR 15
170:INPUT E:A(I,J)
    =E
180:NEXT J
200:NEXT I:END
210:"C":GOSUB 500
215:"D":FOR M=NTO
    1STEP -1
220:P=A(M,M)
225:IF P=0THEN 900
230:FOR I=0TO M-1
240:Q=A(I,M)/P
250:FOR J=0TO M
260:A(I,J)=A(I,J)-
    Q*A(M,J)
270:NEXT J:NEXT I:
    NEXT M
275:D=A(0,0)
280:FOR I=1TO N
290:D=D*A(I,I)
300:NEXT I
310:BEEP 1:LPRINT
    "det=";D:END
500:FOR I=0TO N
510:FOR J=0TO N
520:A$="A("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
530:LPRINT A$;A(I,
    J)
540:NEXT J:NEXT I:
    RETURN
900:LPRINT "ERROR"
    :END


STATUS 1     584
```

[Memory Contents]

A	
B	
C	
D	Determinant Value
E	Correction Data
F	
G	
H	
I	√
J	√
K	
L	
M	√
N	Number of the order
O	
P	√
Q	√
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
A$	Input/Output message
A(N,N)	Input data

SHARP

PROGRAM TITLE	INVERSE MATRIX	PROGRAM NO. P5–A–11	1
		CE–150 required	

[Outline]

This program determines the inverse matrix of a given n order matrix according to the sweeping-out method.

Processing is divided into the following:

1. Data input
2. Data verification and correction
3. Execution

[Operating Guide]

Input: Processing selection

- DEF A : Data input (Input of n order matrix elements)
- DEF B : Data verification and correction (Verification and correction of n order matrix elements)
- DEF C : Execution (Inverse matrix determination)

Output: Output of the entered matrix elements. The output appears after a beep tone.
The order is possible up to 11.

[Example]

$$\begin{bmatrix} 1 & -2 & 0 \\ -1 & 3 & 2 \\ 1 & -1 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} 7 & 4 & -2 \\ 3 & 2 & -1 \\ -1 & -0.5 & 0.5 \end{bmatrix}$$

[Contents] (Formulas)

Assume that a matrix is $A = \{a_{ij}\}$ $(i, j = 1 \sim n)$

$a_{ij} = a_{ij} + 1$ $(i = 1 \sim n)$

$P = a_{mm} - 1$ $(m = 1 \sim n)$

$a_{mj} = a_{mj}/P$ $(j = 1 \sim n)$

$a_{ij} = a_{ij} - a_{im} a_{mj}$ $(i = 1 \sim n,\ i \cdot m)$

$a_{ii} = a_{ii} - 1$ $(i = 1 \sim n)$

After computation, (a_{ij}) turns out to be the inverse matrix of the original matrix. With P=0 during calculation, however, no computation is possible, resulting in an error.

[Printout]

```
A(1, 1)= 1
A(1, 2)=-2
A(1, 3)= 0
A(2, 1)=-1
A(2, 2)= 3
A(2, 3)= 2
A(3, 1)= 1
A(3, 2)=-1
A(3, 3)= 4
C(1, 1)= 7
C(1, 2)= 4
C(1, 3)=-2
C(2, 1)= 3
C(2, 2)= 2
C(2, 3)=-1
C(3, 1)=-1
C(3, 2)=-0.5
C(3, 3)= 0.5
```

PROGRAM TITLE	INVERSE MATRIX	PROGRAM NO. P5– A–11	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	N = _	No. of the order input
2	3 ENTER	A(1, 1) = ?	Data input
3	1 ENTER	A(1, 2) = ?	
4	–2 ENTER	A(1, 3) = ?	
5	2 ENTER	A(2, 1) = ?	Wrong data
6	–1 ENTER	A(2, 2) = ?	
7	3 ENTER	A(2, 3) = ?	
8	2 ENTER	A(3, 1) = ?	
9	1 ENTER	A(3, 2) = ?	
10	–1 ENTER	A(3, 3) = ?	
11	4 ENTER	>	
1	DEF B	A(1, 1) = 1 ?	
2	ENTER	A(1, 2) = –2 ?	
3	ENTER	A(1, 3) = 2 ?	
4	0 ENTER	A(2, 1) = –1 ?	Correct data input
5	ENTER	A(2, 2) = 3 ?	
6	ENTER	A(2, 3) = 2 ?	
7	ENTER	A(3, 1) = 1 ?	
8	ENTER	A(3, 2) = –1 ?	
9	ENTER	A(3, 3) = 4 ?	
10	ENTER	>	
11	DEF C	>	Printout

PROGRAM TITLE	INVERSE MATRIX	PROGRAM NO. P5–A–11	3

[Program List]

```
 10:"A":CLEAR :
    WAIT 0
 20:CLS :INPUT "N=
    ";N:N=N-1
 30:DIM A(N,N)
 40:FOR I=0TO N
 50:FOR J=0TO N
 60:A$="A("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
 65:PRINT A$;
 70:INPUT A(I,J):
    CLS
 80:NEXT J
120:NEXT I:END
130:"B":FOR I=0TO
    N
140:FOR J=0TO N
150:A$="A("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
160:CLS :PRINT A$;
    A(I,J);
165:CURSOR 15
170:INPUT E:A(I,J)
    =E
180:NEXT J
200:NEXT I:END
210:"C":GOSUB 500:
    FOR I=0TO N
220:A(I,I)=A(I,I)+
    1:NEXT I
230:FOR M=0TO N
240:P=A(M,M)-1
245:IF P=0THEN 900
250:FOR J=0TO N
260:A(M,J)=A(M,J)/
    P:NEXT J
265:FOR I=0TO N
270:IF I=MTHEN 290
275:Q=A(I,M)
277:FOR J=0TO N
280:A(I,J)=A(I,J)-
    Q*A(M,J)
285:NEXT J
290:NEXT I:NEXT M
295:FOR I=0TO N
300:A(I,I)=A(I,I)-
    1
310:NEXT I
320:GOSUB 550:END
500:FOR I=0TO N
510:FOR J=0TO N
520:A$="A("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
530:LPRINT A$;A(I,
    J)
540:NEXT J:NEXT I:
    RETURN
550:BEEP 1:FOR I=0
    TO N
560:FOR J=0TO N
570:A$="C("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
580:LPRINT A$;A(I,
    J)
590:NEXT J:NEXT I:
    RETURN
900:LPRINT "ERROR"
    :END

STATUS 1
                  735
```

[Memory Contents]

A	
B	
C	
D	
E	Correction data
F	
G	
H	
I	
J	√
K	√
L	
M	√
N	No. of the order
O	
P	√
Q	√
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
A$	Input message
A(N,N)	Input data

SHARP

PROGRAM TITLE	MATRIX PRODUCT	PROGRAM NO. P5–A–12	1

CE–150 required

[Outline]

With this program, you can determine matrix product C of mℓ type matrix A and ℓn type matrix B.

[Operating Guide]

Input: 1. Inputs of No. of rows m and No. of columns ℓ for matrix A. Key-in the elements of matrix A.
2. Input of No. of columns n in for matrix B. Key-in the elements of matrix B.

The limits of ℓ, m and n are $\ell(m+n) \leqq 170$.

Output: The outputs of elements of product matrix C.

[Example]

mℓ type matrix A ℓn type matrix B Product (mn type matrix C)

$$\begin{bmatrix} 4 & 0 & -1 \\ -3 & 3 & 7 \\ -9 & 2 & 5 \\ 5 & -1 & 3 \end{bmatrix} \cdot \begin{bmatrix} -1 & 5 \\ -6 & -6 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} -5 & 16 \\ -8 & -5 \\ 2 & -37 \\ 4 & 43 \end{bmatrix}$$

[Contents] (Formulas)

The following calculation is carried out.

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

$$m \left\{ \underbrace{\begin{bmatrix} a_{11} & a_{21}\cdots a_{1,\ell} \\ a_{21} & a_{22}\cdots a_{2,\ell} \\ \vdots & \vdots \\ a_{m_1} & a_{m_2}\cdots a_{m\ell} \end{bmatrix}}_{\ell} \right. \underbrace{\begin{bmatrix} b_{11} & b_{12}\cdots b_{1n} \\ b_{21} & b_{22}\cdots b_{2n} \\ \vdots & \vdots \\ b_{\ell_1} & b_{\ell_2}\cdots b_{\ell n} \end{bmatrix}}_{n} = \begin{bmatrix} c_{11} & c_{12}\cdots c_{1n} \\ c_{21} & c_{22}\cdots c_{2n} \\ \vdots & \vdots \\ c_{m_1} & c_{m_2}\cdots c_{mn} \end{bmatrix}$$

PROGRAM TITLE	MATRIX PRODUCT	PROGRAM NO. P5–A–12	2

[Printout]

```
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
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	M = ? L =	No. of rows input for mℓ type matrix A
2	4 ENTER	M = 4 L = ?	No. of columns input for mℓ type matrix A
3	3 ENTER	a(1, 1) = ?	Elements input for matrix A
4	4 ENTER	a(1, 2) = ?	
5	0 ENTER	a(1, 3) = ?	
6	–1 ENTER	a(2, 1) = ?	
7	–3 ENTER	a(2, 2) = ?	
⋮	⋮	⋮	⋮
15	3 ENTER	L = 3, N = ?	No. of columns input for ℓn type matrix B
16	2 ENTER	b(1, 1) = ?	Elements input for matrix B
17	–1 ENTER	b(1, 2) = ?	
18	5 ENTER	b(2, 1) = ?	
19	–6 ENTER	b(2, 2) = ?	
20	–6 ENTER	b(3, 1) = ?	
21	1 ENTER	b(3, 2) = ?	
22	4 ENTER	>	Printout

PROGRAM TITLE	MATRIX PRODUCT	PROGRAM NO. P5– A–12	3

[Program List]

```
 10:"A":CLEAR :
    WAIT 0:CLS
 20:PRINT "M=
    L="
 22:CURSOR 3:INPUT
    M:CURSOR 10:
    INPUT L
 23:M=M-1:L=L-1
 25:DIM A(M,L)
 30:FOR I=0TO M:
    FOR J=0TO L
 40:A$="a("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
 50:CLS :PRINT A$;
 60:INPUT A(I,J)
 70:NEXT J:NEXT I
 80:CLS :PRINT "L=
    ";L+1:CURSOR 8
 90:INPUT "N=";N:N
    =N-1
100:DIM C(M,N)
110:FOR I=0TO L
120:FOR J=0TO N
130:A$="b("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
140:CLS :PRINT A$;
150:INPUT B
160:FOR K=0TO M
170:C(K,J)=C(K,J)+
    A(K,I)*B
180:NEXT K:NEXT J:
    NEXT I
190:FOR I=0TO M:
    FOR J=0TO N
200:A$="c("+STR$ (
    I+1)+","+STR$
    (J+1)+")="
210:LPRINT A$;C(I,
    J)
220:NEXT J:NEXT I
230:END
```

```
STATUS 1
                446
```

[Memory Contents]

A	
B	Elements of matrix B (Input data)
C	
D	
E	
F	
G	
H	
I	√
J	√
K	√
L	Columns for matrix A/Rows for matrix B
M	Rows of matrix A
N	Columns of matrix B
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
A$	Input message
A(M,L)	Elements of matrix A (Input data)
C(M,N)	Elements of product matrix

SHARP

PROGRAM TITLE	CORRELATION COEFFICIENT, LINEAR REGRESSION AND PLOT	PROGRAM NO. P5–B–1	1

CE–150 required

[Outline] (Statistics)

Data exists for analyses and estimations.

This program calculates the covariance, correlation coefficient, and linear regression coefficients between related datas $(X_1, Y_1) \ldots (X_n, Y_n)$. The given data is estimated for application to Y = AX+B, with a graphic printout of the results.

[Operating Guide]

1. Data input (X_i, Y_i) (where the capacity is $i \leq 10$, in the standard memory size.)
2. The covariance, correlation coefficient, linear regression coefficient and mean value are calculated for printouts.
3. The graph with $\overline{X}$ and $\overline{Y}$ centered on the X-axis and Y-axis is generated, on which the input data and estimated values are displayed in different colors.
4. The estimated value Y is determined from the value X for the printout of the X and Y values.

[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

Covariance = −3.060714286

Correlation coefficient = −9.693968513 E-01

Linear regression coefficient
a = −3.942042318
b = 39.4475621

Mean value X = 7.375, Y = 10.375
Estimated value
X = 7 , Y = 11.85326587
X = 8 , Y = 7.911223556
X = 7.5 , Y = 9.882244715
X = 7.3 , Y = 10.67065318
X = 7.4 , Y = 10.27644895

[Contents] (Formulas)

$Sxx = \Sigma x_i^2 - n\bar{x}^2$

$Sxy = \Sigma x_i y_i - n\bar{x}\bar{y}$

$Syy = \Sigma y_i^2 - n\bar{y}^2$

$C = Sxy/(n-1)$ ····· Covariance

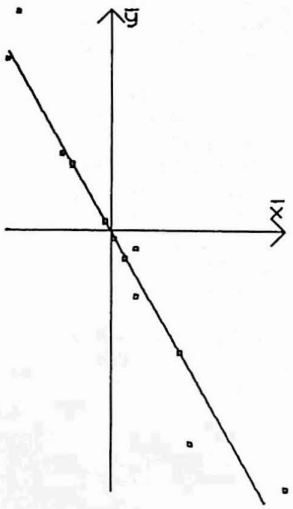
$r = Sxy/\sqrt{Sxx\,Syy}$ ····· Correlation coefficient

$a = Sxy/Sxx$
$b = \bar{y} - a\bar{x}$
} Regression coefficient ($y = ax + b$)

[Printout]

The actual printout is colored. Refer to page 1 .

```
COVARIANCE=
      -3.060714286
CORRELATION=
  -9.693968513E-01
REGRESS.COEFF.
A=-3.942042318
B= 39.4475621
*MEAN*
X= 7.375
Y= 10.375
```




```
*ESTIMATION*
X= 7
Y= 11.85326587
X= 8
Y= 7.911223556
X= 7.5
Y= 9.882244715
X= 7.3
Y= 10.67065318
X= 7.4
Y= 10.27644895
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	X = _	
2	6.9 ENTER	Y = _	
3	12 ENTER	X =	The display returns to step (1). Press the ENTER key in step (2) or repeat the procedure until 10 sets of data are keyed-in.
⋮	⋮	⋮	
18	ENTER	>	
19	DEF S	>	Data output with the > display ends the operation, during which the variance, and other data are printed.
20	DEF D	ESTIMATION = _	The graph is printed before the display appears.
21	7 ENTER	ESTIMATION = _	The display returns to step (20). Key-in 10 data or repeat the procedure until only the ENTER is pressed.
⋮	⋮	⋮	

PROGRAM TITLE	CORRELATION COEFFICIENT, LINEAR REGRESSION AND PLOT	PROGRAM NO. P5-B-1	3

[Program List]

```
 10:"A":CLEAR
 20:DIM X(9),Y(9)
 30:FOR B=1TO 10
 40:X(B-1)=0:Y(B-1
    )=0
 50:NEXT B
 60:N=0
 70:FOR B=1TO 10
 80:INPUT "X=";X(B
    -1):GOTO 95
 90:GOTO 120
 95:INPUT "Y=";Y(B
    -1)
100:N=N+1
110:NEXT B
120:END
130:"S":I=0:J=0:K=
    0:L=0:M=0
140:P=10^(98):O=-P
    :R=P:Q=O
150:FOR B=1TO N
155:Z=B-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)
210:IF P>X(Z)LET P
    =X(Z)
220:IF O<X(Z)LET O
    =X(Z)
230:IF R>Y(Z)LET R
    =Y(Z)
240:IF Q<Y(Z)LET Q
    =Y(Z)
250:NEXT B
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=J(K*M)
307:H=L/H
310:COLOR 0:LPRINT
    "COVARIANCE=",
    L/(N-1)
320:LPRINT "CORREL
    ATION=",H
330:LPRINT "REGRES
    S.COEFF."
340:S=L/K:T=J-S*I
350:LPRINT "A=";S
360:LPRINT "B=";T
362:LPRINT "*MEAN*
    "
364:LPRINT "X=";I
366:LPRINT "Y=";J
370:END
500:"D":GRAPH
510:A=(O-P)/200
520:B=(Q-R)/350
530:C=(I-P)/A
540:D=(R-J)/B
550:GLCURSOR (C,D)
560:SORGN
570:X1=-(I-P)/A:Y1
    =0
580:X2=(O-I)/A:Y2=
    0
590:GOSUB 900
600:LINE (X2-10,Y2
    -10)-(X2,Y2)
605:LINE (X2,Y2)-(
    X2-10,Y2+10)
610:LPRINT "x"
620:LINE (X2-10,Y2
    +23)-(X2,Y2+23
    )
630:X1=0:Y1=-(J-R)
    /B
640:X2=0:Y2=(Q-J)/
    B
650:GOSUB 900
660:LINE (X2-10,Y2
    -10)-(X2,Y2)
665:LINE (X2,Y2)-(
    X2+10,Y2-10)
670:LPRINT "y"
680:LINE (X2+10,Y2
    )-(X2+20,Y2)
690:FOR E=1TO N
700:X=(X(E-1)-I)/A
    :Y=(Y(E-1)-J)/
    B
710:GOSUB 920
720:NEXT E
730:X1=-(I-P)/A:Y1
    =((S*P+T)-J)/B
740:X2=(O-I)/A:Y2=
    ((S*O+T)-J)/B
750:COLOR 2
760:GOSUB 900
770:N=1
780:INPUT "ESTIMAT
    ION=";X(N-1):
    GOTO 800
790:GOTO 840
800:Y(N-1)=S*X(N-1
    )+T
810:X=(X(N-1)-I)/A
    :Y=(Y(N-1)-J)/
    B
820:LINE (X-1,Y-1)
    -(X+2,Y+2),0,3
    ,B
830:N=N+1:GOTO 780
840:GLCURSOR (-(I-
    P)/A,-(J-R)/B-
    20)
845:TEXT
850:IF N=1END
860:COLOR 0:LPRINT
    "*ESTIMATION*"
870:FOR W=1TO N-1
880:LPRINT "X=";X(
    W-1)
890:LPRINT "Y=";Y(
    W-1)
895:NEXT W
896:END
900:LINE (X1,Y1)-(
    X2,Y2)
910:RETURN
920:LINE (X,Y)-(X+
    2,Y+2),0,1,B
930:RETURN
```

```
STATUS 1
                    1468
```

PROGRAM TITLE	CORRELATION COEFFICIENT, LINEAR REGRESSION AND PLOT	PROGRAM NO. P5–B–1	4

[Memory Contents]

A	Graph coefficient (Par 1 dot) X	A$		X(9)	Input. Estimation (= x) data table
B	Graph coefficient (Par 1 dot) Y	B$		Y(9)	Input. Estimation (= y) data table
C	√	C$			
D	√	D$			
E	√	E$		X1	Line draw subroutine (Start X coordinate)
F		F$		Y1	Line draw subroutine (Start Y coordinate)
G		G$		X2	Line draw subroutine (End X coordinate)
H		H$		Y2	Line draw subroutine (End Y coordinate)
I	$\overline{X}$	I$			
J	$\overline{Y}$	J$			
K	$Sxx = \Sigma Xi^2 - n\overline{X}^2$	K$			
L	$Sxy = \Sigma Xi \cdot Yi - n\overline{X}\overline{Y}$	L$			
M	$Syy = \Sigma Yi^2 - n\overline{Y}^2$	M$			
N	Number (Data) n	N$			
O	X-MAX	O$			
P	X-MIN	P$			
Q	Y-MAX	Q$			
R	Y-MIN	R$			
S	Regression coefficient a	S$			
T	Regression coefficient b	T$			
U		U$			
V		V$			
W	√	W$			
X	√	X$			
Y	√	Y$			
Z	√	Z$			

SHARP

PROGRAM TITLE	EXPONENTIAL REGRESSION AND PLOT	PROGRAM NO. P5–B–2	1

CE–150 required

[Outline]

With the input data x and y applied to the exponential curve $y = a \cdot b^x$, coefficients a and b, and correlation coefficient r are determined.
Next, the exponential curve is printed out by the printer, and the input data and estimated values are plotted.

[Operating Guide]

DEF A : Data input, printouts of coefficients a and b, and correlation coefficient r. Up to 39 data are possible.

DEF B : Exponential curve output and input data are plotted on the graph.
New X data are keyed-in and corresponding Y will be plotted. The inputs of X are possible up to 39.
For plottable data of estimations, the estimated y should be less than the maximum value of the input data Y_i.

[Example]

x	0.5	1.2	3.1	7.4
y	7.01	11.72	44.54	936.71

n = 4

Apply the above data to $y = ab^x$, and estimate the values when x = 2, 4, 6, and 6.5.

[Contents] (Formulas)

Find the coefficients a and b so that the graph of $y=ab^x$ . . . (1) is most applicable to the given number (n) of points $(x_1, y_1), (x_2, y_2) \ldots . (x_n, y_n)$.
The method of least squares is normally used for the curve application. The exponential function is, however, difficult to handle, therefore, the conversion is made by using the logarithm.
Taking the logarithm of both sides of Eq. (1) $y=ab^x$ (using natural logarithm) yields:

$$\ell n\, y = \ell n\, a + x \ell n\, b \quad \ldots\ldots (2)$$

Now, assuming $Y = \ell n\, y$, $A = \ell n\, a$, $B = \ell n\, b$, the following is obtained:

$$Y = A + Bx \quad \ldots\ldots (3)$$

Hence, A and B can be calculated as follows:

$$A = \bar{Y} - B\bar{x}\ ,\ B = \frac{\Sigma\, xi\, Yi - n\bar{x}\bar{Y}}{\Sigma\, x^2\, i - n\bar{x}^2} \qquad \left(Y = \frac{1}{n}\sum_{i=1}^{n} Y\, i\ ,\ Y\, i = \ell n\, y\, i\ ,\ \bar{x} = \frac{1}{n}\sum_{i+1}^{n} x\, i\right.$$

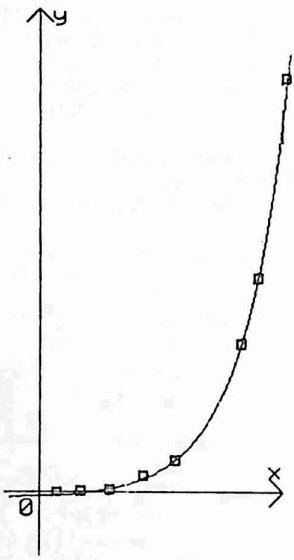
When A and B are found, a and b are determined from $a=e^A$ and $b=e^B$ since $A=\ell n\, a$ and $B=\ell n\, b$.

PROGRAM TITLE	EXPONENTIAL REGRESSION AND PLOT	PROGRAM NO. P5-B-2	2

[Printout]

The actual printout is colored. Refer to page 1.

```
R= 9.999942365E-01
A= 4.960331916
B= 2.03057723
```




```
* ESTIMATION *
X= 2
Y= 20.45265825
X= 4
Y= 84.3312981
X= 6
Y= 347.7185094
X= 6.5
Y= 495.4930476
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	N =_	
2	4 ENTER	X(1) =?	
3	0.5 ENTER	Y(1) =?	
4	7.01 ENTER	X(2) =?	
5	1.2 ENTER	Y(2) =?	
6	11.72 ENTER	X(3) =?	
7	3.1 ENTER	Y(3) =?	
8	44.54 ENTER	X(1) =?	
9	7.4 ENTER	Y(1) =?	
10	936.71 ENTER	>	A, B, R are printed out to complete key operation.
11	DEF B	ESTIMATION X =_	Display appears after the graph output.
12	2 ENTER	ESTIMATION X =_	
13	4 ENTER	ESTIMATION X =_	
14	6 ENTER	ESTIMATION X =_	
15	6.5 ENTER	>	

PROGRAM TITLE	EXPONENTIAL REGRESSION AND PLOT	PROGRAM NO. P5–B–2	3

[Program List]

```
 10:"A":CLEAR :
    WAIT 0:CLS
 20:INPUT "N=";N
 30:DIM X(N-1),Y(N
    -1):E=10^8:G=E
    :D=-E:F=D
 40:FOR I=0TO I
 50:CLS :A$="X("+
    STR$ (I+1)+")=
    "
 60:PRINT A$;
 70:INPUT X(I):
    GOTO 90
 80:N=I:GOTO 150
 90:CLS :A$="Y("+
    STR$ (I+1)+")=
    "
100:PRINT A$;
110:INPUT Y(I):Y=
    LN Y(I)
112:IF D<X(I)LET D
    =X(I)
114:IF E>X(I)LET E
    =X(I)
116:IF F<Y(I)LET F
    =Y(I)
118:IF G<Y(I)LET G
    =Y(I)
120:O=O+X(I):P=P+Y
130:Q=Q+X(I)*X(I):
    R=R+Y*Y:S=S+X(
    I)*Y
140:NEXT I
150:X=O/N:Y=P/N
160:T=Q-N*X*X
170:U=S-N*X*Y
180:V=R-N*Y*Y
190:C=U/SQR(T*V)
200:B=U/T
210:A=EXP (Y-B*X)
220:B=EXP B
225:COLOR 0
230:LPRINT "R=";C
240:LPRINT "A=";A
250:LPRINT "B=";B:
    END
260:"B":M=F/300
270:IF E>=0LET Z=2
    5:L=D/175:GOTO
    290
280:L=(D+ABS E)/20
    0:Z=ABS E/L+5
290:GRAPH :
    GLCURSOR (Z,-3
    50):SORGN
300:LINE (-Z,0)-(2
    00-Z,0)-(200-Z
    -10,-10)-(200-
    Z,0)-(200-Z-10
    ,10):LPRINT "x
    "
310:LINE (0,-50)-(
    0,350)-(-10,34
    0)-(0,350)-(10
    ,340):LPRINT "
    y"
320:GLCURSOR (-15,
    -15):LPRINT "0
    "
330:COLOR 1:FOR I=
    0TO N-1
340:J=X(I)/L:K=Y(I
    )/M
350:LINE (J-3,K-3)
    -(J+3,K+3),0,1
    ,B
360:NEXT I:COLOR 2
370:J=-Z:K=A*B^(J*
    L)/M
380:J1=J+2:IF J>20
    0-ZGOTO 400
390:K1=A*B^(J1*L)/
    M:IF K1>350
    GOTO 400
395:LINE (J,K)-(J1
    ,K1):J=J1:K=K1
    :GOTO 380
400:I=0
410:IF I>=NTHEN 47
    0
420:CLS :INPUT "ES
    TIMATION X=";X
    (I):GOTO 440
430:N=I:GOTO 470
440:J=X(I)/L:Y(I)=
    A*B^X(I):K=Y(I
    )/M
445:IF K>350GOTO 4
    60
450:LINE (J-3,K-3)
    -(J+3,K+3),0,3
    ,B
460:I=I+1:GOTO 410
470:GLCURSOR (0,-1
    00):TEXT :
    COLOR 0
500:LPRINT "* ESTI
    MATION *"
510:FOR I=0TO N-1
520:LPRINT "X=";X(
    I)
530:LPRINT "Y=";Y(
    I)
540:NEXT I
550:END

STATUS 1
                1187
```

[Memory Contents]

A	a
B	$b' \cdot b$
C	Correlation coefficient r
D	X-MAX
E	X-MIN
F	Y-MAX
G	Y-MIN
H	
I	√
J	√
K	√
L	X print coefficient
M	Y print coefficient
N	No. of coordinates
O	ΣXi
P	ΣY
Q	$\Sigma X^2 i$
R	ΣY^2
S	ΣXiY
T	Sxx
U	Sxy
V	Syy
W	
X	$\overline{X}$
Y	$\ell n\ yi, \overline{Y}$
Z	√
A$	√
X(N–1)	X data: Estimated X
Y(N–1)	Y data: Estimated Y
J1	√
K1	√

SHARP

PROGRAM TITLE	MODIFIED EXPONENTIAL CURVE	PROGRAM NO. P5–B–3	1

CE–150 and CTR required

[Oultine]

With a modified exponential curve written as $y = k - ab^x$, factors a and b (also k if unknown) are calculated when k is known and unknown. This program also estimates value of y for the new x.

[Operating Guide]

DEF A k is known;

Input { No. of data, k value, (x_i, y_i), Estimate x } Output { Coefficient a, b, Estimate x, y }

No. of data is possible up to 36. Cassette tape File name "MEC–DATA (K)"

DEF B k is unknown;

Input { No. of data, yi, Estimate x } Output { Coefficient a, b, k, Estimate x, y }

No. of data is unlimited. The cassette tape file name is "MEC–DATA".

[Example]

1. k is known;

$k = 550$

x_i	1	2	12	35	60
y_i	540.2	540.4	542	545	547

Estimate $x = 5$, $x = 15$

2. k is unknown;

x_i	1	2	3	4	5
y_i	33.8	38.9	37.7	42.5	46.3
x_i	6	7	8	9	10
y_i	50.6	55.2	58.9	58.0	60.5
x_i	11	12	13	14	15
y_i	62.8	63.5	60.4	63.9	68.2

Estimate $x = 15$, $x = 16$

PROGRAM TITLE	MODIFIED EXPONENTIAL CURVE	PROGRAM NO. P5–B–3	2

[Contents] (Formulas)

1. k is known;

When taken the logarithms of both members in $k-y=ab^x$, which is from $y=k-ab^x$, result in; $\ell n\,(k-y) = \ell na + x\ell nb$.

With $Y = \ell n\,(k-y)$, $A = \ell na$, $B = \ell nb$, it is obtained that $Y = A + Bx$.

From the least square method, the results are;

$$A = \frac{\Sigma x^2 \Sigma Y - \Sigma x \Sigma xY}{n\Sigma x^2 - (\Sigma x)^2}$$

$$B = \frac{n\Sigma xY - \Sigma x \Sigma Y}{n\Sigma x^2 - (\Sigma x)^2} \qquad \begin{cases} a = e^A \\ b = e^B \end{cases}$$

2. k is unknown;

Datas, assumed as 3n (if No. of datas is undividable by 3, the remainder is omitted), is divided into 3 parts; $0 \leq x < n$, $n \leq x < 2n$, and $2n \leq x < 3n$, with sums of respective parts written as;

$$\Sigma_1 y = \sum_{i=1}^{n} y_i$$

$$\Sigma_2 y = \sum_{i=n+1}^{2n} y_i$$

$$\Sigma_3 y = \sum_{i=2n+1}^{3n} y_i$$

the following is obtained:

$$b = \left(\frac{\Sigma_3 y - \Sigma_2 y}{\Sigma_2 y - \Sigma_1 y}\right)^{\frac{1}{n}}$$

$$a = (\Sigma_1 y - \Sigma_2 y)\,\frac{b-1}{(b^n-1)^2}$$

$$k = \frac{1}{n}\left(\Sigma_1 y + \left(\frac{b^n-1}{b-1}\right) a\right)$$

[Printout]

```
a= 10.0556453
b= 9.801181777E-01

* ESTIMATE *
X= 5
Y= 540.9050113
X= 15
Y= 542.5597658
```

```
a= 39.91657038
b= 8.422366627E-01
k= 68.9970248

* ESTIMATE *
X= 15
Y= 65.95848202
X= 16
Y= 66.43785267
```

PROGRAM TITLE	MODIFIED EXPONENTIAL CURVE	PROGRAM NO. P5–B–3	3

[Key Operation Procedure] : k is known;

Step No.	Input	Display	Remarks
1	DEF A	DATA CLOAD? (Y, N) _	
2	Y ENTER	DATA CSAVE? (Y, N) _	After data input from cassette tape and data printout, go to step 15.
	N ENTER	N = _	Goes to step 3.
3	5 ENTER	K = _	
4	550 ENTER	X (1) =?	
5	1 ENTER	Y (1) =?	
6	540.2 ENTER	X (2) =?	
7	2 ENTER	Y (2) =?	Repeated data input
⋮	⋮	⋮	
13	60 ENTER	Y (5) =?	
14	547 ENTER	DATA CSAVE? (Y, N) _	After data print
15	Y ENTER	X = _	Data output to cassette tape
	N ENTER	X = _	
16	5 ENTER	X = _	x Input
17	15 ENTER	X = _	x Input
18	ENTER	>	End

[Key Operation Procedure] : k is unknown;

Step No.	Input	Display	Remarks
1	DEF B	DATA CLOAD? (Y, N) _	
2	Y ENTER	DATA CSAVE? (Y, N) _	After data input from cassette tape and data printout, goes to step 9.
	N ENTER	N = _	Goes to step 3.
3	15 ENTER	Y (1) =?	
4	33.8 ENTER	Y (2) =?	
5	38.9 ENTER	Y (3) =?	Repeated data input
⋮	⋮	⋮	
17	63.9 ENTER	Y (15) =?	
18	68.2 ENTER	DATA CSAVE ? (Y, N) _	Display after printout.

PROGRAM TITLE	MODIFIED EXPONENTIAL CURVE	PROGRAM NO. P5–B–3	4

Step No.	Input	Display	Remarks
19	Y [ENTER]	X = _	Display after data output to cassette tape
	N [ENTER]	X = _	
20	15 [ENTER]	X = _	x input
21	16 [ENTER]	X = _	x input
22	[ENTER]	>	End

[Program List]

```
10:"A":CLEAR :
   WAIT 0
20:INPUT "DATA CL
   OAD?(Y,N)";A$
30:IF (A$="Y")+(A
   $="N")<>1GOTO
   20
40:IF A$="Y"GOTO
   150
50:INPUT "N=";D,"
   K=";C
55:DIM X(D-1),Y(D
   -1)
60:FOR I=0TO D-1
70:CLS :A$="X("+
   STR$ (I+1)+")=
   "
80:PRINT A$;
90:INPUT X(I):
   GOTO 110
100:CLS :D=I-1:
   GOTO 180
110:CLS :A$="Y("+
   STR$ (I+1)+")=
   "
120:PRINT A$;
130:INPUT Y(I)
140:NEXT I:GOTO 18
   0
150:INPUT #"MEC-DA
   TA(K)";D,C
160:DIM X(D-1),Y(D
   -1)
170:INPUT #"MEC-DA
   TA(K)";X(*),Y(
   *)
180:CLS :FOR I=0TO
   D-1
185:Y=LN (C-Y(I))
190:E=E+X(I):F=F+X
   (I)*X(I)
200:G=G+Y:H=H+X(I)
   *Y
210:NEXT I
220:B=D*F-E*E
230:A=(F*G-E*H)/B
240:B=(D*H-E*G)/B
250:A=EXP A:B=EXP
   B
260:BEEP 3:LPRINT
   "a=";A
270:LPRINT "b=";B
280:BEEP 5:INPUT "
   DATA CSAVE?(Y,
   N)";A$
290:IF (A$="Y")+(A
   $="N")<>1GOTO
   280
300:IF A$="N"GOTO
   700
310:PRINT #"MEC-DA
   TA(K)";D,C
320:PRINT #"MEC-DA
   TA(K)";X(*),Y(
   *)
330:GOTO 700
400:"B":CLEAR :
   WAIT 0:CLS :
   DIM Y(2)
410:INPUT "DATA CL
   OAD?(Y,N)";A$
420:IF (A$="Y")+(A
   $="N")<>1GOTO
   410
430:IF A$="Y"THEN
   530
440:INPUT "N=";N
450:N=INT (N/3)
460:FOR C=1TO 3
470:FOR X=N*(C-1)
   TO N*C-1
480:CLS :A$="Y("+
   STR$ (X+1)+")=
   "
490:PRINT A$;
500:INPUT L
510:Y(C-1)=Y(C-1)+
   L
520:NEXT X:NEXT C:
   GOTO 540
530:INPUT #"MEC-DA
   TA";N,Y(*)
540:CLS :C=N:B=((Y
   (2)-Y(1))/(Y(1
   )-Y(0)))^(1/C)
550:D=B^C-1:A=(Y(0
   )-Y(1))*(B-1)/
   (D*D)
560:C=(Y(0)+D*A/(B
   -1))/C
570:BEEP 3:LPRINT
   "a=";A
580:LPRINT "b=";B
590:LPRINT "k=";C
600:BEEP 5:INPUT "
   DATA CSAVE?(Y,
   N)";A$
610:IF (A$="Y")+(A
   $="N")<>1GOTO
   600
620:IF A$="N"GOTO
   700
630:PRINT #"MEC-DA
   TA";N,Y(*)
700:LF 1:LPRINT "*
    ESTIMATE *":
   CLS
710:BEEP 1:INPUT "
   X=";X:GOTO 730
720:END
730:LPRINT "X=";X
740:LPRINT "Y=";C-
   A*B^X
750:GOTO 710

STATUS 1
                1264
```

[Memory Contents]

A	a
B	b
C	k
D	n
E	Σxi
F	Σxi^2
G	Σy
H	$\Sigma xi \cdot y$
I	
J	
K	
L	yi
M	
N	n
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	x
Y	y
Z	
F$	xi
G$	yi
A$	$\sqrt{}$

SHARP

PROGRAM TITLE	LOGISTIC CURVE	PROGRAM NO. P5–B–4	1

CE–150, CE–151 and CTR required

[Outline]

Using a logistic curve, the input data are approximated to find the estimated value of y for the new value of x.

General form of Logistic curve: $y = \dfrac{k}{1 + me^{-ax}}$

[Operating Guide]

[DEF] [A] : Used for coefficient calculation when k is known.

Input:
- n: No. of data
- k:
- $X_1 \sim Xn$
- $Y_1 \sim Yn$

Output:
- Coefficient a
- Coefficient m

[DEF] [B] : Used for coefficient calculation when k is unknown.

Input:
- n: No. of data
- $Y_1 \sim Yn$

Output:
- Coefficient a
- Coefficient m
- Coefficient k

The effective number of data is up to the multiple of 3.

[DEF] [C] : Graph, data and plot outputs of estimate value.

Input: Estimate value (X). The number of estimate value inputs is up to the number of data designated by the [DEF] [A] and [DEF] [B]

[DEF] [D] : Printouts of the estimate value, X and Y.

Note : Data input in the [DEF] [A] and [DEF] [B] is also possible from the cassette tape recorder.
The keyed-in data can be output to the cassette tape.

[Example]

1. k is known:

$k = 195$

i	y_i
2	11
6	54
10	150

Estimate value

$x = 5$

$x = 12$

2. k is unknown:

y

i	yi	i	yi
1	40	11	388
2	50	12	475
3	67	13	591
4	88	14	713
5	119	15	845
6	146	16	983
7	182	17	1143
8	223	18	1256
9	273	19	1377
10	322	20	1513

Estimate value

$x = 10$

〃 $= 15$

〃 $= 18$

〃 $= 19$

[Contents] (Formulas)

1. k is known:

$$y = \frac{k}{1 + m e^{-ax}}$$

ℓn: Natural logarithm

$$\frac{k}{y} - 1 = m e^{-ax}$$

$$\ell n \left(\frac{k}{y} - 1\right) = \ell n\, m - a x$$

Putting $Y = \ell n \left(\frac{k}{y} - 1\right)$, $A = \ell n\, m$, $B = a$, the following is obtained.

From the least square method, the results are:

$$A = \frac{\Sigma x^2 \Sigma Y - \Sigma x \Sigma x Y}{n \Sigma x^2 - (\Sigma x)^2}$$

$$B = \frac{n \Sigma x Y - \Sigma x \Sigma Y}{n \Sigma x^2 - (\Sigma x)^2} \qquad \begin{cases} m = e^{A} \\ a = -B \end{cases}$$

2. k is unknown:

The reciprocal of both members in a curve formula is taken to write the following:

$$\frac{1}{y} = \frac{1}{k} + \frac{m}{k} e^{-ax}$$

with $Y = \frac{1}{y}$, $K = \frac{1}{k}$, $A = \frac{m}{k}$, $B = e^{-a}$

$Y = K - AB^x$ is obtained.

This is determined by the method of a modified exponential curve, as follows:

$$B = \left(\frac{\Sigma_3 Y - \Sigma_2 Y}{\Sigma_2 Y - \Sigma_1 Y}\right)^{\frac{1}{n}}$$

$$A = (\Sigma_1 Y - \Sigma_2 Y)\ \frac{B-1}{(B^n-1)^2}$$

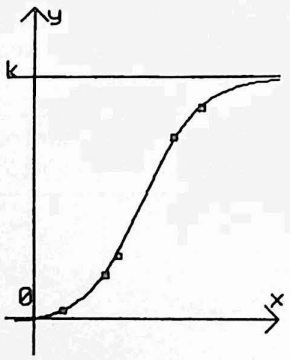
$$K = \frac{1}{n}\left[\Sigma_1 Y + \left(\frac{B^n - 1}{B-1}\right) A\right]$$

$$\begin{cases} a = -\ell n\, B \\ k = 1/K \\ m = -kA \end{cases}$$

[Printout]

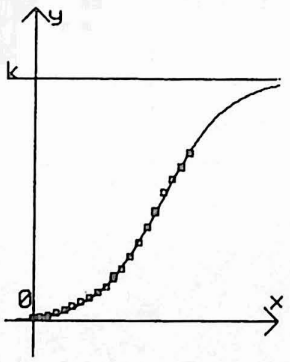
The real printout is colored.
Refer to page 1.

```
A= 5.026266613E-01
M= 48.10443978
```




```
* ESTIMATE *
X= 5
Y= 39.8192162
X= 12
Y= 174.8033605
```

```
A= 2.507446178E-01
M= 50.49168896
K= 2115.67291
```




```
* ESTIMATE *
X= 10
Y= 413.7132289
X= 15
Y= 973.0535461
X= 18
Y= 1361.923995
X= 19
Y= 1478.765671
```

PROGRAM TITLE	LOGISTIC CURVE	PROGRAM NO. P5–B–4	4

[Key Operation Procedure] : k is known.

Step No.	Input	Display	Remarks
1	DEF A	DATA CLOAD? (Y, N) _	
2	Y ENTER		After data input from the cassette tape, the results are printed out to complete processing.
	N ENTER	N = _	To 3
3	3 ENTER	K = _	
4	195 ENTER	X (1) =?	
5	2 ENTER	Y (1) =?	
6	11 ENTER	X (2) =?	
7	6 ENTER	Y (2) =?	
8	54 ENTER	X (3) =?	
9	10 ENTER	Y (3) =?	
10	150 ENTER	DATA CSAVE? (Y, N) _	
11	Y ENTER	>	After data output to the cassette tape, the results are printed out to complete processing.
	N ENTER	>	The results are printed out to complete processing.

PROGRAM TITLE	LOGISTIC CURVE	PROGRAM NO. P5–B–4	5

[Key Operation Procedure] : k is unknown.

Step No.	Input	Display	Remarks
1	DEF B	DATA CLOAD? (Y, N)_	
2	Y ENTER		After data input from the cassette tape, the results are printed out to complete processing.
	N ENTER	N =_	To 3
3	20 ENTER	Y (1) = ?	
4	40 ENTER	Y (2) = ?	
5	50 ENTER	Y (3) = ?	
⋮	⋮	⋮	Repeated input.
19	983 ENTER	Y (17) = ?	
20	1143 ENTER	Y (18) = ?	
21	1256 ENTER	DATA CSAVE? (Y, N)_	
22	Y ENTER	>	After data input from the cassette tape, the results are printed out to complete processing.
	N ENTER	>	The results are printed out to complete processing.

[Key Operation Procedure] : Graph output, Estimate plot and Estimate value printout

Step No.	Input	Display	Remarks
1	DEF C	ESTIMATE X =_	After graph output, the display appears.
2	5 ENTER	ESTIMATE X =_	
3	12 ENTER	ESTIMATE X =_	
4	ENTER	>	Processing end.

PROGRAM TITLE	LOGISTIC CURVE	PROGRAM NO. P5-B-4	6

[Program List]

```
 10:"A":CLEAR :
    WAIT 0
 20:INPUT "DATA CL
    OAD?(Y,N)";A$
 30:IF (A$="Y")+(A
    $="N")<>1GOTO
    20
 40:IF A$="Y"GOTO
    130
 50:INPUT "N=";D,"
    K=";C
 60:DIM X(D-1),Y(D
    -1)
 70:FOR I=0TO D-1
 80:A$="X("+STR$ (
    I+1)+")=":
    PRINT A$;
 90:INPUT X(I)
100:CLS :A$="Y("+
    STR$ (I+1)+")=
    ":PRINT A$;
110:INPUT Y(I)
120:CLS :NEXT I
122:INPUT "DATA CS
    AVE?(Y,N)";A$
124:IF (A$="Y")+(A
    $="N")<>1GOTO
    122
126:IF A$="N"GOTO
    150
127:PRINT #D,C
128:PRINT #X(*),Y(
    *):GOTO 150
130:INPUT #D,C
140:DIM X(D-1),Y(D
    -1):INPUT #X(*
    ),Y(*)
150:X1=10^8:X2=-X1
160:FOR I=0TO D-1
170:IF X(I)<X1LET
    X1=X(I)
180:IF X(I)>X2LET
    X2=X(I)
210:Y=LN (C/Y(I)-1
    ):E=E+X(I):F=F
    +X(I)*X(I)
220:G=G+Y:H=H+X(I)
    *Y:NEXT I
230:B=D*F-E*E:A=(F
    *G-E*H)/B:B=(D
    *H-E*G)/B
240:A=EXP A:B=-B:
    LPRINT "A=";B:
    LPRINT "M=";A
250:END
260:"B":CLEAR :
    WAIT 0:USING
270:INPUT "DATA CL
    OAD?(Y,N)";A$
280:IF (A$="Y")+(A
    $="N")<>1GOTO
    270
290:IF A$="Y"GOTO
    420
300:INPUT "N=";D:A
    =INT (D/3)
310:DIM X(A*3-1),Y
    (A*3-1),B(2)
320:FOR C=1TO 3
330:FOR I=(C-1)*A
    TO C*A-1
340:A$="Y("+STR$ (
    I+1)+")=":
    PRINT A$;
350:INPUT Y(I):X(I
    )=I:CLS
360:B(C-1)=B(C-1)+
    1/Y(I)
370:NEXT I:NEXT C:
    Z=I:D=3*A
372:Y1=Y(0):Y2=Y(0
    )
373:FOR I=1TO D-1
374:IF Y(I)<Y1LET
    Y1=Y(I)
375:IF Y(I)>Y2LET
    Y2=Y(I)
376:NEXT I
380:INPUT "DATA CS
    AVE?(Y,N)";A$
390:IF (A$="Y")+(A
    $="N")<>1GOTO
    380
400:IF A$="N"GOTO
    425
410:PRINT #D,A:
    PRINT #X(*),Y(
    *),B(*):GOTO 4
    25
420:INPUT #D,A:DIM
    X(D-1),Y(D-1),
    B(2):INPUT #X(
    *),Y(*),B(*):
    GOTO 372
425:X1=0:X2=I*3*A-
    1
430:C=A:B=((B(2)-B
    (1))/(B(1)-B(0
    )))^(1/C)
440:D1=B^C-1:A=(B(
    0)-B(1))*(B-1)
    /(D1*D1)
450:C=(B(0)+D1*A/(
    B-1))/C
460:C=1/C:A=-A*C:B
    =-LN B
462:X1=-1/B*LN ((C
    /Y1-1)/A)
464:X2=-1/B*LN ((C
    /Y2-1)/A)
470:LPRINT "A=";B
480:LPRINT "M=";A
490:LPRINT "K=";C:
    END
500:"C":GRAPH :U=1
    0:W=-250:IF C<
    0LET U=-20:W=-
    50
505:IF X1>0LET X1=
    0
510:X3=X1:M=X2-X3
520:N=M/100:L=C/17
    5
530:GLCURSOR (25,W
    ):SORGN
540:COLOR 0:
    GLCURSOR (-10,
    U):LPRINT "0"
550:LINE (-20,0)-(
    175,0):LINE (1
    65,10)-(175,0)
    -(165,-10)
560:GLCURSOR (170,
    U):LPRINT "x"
570:IF C>=0LINE (0
    ,-25)-(0,225):
    LINE (-10,215)
    -(0,225)-(10,2
    15):GOTO 590
580:LINE (0,25)-(0
    ,-225):LINE (-
    10,-215)-(0,-2
    25)-(10,-215)
590:LPRINT "y":
    COLOR 1:T=C/L
600:LINE (175,T)-(
    -20,T):LPRINT
    "k"
610:COLOR 2:O=-15:
    S=O*N:P=(C/(1+
    A*EXP (-B*S)))
    /L
620:IF O>=175GOTO
    650
630:S=(O+5)*N:Q=(C
    /(1+A*EXP (-B*
    S)))/L
```

(To be continued)

PROGRAM TITLE	LOGISTIC CURVE	PROGRAM NO. P5–B–4	7

[Program List]

```
640:LINE (O,P)-(O+
    5,Q):O=O+5:P=Q
    :GOTO 620
650:FUR I=0TO D-1
660:S=X(I)/N:R=Y(I
    )/L
670:LINE (S-2,R-2)
    -(S+2,R+2),0,1
    ,B
680:NEXT I
690:I=0
700:IF I<DINPUT "E
    STIMATE X=";X(
    I):GOTO 730
710:END


730:Y(I)=C/(1+A*
    EXP (-B*X(I)))
    :S=X(I)/N:R=Y(
    I)/L:I=I+1
740:LINE (S-2,R-2)
    -(S+2,R+2),0,3
    ,B:GOTO 700
800:"D":GLCURSOR (
    0,0):GLCURSOR
    (0,-(300+W))
810:TEXT :COLOR 0:
    LPRINT "* ESTI
    MATE *"
820:FOR J=0TO I-1
830:LPRINT "X=";X(
    J)
840:LPRINT "Y=";Y(
    J)
850:NEXT J
860:END


STATUS 1
          2176
```

[Memory Contents]

A	m	X(D–1)	X_n } Data
B	a	Y(D–1)	Y_n }
C	k		
D	n	X1	Min. of Xn
E	ΣX	X2	Max. of Xn
F	ΣX^2	X3	X min. on the graph
G	ΣY	B(2)	B(0) : $\Sigma_1 Y$
H	ΣxY		B(1) : $\Sigma_2 Y$
I	√		B(2) : $\Sigma_3 Y$
J	√	D1	√
K		Y1	Min. of y_n
L	Graph coefficient y	Y2	Max. of y_n
M	Range (graph) value		
N	Graph coefficient X		
O	√		
P	√		
Q	√		
R	√		
S	√		
T	√		
U	√		
V	√		
W	√		
X			
Y	X		
Z	y, Y		
A$	√		

SHARP

PROGRAM TITLE	MODIFIED MOVING AVERAGE	PROGRAM NO. P5–B–6

CE–150 required

[Outline]

This program is used to determine the modified moving average.
For regression analysis, the effects of minor cyclic variations can be cancelled by averaging the movement, if any, based on a cycle.

[Operating Guide]

DEF A : Number input of averaging items (n) of the modified moving average.
With the input of data, the printouts are made for input values and mean values.

[Example]

1. Find the modified moving average of 4 items.

Data: 56, 79, 0, 97
20, 23, 99, 68
34, 93, 31

[Contents] (Formulas)

Processing varies with the number of averaging items (n) being an odd number or even number.

1. n is an odd number:

$$X_1 = \sum_{i=1}^{n} Xi / n$$

$$X_2 = \sum_{i=2}^{n+1} Xi / n$$

⋮

2. n is an even number:

$$X_1 = \left(\frac{X_1}{2} + \frac{X_{n+1}}{2} + \sum_{i=2}^{n} Xi\right) / n$$

$$X_2 = \left(\frac{X_2}{2} + \frac{X_{n+2}}{2} + \sum_{i=3}^{n+1} Xi\right) / n$$

⋮

PROGRAM TITLE	MODIFIED MOVING AVERAGE	PROGRAM NO. P5–B–6	2

[Printout]

```
** X= 56
** X= 79
** X= 0
** X= 97
** X= 20
M.U.= 53.5
** X= 23
M.U.= 42
** X= 99
M.U.= 47.375
** X= 68
M.U.= 56.125
** X= 34
M.U.= 54.25
** X= 93
M.U.= 64.75
** X= 31
M.U.= 65
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	N =–	
2	4 ENTER	X =–	Printouts of data
3	56 ENTER	X =–	
4	79 ENTER	X =–	
5	0 ENTER	X =–	
6	97 ENTER	X =–	
7	20 ENTER	X =–	Printouts of mean value
8	23 ENTER	X =–	
⋮	⋮	⋮	
13	31 ENTER	X =–	
14	ENTER	>	

PROGRAM TITLE	MODIFIED MOVING AVERAGE	PROGRAM NO. P5–B–6	3

[Program List]

```
10:"A":CLEAR :
   INPUT "N= ";A
20:E=0:DIM X(A-1)
30:IF A<>INT (A*0
   .5)*2GOTO 130
40:FOR C=0TO A-1
50:GOSUB 500
60:NEXT C
70:FOR C=0TO A-1
80:INPUT "X=";D:
   GOTO 90
85:END
90:E=E+D:LPRINT "
   ** X=";D
95:LPRINT "M.V.="
   ;(E-.5*(D+X(C)
   ))/A
100:E=E-X(C):X(C)=
   D
110:NEXT C
120:GOTO 70
130:FOR C=0TO A-2
140:GOSUB 500
150:NEXT C
160:B=A-1:INPUT "X
   =";X(B)
170:E=E+X(B):
   LPRINT "** X="
   ;X(B)
180:LPRINT "M.V.="
   ;E/A
190:FOR C=0TO B
200:INPUT "X=";D:
   GOTO 210
205:END
210:E=E-X(C)+D:X(C
   )=D
220:LPRINT "** X="
   ;D
225:LPRINT "M.V.="
   ;E/A:NEXT C
230:GOTO 190
500:INPUT "X=";D
505:LPRINT "** X="
   ;D
510:E=E+D:X(C)=D:
   RETURN
```

STATUS 1

458

[Memory Contents]

A	n
B	n–1
C	√
D	x
E	Σx
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
X(n–1)	Data Table

SHARP

PROGRAM TITLE	TEST OF MEAN VALUE DIFFERENCE AND VARIANCE RATIO	PROGRAM NO. P5–B–7	1

CE–150 required

[Outline]

When 2 populations are normally distributed and their standard deviations are equal, the mean value of normal populations whose values are unknown is equal. Using this program you can test this hypothesis, which also tests whether 2 populations are equal in variance.

[Operating Guide]

DEF A : Used for test of mean value difference (Processed data).

Input:
- No. of data for population 1
- No. of data for population 2
- Mean value of population 1
- Mean value of population 2
- Standard deviation of population 1
- Standard deviation of population 2

Output:
- Test value (T)
- Freedom degree

DEF B : Used for test of mean value difference (processed data).

Input:
- Data of population 1
- Data of population 2

Output:
- Mean value of population 1
- Standard deviation of population 1
- Mean value of population 2
- Standard deviation of population 2
- Test value (T)
- Freedom degree

DEF X : Used to examine variance ratios (processed data).

Input:
- No. of data for population 1
- No. of data for population 2
- Standard deviation of population 1
- Standard deviation of population 2

Output:
- Test value (F)
- Freedom degree 1
- Freedom degree 2

PROGRAM TITLE	TEST OF MEAN VALUE DIFFERENCE AND VARIANCE RATIO	PROGRAM NO. P5–B–7	2

DEF Z : Used to examine variance ratios. (inprocessed data).

Input: { Data of population 1
Data of population 2

Output: { Mean value of population 1
Standard deviation of population 1
Mean value of population 2
Standard deviation of population 2
Test value (F)
Freedome degree 1
Freedome degree 2

[Example]

Test of mean value difference

1	2.3	1.6	2.1	2.2	2.3	2.0	1.9	2.2
2	2.3	2.5	2.0	2.1	2.2	2.1		

Using this data, T testing can be conducted.

$n_1 = 8$ $\bar{x}_1 = 2.075$
$\sigma_1 = 2.375469878\,E-1$
$n_2 = 6$ $\bar{x}_2 = 2.2$
$\sigma_2 = 1.7888854382\,E-1$
σ : Standard deviation

Test of variance ratio

1	1.375	1.407	1.068	1.752	1.201
	1.042	1.223	1.633	1.773	0.779
2	1.033	1.217	1.615	0.673	1.252
	0.984	1.693	0.840		

$n_1 = 10$, $\sigma_1 = 3.261141757\,E-1$
$n_2 = 8$, $\sigma_2 = 3.564527359\,E-1$

F testing is performed on the basis of this data.

[Contents] (Formulas)

Test of mean value difference

When 2 normal populations are equal in variance, and their values remain unknown, testing is done on the hypothesis that their mean values are equal.

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{S_{xx_1} + S_{xx_2}}} \sqrt{\frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2}}$$

This is based on the t distribution of $\phi = n_1 + n_2 - 2$

Test of variance ratio

Testing is conducted to find whether 2 populations are equal in variance.

$F = V_1 / V_2$ is based on the F distribution of $\phi_1 = n_1 - 1$, and $\phi_2 = n_2 - 1$
If $V_1 < V_2$, indices 1 and 2 are interchanged.

PROGRAM TITLE	TEST OF MEAN VALUE DIFFERENCE AND VARIANCE RATIO	PROGRAM NO. P5–B–7	3

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	N1 = _	Test of mean value difference (Processed data)
2	8 ENTER	N2 = _	
3	6 ENTER	MEAN 1 = _	
4	2.075 ENTER	MEAN 2 = _	
5	2.2 ENTER	STD.DEV.1= _	
6	0.2375469878 ENTER	STD.DEV.2= _	
7	0.1788854382 ENTER	>	
8	DEF B	X = _	Test of mean value difference (Inprocessed data)
9	2.3 ENTER	X = _	Sequential inputs of population 1 data
10	1.6 ENTER	X = _	
	⋮	⋮	Repeated data input
17	2.2 ENTER	X = _	
18	ENTER	X = _	Mean value and standard deviation printouts of population 1.
19	2.3 ENTER	X = _	
20	2.5 ENTER	X = _	
21	2.0 ENTER	X = _	
22	2.1 ENTER	X = _	
23	2.2 ENTER	X = _	
24	2.1 ENTER	X = _	
25	ENTER	>	
26	DEF X	N1 = _	Test of variance ratio (Processed data)
27	10 ENTER	N2 = _	
28	8 ENTER	STD. DEV. 1 = _	
29	0.3261141757 ENTER	STD. DEV. 2 = _	
30	0.3564527359 ENTER	>	

PROGRAM TITLE	TEST OF MEAN VALUE DIFFERENCE AND VARIANCE RATIO	PROGRAM NO. P5–B–7	4

Step No.	Input	Display	Remarks
31	DEF Z	X =_	Test of variance ratio (Inprocessed data)
32	1.375 ENTER	X =_	
33	1.407 ENTER	X =_	
34	1.068 ENTER	X =_	
⋮	⋮	⋮	Repeated data input
37	1.773 ENTER	X =_	
38	0.779 ENTER	X =_	
39	ENTER	X =_	Mean value and standard deviation printouts of population 1
40	1.033 ENTER	X =_	
41	1.217 ENTER	X =_	
⋮	⋮	⋮	
47	0.840 ENTER	X =_	
48	ENTER	>	

[Printout]

Test of mean value difference (processed data)

```
T=-1.076244005
PHI= 12
```

Tast of variance ratio (processed data)

```
F= 1.194715643
PHI1= 7
PHI2= 9
```

Test of mean value difference (inprocessed data)

```
MEAN= 2.075
STD.DEV.=
   2.375469878E-01

MEAN= 2.2
STD.DEV.=
   1.788854382E-01

T=-1.076244005
PHI= 12
```

Test of variance ratio (inprocessed data)

```
MEAN= 1.3253
STD.DEV.=
   3.261141756E-01

MEAN= 1.163375
STD.DEV.=
   3.564527368E-01

F= 1.194715649
PHI1= 7
PHI2= 9
```

PROGRAM TITLE	TEST OF MEAN VALUE DIFFERENCE AND VARIANCE RATIO	PROGRAM NO. P5–B–7	5

[Program List]

```
 10:"A":CLEAR :
    INPUT "N1=";M,
    "N2=";N
 20:INPUT "MEAN 1=
    ";Y,"MEAN 2=";
    X
 30:INPUT "STD.DEV
    . 1=";R,"STD.D
    EV. 2=";S
 40:R=R*R*(M-1)
 50:S=S*S*(N-1):
    GOTO 80
 60:"B":GOSUB 500
 70:M=N:R=S:Y=X:
    GOSUB 500
 80:L=M+N
 90:T=(Y-X)*√(M*N*
    (L-2)/(L*(R+S)
    ))
100:LPRINT "T=";T:
    LPRINT "PHI=";
    L-2
110:END
120:"X":INPUT "N1=
    ";M,"N2=";N
130:INPUT "STD.DEV
    . 1=";R,"STD.D
    EV. 2=";S
140:R=R*R:S=S*S:
    GOTO 180
150:"Z":GOSUB 500
160:M=N:R=S:GOSUB
    500
170:R=R/(M-1):S=S/
    (N-1)
180:IF S>RLET Z=M:
    M=N:N=Z:Z=S:S=
    R:R=Z
190:LPRINT "F=";R/
    S
200:LPRINT "PHI1="
    ;M-1
210:LPRINT "PHI2="
    ;N-1
220:END
500:N=0:T=0:S=0
510:INPUT "X=";X:
    GOTO 530
520:GOTO 550
530:N=N+1:T=T+X
540:S=S+X*X:GOTO 5
    10
550:X=T/N:S=S-N*X*
    X
560:CLS :LPRINT "M
    EAN=";X
570:LPRINT "STD.DE
    V.=",√(S/(N-1)
    )
580:LF 1:RETURN
```

STATUS 1

611

[Memory Contents]

A	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	M + N
M	No. of data for population 1
N	No. of data for population 2
O	
P	
Q	
R	Standard deviation of population 1
S	Standard deviation of population 2
T	Test value
U	
V	
W	
X	Mean value of population 2
Y	Mean value of population 1
Z	√

SHARP

PROGRAM TITLE	ONE-WAY LAYOUT	PROGRAM NO. P5–B–9	1
		CE–150 required	

[Outline]

This program performs analysis of variance using the one-way layout method.

[Operating Guide]

Input: 1. Input the number of levels of the factors.
When "a=" appears, key-in the number of levels.
2. Input the number of replications.
When "n=" appears, key-in the number of replications.
3. Data input
i = 1,2 . . . . . a
j = 1,2 . . . . . n
When "X (i, j) = "appears" key-in the data.

Output: Results of analysis of variance.
Outputs of square sums, freedom degree, unbiased variance and unbiased variance ratio in between or inside classes.

[Example]

Factor	A1	A2	A3	A4
1	25.5	25.5	27.5	28.0
2	26.5	24.5	25.5	29.5
3	27.0	23.5	26.5	28.5

[Contents] (Formulas)

No. of levels : a
No. of replications : n

Data: x_{ij} (i = 1~a, j=1~n) No. of data: a n

1. $[X] = x^2 / an$
   $[A] = x^2_i / n$
   $[AS] = \Sigma x^2_{ij}$

2. $S_A = [A] - [X]$
   $S_T = [AS] - [X]$
   $S_E = [AS] - [A]$

3. $\phi_A = a - 1$
   $\phi_E = a_n - a$
   $\phi_T = a_n - 1$

4. $\{V\} = \{S\} / \{\phi\}$

5. $F = V_A / V_E$

V_T is not calculated.

PROGRAM TITLE	ONE-WAY LAYOUT	PROGRAM NO. P5–B–9	2

[Printout]

```
Sa= 26.166667
Se= 6.333333
St= 32.5
DFa= 3
DFe= 8
DFt= 11
Va= 8.722222333
Ve= 0.791666625
Fa= 11.01754458
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	a=_	No. of factors
2	4 ENTER	n =_	No. of replications
3	3 ENTER	x (1, 1) = ?	Data
4	25.5 ENTER	x (1, 2) = ?	Repeat for data inputs
⋮	⋮	⋮	
14	29.5 ENTER	x (4, 3) = ?	
15	28.5 ENTER	>	Printout

PROGRAM TITLE	ONE-WAY LAYOUT	PROGRAM NO. P5–B–9	3

[Program List]

```
 10:"A":CLEAR :CLS
    :WAIT 0
 20:INPUT "a=";A:
    INPUT "n=";N
 70:E=0:L=0
 75:B$=STR$ (I+1)+
    ",";STR$ (L+1)
 76:A$="x("+B$+")=
    "
 77:PRINT A$;
 80:INPUT D:CLS
 90:E=E+D:Z=Z+D*D
100:IF L<>N-1LET L
    =L+1:      5
210:S=S+E*E
220:R=R+E
230:IF I<>A-1LET J
    =I+1:GOTO 70
240:R=R*R/(A*N)
250:S=S/N
430:S=S-R:LPRINT "
    Sa=";S
510:Z=Z-R
520:P=Z-S:LPRINT "
    Se=";P:LPRINT
    "St=";Z
530:F=A-1:LPRINT "
    DFa=";F
540:S=S/F
690:O=A*(N-1):
    LPRINT "DFe=";
    O
700:P=P/O
710:O=A*N-1:LPRINT
    "DFt=";O
720:LPRINT "Va=";S
800:LPRINT "Ve=";P
810:F=S/P:LPRINT "
    Fa=";F
890:END
```

STATUS 1

415

[Memory Contents]

A	a (No. of factors)
B	
C	
D	Input
E	Σxij
F	ϕa, Fa
G	
H	
I	√
J	
K	
L	√
M	
N	No. of replications
O	ϕe, ϕt
P	√
Q	
R	x^2. /(an)
S	Σxi^2/n Sa, Va
T	
U	
V	
W	
X	
Y	
Z	Σx^2ij ·St
A$	Input message
B$	Input message

SHARP

PROGRAM TITLE	TWO-WAY LAYOUT (WITH NO REPLICATIONS)	PROGRAM NO. P5–B–10	1

CE–150 required

[Outline]

This program executes the analysis of variances under two-way layout method with no replications.

[Operating Guide]

Input: 1. Program start
Program starts with the DEF A key pressed.

2. Factor dimension input
Enter the dimensions of factor A (number of A levels) with "a=".
Enter the dimensions of factor B (number of B levels) with "b=".

3. Data input
i = 1 ~ a, j = 1 ~ b
Enter the data with "x(i,j)=".

Output: Variations, freedom degrees, unbiased variances, and unbiased variance ratios.

[Example]

A \ B	B1	B2	B3	B4
A1	−15	−11	−29	3
A2	−11	−9	−3	−7
A3	−7	−1	7	19
A4	9	41	21	48

[Contents] (Formulas)

Number of levels of each factor: a, b
Data: x_{ij} (i=1~a, j=1~b) No. of data: ab

1. $[X] = x^2.. / ab$
   $[A] = \Sigma x^2 i. / b$
   $[B] = \Sigma x^2 . j / a$
   $[ABS] = \Sigma x^2 ij$

2. $S_A = [A] - [X]$
   $S_B = [B] - [X]$
   $S_T = [ABS] - [X]$
   $S_E = [ABS] - [A] - [B]$

3. $\phi_A = a - 1$
   $\phi_A = b - 1$
   $\phi_E = ab - a - b - 1$
   $\phi_T = ab - 1$

4. $\{V\} = \{S\} / \{\phi\}$
   V_T is not calculated.

5. $\{F\} = \{V\} / \{V_E\}$
   F_T and F_E are not calculated.

PROGRAM TITLE	TWO-WAY LAYOUT (WITH NO REPLICATIONS)	PROGRAM NO. P5–B–10	2

[Printout]

```
Sa= 4333.1875
Sb= 1051.1875
Se= 849.5625
St= 6233.9375
DFa= 3
DFb= 3
DFe= 9
DFt= 15
Va= 1444.395833
Vb= 350.3958333
Ve= 94.39583333
Fa= 15.3014787
Fb= 3.711984109
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	a =_	Dimensions of factor A
2	4 ENTER	b =_	Dimensions of factor B
3	4 ENTER	X(1, 1) = ?	Data
4	−15 ENTER	X(1, 2) = ?	Repeated data input
⋮	⋮	⋮	
18	21 ENTER	X(4, 4) = ?	
19	48 ENTER	>	

PROGRAM TITLE	TWO-WAY LAYOUT (WITH NO REPLICATIONS)	PROGRAM NO. P5– B–10	3

[Program List]

```
 10:"A":CLEAR :CLS
    :WAIT 0
 20:INPUT "a=";A:
    INPUT "b=";B
 40:DIM O(B-1)
 50:P=0:J=0
 75:B$=STR$ (I+1)+
    ","+STR$ (J+1)
 76:A$="x("+B$+")=
    "
 77:PRINT A$;
 80:INPUT E:CLS
170:Z=Z+E*E
180:O(J)=O(J)+E
190:P=P+E
200:IF J<>B-1LET J
    =J+1:GOTO 75
210:S=S+P*P
220:R=R+P
230:IF I<>A-1LET I
    =I+1:GOTO 50
240:R=R*R/(A*B)
250:S=S/B
260:FOR I=0TO B-1
270:T=T+O(I)*O(I):
    NEXT I
300:T=T/A
430:S=S-R:LPRINT "
    Sa=";S
440:T=T-R:LPRINT "
    Sb=";T
510:Z=Z-R
520:P=Z-S-T:LPRINT
    "Se=";P:LPRINT
    "St=";Z
530:F=A-1:LPRINT "
    DFa=";F
540:S=S/F
550:G=B-1:LPRINT "
    DFb=";G
560:T=T/G
690:O=(A-1)*(B-1):
    LPRINT "DFe=";
    O
700:P=P/O
710:O=A*B-1:LPRINT
    "DFt=";O
720:LPRINT "Va=";S
730:LPRINT "Vb=";T
800:LPRINT "Ve=";P
810:F=S/P:LPRINT "
    Fa=";F
820:F=T/P:LPRINT "
    Fb=";F
890:END

STATUS 1
                575
```

[Memory Contents]

A	a (No. of factors)
B	b (No. of factors)
C	
D	
E	For input
F	ϕ_a Fa Fb
G	ϕ_b
H	
I	√
J	√
K	
L	
M	
N	
O	ϕe ϕx
P	$\Sigma x i$. Se Ve
Q	
R	$\Sigma x^2 i$ ./a b
S	$\Sigma x^2 i$ /b Sa Va
T	$\Sigma O(I)^2$/a Sb Vb
U	
V	
W	
X	
Y	
Z	$\Sigma x^2 ij$ St
A$	Input message
B$	Input message
O(B–1)	Calculation of $\Sigma x^2 \cdot j$

SHARP

PROGRAM TITLE	THREE-WAY LAYOUT (WITH NO REPLICATIONS)	PROGRAM NO. P5–B–12	1
		CE–150 required	

[Outline]

With this program, analyses of variances can be done by the three-way layout method with no replications.

[Operating Guide]

Input: 1. Program start
Press the [DEF] [A] keys to start Program.

2. Factor dimension input
With "a=", enter the dimension of factor A (number of A levels).
With "b=", enter the dimension of factor B (number of B levels).
With "c=", enter the dimension of factor C (number of C levels).

3. Data input
i=1~a, j=1~b, k=1~c
(These are determined by the input values in step 2.)
With "X(i,j,k)=", enter the data.

Output: Variations, freedom degrees, unbiased variances, and unbiased variance ratios.

[Example]

Day	Experimentalist	Thermometer C1	C2	C3	C4	Sum
A1	B1	2.0	1.0	–0.5	1.5	
	B2	1.0	0.0	–1.0	–1.0	7.0
	B3	1.5	1.0	1.0	0.5	
A2	B1	1.5	1.5	0.5	1.5	
	B2	1.0	1.0	0.0	0.0	11.5
	B3	1.0	1.5	1.0	1.0	
	Sum	8.0	6.0	1.0	3.5	18.5

PROGRAM TITLE	THREE-WAY LAYOUT (WITH NO REPLICATIONS)	PROGRAM NO. P5-B-12	2

[Contents] (Formulas)

Numbers of levels of factors a, b and c
Data: $\{X_{ijk}\}$ (i=1~a, j=1~b, k=1~c) No. of data: abc

1. $[X] = x^2 \cdots / abc$
   $[A] = \sum x^2{}_{i\cdot\cdot} / bc$
   $[B] = \sum x^2{}_{\cdot j\cdot} / ac$
   $[C] = \sum x^2{}_{\cdot\cdot k} / ab$
   $[AB] = \sum x^2{}_{ij\cdot} / c$
   $[BC] = \sum x^2{}_{\cdot jk} / a$
   $[AC] = \sum x^2{}_{i\cdot k} / b$
   $[ABCS] = \sum x^2{}_{ijk}$

2. $S_A = [A] - [X]$
   $S_B = [B] - [X]$
   $S_C = [C] - [X]$
   $S_{A\times B} = [AB] - [X] - S_A - S_B$
   $A_{B\times C} = [BC] - [X] - S_B - S_C$
   $S_{A\times C} = [AC] - [X] - S_A - S_C$
   $S_T = [ABCS] - [X]$
   $S_E = S_T - S_A - S_B - S_C - S_{A\times B} - S_{A\times C} - S_{B\times C}$

3. $\phi_A = a - 1$
   $\phi_B = b - 1$
   $\phi_C = C - 1$
   $\phi_{A\times B} = \phi_A \phi_B$
   $\phi_{A\times C} = \phi_A \phi_C$
   $\phi_{B\times C} = \phi_B \phi_C$
   $\phi_E = \phi_A \phi_B \phi_C$
   $\phi_T = abc - 1$

4. $\{V\} = \{S\} / \{\phi\}$
   V_T is not calculated.

5. $\{F\} = \{V\} / \{V_E\}$
   F_T and F_E are not calculated.

[Printout]

```
Sa= 0.84375
Sb= 5.02083333
Sc= 4.61458333
Sa*b= 0.4375
Sa*c= 1.03125
Sb*c= 2.72916667
Se= 0.3125
St= 14.98958333
DFa= 1
DFb= 2
DFc= 3
DFa*b= 2
DFa*c= 3
DFb*c= 6
DFe= 6
DFt= 23
```

```
Va= 0.84375
Vb= 2.510416665
Vc= 1.538194443
Va*b= 0.21875
Va*c= 0.34375
Vb*c= 4.548611117E
-01
Ve= 5.208333333E-0
2
Fa= 16.2
Fb= 48.19999997
Fc= 29.53333331
Fa*b= 4.2
Fa*c= 6.6
Fb*c= 8.733333345
```

PROGRAM TITLE	THREE-WAY LAYOUT (WITH NO REPLICATIONS)	PROGRAM NO. P5–B–12	3

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	$a = \_$	Dimension of factor A
2	2 ENTER	$b = \_$	Dimension of factor B
3	3 ENTER	$c = \_$	Dimension of factor C
4	4 ENTER	$X(1, 1, 1) = ?$	
5	2.0 ENTER	$X(1, 1, 2) = ?$	Repeated data input.
⋮	⋮	⋮	⋮
27	1.0 ENTER	$X(2, 3, 4) = ?$	
28	1.0 ENTER	>	Printout

[Program List]

```
10:"A":CLEAR :CLS
   :WAIT 0
20:INPUT "a=";A:
   INPUT "b=";B
30:INPUT "c=";C
40:DIM F(B-1,C-1)
   ,G(A-1,C-1),Q(
   C-1),O(B-1)
50:P=0:J=0
60:H=0:K=0
75:B$=STR$ (I+1)+
   ",","+STR$ (J+1)
   +",","+STR$ (K+1
   )
76:A$="x("+B$+")="
   "
77:PRINT A$;
80:INPUT E:CLS
110:Z=Z+E*E
120:F(J,K)=F(J,K)+
   E
130:G(I,K)=G(I,K)+
   E
140:Q(K)=Q(K)+E
150:H=H+E
160:IF K<>C-1LET K
   =K+1:GOTO 75
170:V=V+H*H
180:O(J)=O(J)+H
190:P=P+H
200:IF J<>B-1LET J
   =J+1:GOTO 60
210:S=S+P*P
220:R=R+P
230:IF I<>A-1LET I
   =I+1:GOTO 50
240:R=R*R/(A*B*C)
250:S=S/(B*C)
260:FOR I=0TO B-1
270:T=T+O(I)*O(I):
   NEXT I
280:FOR I=0TO C-1
290:U=U+Q(I)*Q(I):
   NEXT I
300:T=T/(A*C):U=U/
   (A*B)
310:U=U/C
320:FOR I=0TO A-1
330:FOR J=0TO C-1
340:W=W+G(I,J)*G(I
   ,J)
350:NEXT J:NEXT I
360:FOR I=0TO B-1
370:FOR J=0TO C-1
380:X=X+F(I,J)*F(I
   ,J)
390:NEXT J:NEXT I
400:W=W/B:X=X/A
430:S=S-R:LPRINT "
   Sa=";S
440:T=T-R:LPRINT "
   Sb=";T
450:U=U-R:LPRINT "
   Sc=";U
460:V=V-R-S-T:
   LPRINT "Sa*b="
   ;V
470:W=W-R-S-U:
   LPRINT "Sa*c="
   ;W
480:X=X-R-T-U:
   LPRINT "Sb*c="
   ;X
510:Z=Z-R
520:P=Z-S-T-U-V-W-
   X:LPRINT "Se="
   ;P:LPRINT "St=
   ";Z
530:F=A-1:LPRINT "
   DFa=";F
540:S=S/F
550:G=B-1:LPRINT "
   DFb=";G
560:T=T/G
570:H=C-1:LPRINT "
   DFc=";H
580:U=U/H
590:O=F*G:LPRINT "
   DFa*b=";O
600:V=V/O
610:O=F*H:LPRINT "
   DFa*c=";O
620:W=W/O
630:O=G*H:LPRINT "
   DFb*c=";O
640:X=X/O
690:O=F*G*H:LPRINT
   "DFe=";O
700:P=P/O
710:O=A*B*C-1:
   LPRINT "DFt=";
   Q
720:LPRINT "Va=";S
730:LPRINT "Vb=";T
740:LPRINT "Vc=";U
750:LPRINT "Va*b="
   ;V
760:LPRINT "Va*c="
   ;W
770:LPRINT "Vb*c="
   ;X
800:LPRINT "Ve=";P
810:F=S/P:LPRINT "
   Fa=";F
820:F=T/P:LPRINT "
   Fb=";F
830:F=U/P:LPRINT "
   Fc=";F
840:F=V/P:LPRINT "
   Fa*b=";F
850:F=W/P:LPRINT "
   Fa*c=";F
860:F=X/P:LPRINT "
   Fb*c=";F
890:END

STATUS 1
                1295
```

PROGRAM TITLE	THREE-WAY LAYOUT (WITH NO REPLICATIONS)	PROGRAM NO. P5–B–12	4

[Memory Contents]

A	a(dimension of factor A)	A$	Input message		
B	b(dimension of factor B)	B$	Input message		
C	c(dimension of factor C)	C$			
D		D$			
E	For input	E$			
F	ϕ_a Fa~Fbc	F$		F(B-1, C-1)	Calculation of Σx^2.jk
G	ϕ_b	G$		G(A-1, C-1)	Calculation of Σx^2i.k
H	Σx_{ij} ϕ_c	H$			
I	√	I$			
J	√	J$			
K	√	K$			
L		L$			
M		M$			
N		N$			
O	$\phi_{a\times b}, \phi_{a\times c}, \phi_{b\times c}, \phi_e, \phi_t$	O$		O(B-1)	Calculation of $\Sigma x^2 \cdot k$.
P	Σxi.. Se Ve	P$			
Q		Q$		Q(C-1)	Calculation of $\Sigma x^2 \cdot\cdot k$
R	[X]	R$			
S	[A] , S_A V_A	S$			
T	T:$\Sigma O(i)^2/_{ac}$, Sb, Vb	T$			
U	U:$\Sigma Q(i)^2/_{ab}$, Sc , Vc	U$			
V	V:[A B], Sa×b, Va×b	V$			
W	W:$\Sigma G(i,j)^2/_b$. Sa×c, Va×c	W$			
X	X:$\Sigma F(i,j)^2/_a$. Sb×c, eV b×c	X$			
Y		Y$			
Z	Σx^2ijk , St	Z$			

SHARP

PROGRAM TITLE	$\overline{X}$ – R CONTROL CHART	PROGRAM NO. P5– B–14	1

CE–150, CE–151 and CTR required

[Outline]

Based on data, the control limit is determined to generate an $\overline{X}$–R control chart. This program also enables outputs of $\overline{X}$ (mean) and R (range) for each group of data.

[Operating Guide]

DEF A : For data input

DEF B : Used to modify and check data, as well as finding $\overline{X}$ (mean) and R (range).

DEF C : For setting coefficients on a table for $\overline{X}$–R control limit calculation, as well as enabling outputs of a central line, upper control limit, and lower control limit.

DEF F : For $\overline{X}$–R control chart generation.

[Contents] (Formulas)

1. The mean value $\overline{x}$ for each group is calculated.

$$\overline{x} = \frac{\text{Total data for each group}}{\text{No. of data}}$$

2. Range R is calculated.
   R = Max. value of each group – Min. value of each group

3. The total mean value $\overline{\overline{x}}$ is calculated.

$$\overline{\overline{x}} = \frac{\text{Grand total of mean value}}{\text{No. of groups}}$$

4. The total range R is calculated.

$$\overline{R} = \frac{\text{Grand total of range R}}{\text{No. of groups}}$$

5. Control lines of $\overline{x}$ control chart.
   Central line CL = $\overline{\overline{x}}$
   Upper control limit UCL = $\overline{\overline{x}} + A_2\overline{R}$
   Lower control limit LCL = $\overline{\overline{x}} - A_2\overline{R}$
   A_2 = coefficient

6. Control lines of $\overline{R}$ control chart.
   Central line CL = $\overline{R}$
   Upper control limit UCL = $D_4\overline{R}$
   Lower control limit LCL = $D_3\overline{R}$ ($D_3 = 2 - D_4$)
   D_3 and D_4 = coefficients

7. File name (on cassette tape): "X–R DATA".

Table III-3 Coefficients for control limit calculation

Group Size n	A_2	D_3	D_4
2	1.880	0	3.268
3	1.023	0	2.574
4	0.729	0	2.288
5	0.577	0	2.114
6	0.483	0	2.004
7	0.419	0.076	1.924
8	0.373	0.136	1.864
9	0.337	0.184	1.816
10	0.308	0.223	1.777

[Example]

1. $\overline{X}-R$ control chart is generated from the next data sheet.

Group No.	Measured values				
	x_1	x_2	x_3	x_4	x_5
1	4	6	6	6	5
2	5	5	5	9	4
3	8	10	13	9	5
4	10	8	2	3	2
5	5	3	4	4	4
6	3	3	4	4	2
7	4	8	11	10	12
8	8	3	12	12	10
9	4	4	5	3	3
10	5	3	4	8	5
11	3	12	12	13	5
12	5	5	13	10	5
13	4	11	4	3	4
14	3	3	3	3	10
15	11	6	10	5	12
16	8	8	5	6	5
17	3	4	4	3	4
18	3	3	3	3	3
19	8	12	8	10	7
20	4	8	4	3	4

PROGRAM TITLE	$\bar{X}$ – R CONTROL CHART	PROGRAM NO. P5– B–14	3

[Printout]

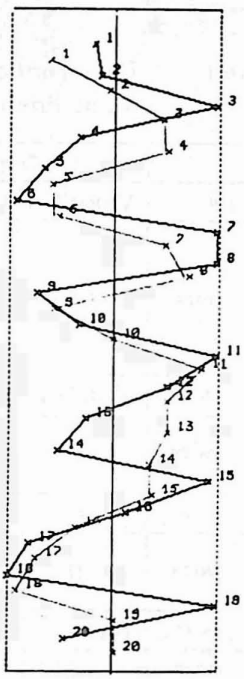
```
*GROUP= 1
  1  4
  2  6
  3  6
  4  6
  5  5
AVL  5.4
 R   2

*GROUP= 2
  1  5
  2  5
  3  5
  4  9
  5  4
AVL  5.6
 R   5

*GROUP= 3
  1  8
  2  10
  3  13
  4  9
  5  5
AVL  9
 R   8

*GROUP= 4
  1  10
  2  8
  3  2
  4  3
  5  2
AVL  5
 R   8

*GROUP= 5
  1  5
  2  3
  3  4
  4  4
  5  4
AVL  4
 R   2

*GROUP= 6
  1  3
  2  3
  3  4
  4  4
  5  2
AVL  3.2
 R   2

*GROUP= 7
  1  4
  2  8
  3  11
  4  10
  5  12
AVL  9
 R   8

*GROUP= 8
  1  8
  2  3
  3  12
  4  12
  5  10
AVL  9
 R   9

*GROUP= 9
  1  4
  2  4
  3  5
  4  3
  5  3
AVL  3.8
 R   2

*GROUP= 10
  1  5
  2  3
  3  4
  4  8
  5  5
AVL  5
 R   5

*GROUP= 11
  1  3
  2  12
  3  12
  4  13
  5  5
AVL  9
 R   10

*GROUP= 12
  1  5
  2  5
  3  13
  4  10
  5  5
AVL  7.6
 R   8

*GROUP= 13
  1  4
  2  11
  3  4
  4  3
  5  4
AVL  5.2
 R   8

*GROUP= 14
  1  3
  2  3
  3  3
  4  3
  5  10
AVL  4.4
 R   7

*GROUP= 15
  1  11
  2  6
  3  10
  4  5
  5  12
AVL  8.8
 R   7

*GROUP= 16
  1  8
  2  8
  3  5
  4  6
  5  5
AVL  6.4
 R   3

*GROUP= 17
  1  3
  2  4
  3  4
  4  3
  5  4
AVL  3.6
 R   1

*GROUP= 18
  1  3
  2  3
  3  3
  4  3
  5  3
AVL  3
 R   0

*GROUP= 19
  1  8
  2  12
  3  8
  4  10
  5  7
AVL  9
 R   5

*GROUP= 20
  1  4
  2  8
  3  4
  4  3
  5  4
AVL  4.6
 R   5
```

PROGRAM TITLE	$\bar{X}$ – R CONTROL CHART	PROGRAM NO. P5–B–14	4

[Printout]

```
** X **
LCL= 3.00075
CL=  6.03
UCL= 9.05925

** R **
LCL= 0
CL=  5.25
UCL= 11.10375
```

```
X CONTROL CHART
R CONTROL CHART

LCL      CL      UCL
```



The real print out is colored.
Refer to page 1 .

[Key Operation Procedure] : $\bar{X}$–R Control Chart Data Input

Step No.	Input	Display	Remarks
1	DEF A	DATA CLOAD ? (Y, N) _	
2	Y ENTER	>	Processing ends after data input from casette tape.
	N ENTER	NO. OF DATA = _	
3	5 ENTER	NO. OF GROUPS = _	
4	20 ENTER	GROUP 1, DATA = _	
5	5 ENTER	GROUP 1, DATA = _	
⋮	⋮	⋮	Repeated data input.

Step No.	Input	Display	Remarks
25	4 [ENTER]	DATA CSAVE ? (Y, N) _	
26	Y [ENTER]	>	Processing ends after data output to cassette tape.
	N [ENTER]	>	With this key pressed, processing completes.

[Key Operation Procedure] : Data Verification and Correction, Control Limit Value Printout and $\bar{X}$–R Control Chart Printout

Step No.	Input	Display	Remarks
1	[DEF] [B]	VERIFY, CORR.? (V, C)_	
2	V [ENTER]	DATA CSAVE? (Y, N) _	→ Step 6 After the verification list output, this display appears.
	C [ENTER]	* GROUP = _	→ Step 3
	[ENTER]	>	With only [ENTER] key pressed, processing ends.
3	1 [ENTER]	NO. = _	→ Step 4
	[ENTER]	VERIFY, CORR.? (V, C)_	→ Step 2 Totalization and display
4	1 [ENTER]	DATA = _	→ Step 5
	[ENTER]	* GROUP = _	→ Step 3
5	4 [ENTER]	NO. = _	→ Step 4
6	Y [ENTER]	>	Processing ends after data is output to cassette tape.
	N [ENTER]	>	Processing is completed.
7	[DEF] [C]	>	Processing ends with CL, UCL and LCL printouts.
8	[DEF] [F]	>	Processing ends with $\bar{X}$–R control chart printout.

PROGRAM TITLE	$\bar{X}$ – R CONTROL CHART	PROGRAM NO. P5–B–14	6

[Program List]

```
10:"A":CLEAR :
   WAIT :DIM Y(8,
   1)
12:INPUT "DATA CL
   OAD?(Y,N)";A$
14:IF (A$="Y")+(A
   $="N")<>1GOTO
   12
15:WAIT 0
16:IF A$="Y"GOTO
   132
20:INPUT "NO. OF
   DATA =";M
30:IF (2<=M)+(M<=
   10)<>2GOTO 20
50:INPUT "N0. OF
   GROUPS =";N:
   DIM X(M+1,N-1)
60:FOR A=0TO N-1
65:Z1=-10^8:Z2=10
   ^8
70:FOR B=0TO M-1
80:CLS :A$="GROUP
   ="+STR$ (A+1)
   +",DATA="
90:CLS :PRINT A$;
100:INPUT X(B,A)
105:X(M,A)=X(M,A)+
   X(B,A)
107:IF Z1<X(B,A)
   LET Z1=X(B,A)
108:IF Z2>X(B,A)
   LET Z2=X(B,A)
110:NEXT B:CLS
115:X(M,A)=X(M,A)/
   M
117:X(M+1,A)=Z1-Z2
120:NEXT A
122:WAIT :INPUT "D
   ATA CSAVE?(Y,N
   )";A$
123:IF (A$="Y")+(A
   $="N")<>1GOTO
   122
124:WAIT 0:IF A$="
   N"GOTO 130
126:PRINT #"X-R DA
   TA";M,N
128:PRINT #"X-R DA
   TA";X(*)
130:CLS :END
132;CLEAR ;WAIT :
   INPUT #"X-R DA
   TA";M,N
134:DIM X(M+1,N-1)
   ,Y(8,1)
136:INPUT #"X-R DA
   TA";X(*):END
140:"B":WAIT 0:
   INPUT "VERIFY,
   CORR.? (V,C) "
   ;N$:GOTO 160
150:END
160:IF (N$="V")+(N
   $="C")<>1GOTO
   140
170:IF N$="V"GOTO
   270
180:INPUT "*GROUP=
   ";A:GOTO 200
190:GOTO 250
200:IF (1<=A)+(A<=
   N)<>2GOTO 180
220:INPUT "No.=";B
   :GOTO 240
230:GOTO 180
240:IF (1<=B)+(B<=
   M)<>2GOTO 220
245:INPUT "DATA=";
   X(B-1,A-1):
   GOTO 220
250:FOR A=0TO N-1
252:X(M,A)=0:Z1=-1
   0^8:Z2=10^8
254:FOR B=0TO M-1
256:IF Z1<X(B,A)
   LET Z1=X(B,A)
257:IF Z2>X(B,A)
   LET Z2=X(B,A)
258:X(M,A)=X(M,A)+
   X(B,A)
259:NEXT B:X(M,A)=
   X(M,A)/M
260:X(M+1,A)=Z1-Z2
   :NEXT A:GOTO 1
   40
262:WAIT :INPUT "D
   ATA CSAVE?(Y,N
   )";A$
263:IF (A$="Y")+(A
   $="N")<>1GOTO
   262
264:WAIT 0:IF A$="
   N"END
265:PRINT #"X-R DA
   TA";M,N
266:PRINT #"X-R DA
   TA";X(*):END
270:FOR A=0TO N-1
280:LF 1:LPRINT "*
   GROUP=";A+1
290:FOR B=0TO M-1
300:LPRINT USING "
   ###";B+1;
305:USING :LPRINT
   X(B,A)
310:NEXT B
315:LPRINT "AVL";X
   (M,A)
317:LPRINT " R ";X
   (M+1,A)
320:NEXT A
330:GOTO 262
340:"C":Y(0,0)=1.8
   80:Y(0,1)=3.26
   7
350:Y(1,0)=1.023:Y
   (1,1)=2.575:Y(
   2,0)=0.729:Y(2
   ,1)=2.282
360:Y(3,0)=0.577:Y
   (3,1)=2.115:Y(
   4,0)=0.483:Y(4
   ,1)=2.004
370:Y(5,0)=0.419:Y
   (5,1)=1.924:Y(
   6,0)=0.373:Y(6
   ,1)=1.864
380:Y(7,0)=0.337:Y
   (7,1)=1.816:Y(
   8,0)=0.308:Y(8
   ,1)=1.777
390:LF 1
400:H=0:P=0
410:FOR A=0TO N-1
420:G=X(0,A):L=X(0
   ,A):S=X(0,A)
430:FOR B=1TO M-1
440:G=G+X(B,A)
450:IF L<X(B,A)LET
   L=X(B,A)
460:IF S>X(B,A)LET
   S=X(B,A)
470:NEXT B
```

(To be continued)

PROGRAM TITLE	$\bar{X}$ – R CONTROL CHART	PROGRAM NO. P5– B–14	7

[Program List]

```
480:R=L-S:G=G/M:H=
    H+G:P=P+R
490:NEXT A
500:U=P/N:T=H/N
510:LPRINT "** X *
    *"
520:D=2-Y(M-2,1):D
    1=D:IF D<0LET
    D=0
530:XC=T:XL=T-(Y(M
    -2,0)*U):XU=T+
    (Y(M-2,0)*U)
550:LPRINT "LCL=";
    XL
552:LPRINT "CL= ";
    XC
554:LPRINT "UCL=";
    XU
556:LF 1:LPRINT "*
    * R **"
560:RC=U:RL=D*U:RU
    =Y(M-2,1)*U
570:LPRINT "LCL=";
    RL
572:LPRINT "CL= ";
    RC
574:LPRINT "UCL=";
    RU
580:END
590:"F":LF 2:COLOR
    3:LPRINT "X CO
    NTROL CHART"
600:COLOR 2:LPRINT
    "R CONTROL CHA
    RT"
610:GRAPH
620:COLOR 0:ROTATE
    0
630:GLCURSOR (10,-
    50):LPRINT "LC
    L"
640:GLCURSOR (90,-
    50):LPRINT "CL
    "
650:GLCURSOR (160,
    -50):LPRINT "U
    CL"
660:GLCURSOR (100,
    -80):SORGN
670:LINE (-75,0)-(
    75,0),0,1
680:LINE (75,0)-(7
    5,-480),1,1
690:LINE (75,-480)
    -(-75,-480),0,
    1
700:LINE (-75,-480
    )-(-75,0),1,1
710:LINE (0,0)-(0,
    -480),0,1
720:F=450/N:DX=Y(M
    -2,0)*U/75:
    COLOR 3
730:X1=X(M,0)
770:Y1=-F
775:COLOR 3:T1=XC
780:FOR B=1TO N
790:IF B=NGOTO 840
800:X2=X(M,B):Y2=Y
    1-F
840:GOSUB 6000
850:X1=X2:Y1=Y2
860:NEXT B
870:COLOR 2
910:X1=X(M+1,0):Y1
    =-F
920:DA=0:IF X1>RC
    LET DA=(RU-RC)
    /75:GOTO 940
930:IF X1<RCLET DA
    =(RC-D1)/75
940:T1=RC
950:FOR B=1TO N
960:IF B=NGOTO 101
    0
970:X2=X(M+1,B):Y2
    =Y1-F
980:DB=0:IF X2>RC
    LET DB=(RU-RC)
    /75:GOTO 1010
990:IF X2<RCLET DB
    =(RC-D1)/75
1010:GOSUB 7000
1020:X1=X2:Y1=Y2:
     DA=DB
1030:NEXT B
1040:GLCURSOR (0,
     -500):CSIZE
     2:COLOR 0:
     TEXT :END
6000:X3=(X1-T1)/D
     X:LINE (X3-2
     ,Y1+2)-(X3+2
     ,Y1-2)
6010:LINE (X3-2,Y
     1-2)-(X3+2,Y
     1+2)
6020:CSIZE 1:
     LPRINT B
6030:IF B=NRETURN
6040:X4=(X2-T1)/D
     X:LINE (X3,Y
     1)-(X4,Y2)
6050:RETURN
7000:X3=(X1-T1)/D
     A:LINE (X3-2
     ,Y1+2)-(X3+2
     ,Y1+2)
7010:LINE (X3-2,Y
     1-2)-(X3+2,Y
     1+2)
7020:CSIZE 1:
     LPRINT B
7030:IF B=NRETURN
7040:X4=(X2-T1)/D
     B:LINE (X3,Y
     1)-(X4,Y2)
7050:RETURN
```

```
STATUS 1
                    3010
```

PROGRAM TITLE	$\overline{X}$ – R CONTROL CHART	PROGRAM NO. P5– B–14	8

[Memory Contents]

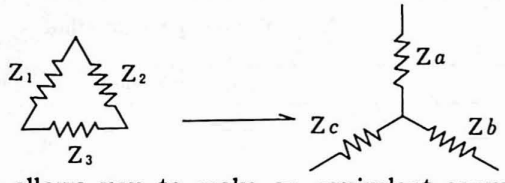
A	Counter for No. of groups	A$	√	X(B, A)	Input data
B	Counter for No. of data	B$		Y(8, 1)	Factor value
C		C$		XL	$\overline{X}$ lower control limit
D	LCL coefficient of R	D$		XC	$\overline{X}$ central line
E		E$		XU	$\overline{X}$ upper control limit
F	Y-coordinate graph factor	F$		RL	R lower control limit
G	Group mean value	G$		RC	R central line
H	Total of each mean values	H$		RU	R upper control limit
I		I$		X1	X-coordinate 1
J		J$		X2	X-coordinate 2
K		K$		X3	X-coordinate 3
L	Group max. value	L$		X4	X-coordinate 4
M	No. of data	M$		Y1	Y-coordinate 1
N	No. of groups	N$		Y2	Y-coordinate 2
O		O$		DA	X-coordinate graph factor
P	Total range	P$		DB	√
Q		Q$		T1	RC
R	Group range	R$		D1	D
S	Group min. value	S$		Z1	√
T	Total mean value	T$		Z2	√
U	Grand total range	U$		DX	√
V		V$			
W		W$			
X		X$			
Y		Y$			
Z		Z$			

SHARP

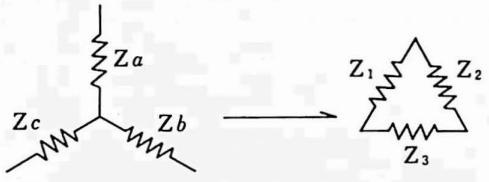
PROGRAM TITLE	Δ ↔ Y CONVERSION	PROGRAM NO. P5–C–1	1

[Outline]

This program allows you to make an equivalent conversion from the impedance of Δ connection to that of Y connection.



Also allows you to make an equivalent conversion from the impedance of Y connection to that of Δ connection.



[Operating Guide]

Refer to the key Operation Procedure.

[Example]

1). Δ → Y Conversion

$$\begin{cases} R_1 = 5 \\ x_1 = 3 \end{cases} \quad \begin{cases} R_2 = 6 \\ x_2 = -2 \end{cases} \quad \begin{cases} R_3 = 9 \\ x_3 = 5 \end{cases} \quad \begin{cases} Za = 1.76 - 0.13j \\ Zb = 3.10 - 0.33j \\ Zc = 2.09 + 1.97j \end{cases}$$

2). Y → Δ Conversion

$$\begin{cases} Ra = 8 \\ x_a = 3 \end{cases} \quad \begin{cases} Rb = 9 \\ x_b = -5 \end{cases} \quad \begin{cases} Rc = 7 \\ x_c = 6 \end{cases} \quad \begin{cases} Z_1 = 14.97 + 16.65\,j \\ Z_2 = 23.25 - 9.21\,j \\ Z_3 = 26.97 - 0.74\,j \end{cases}$$

[Contents] (Formulas)

1). Δ → Y Conversion

$$\dot{Z}a = \frac{\dot{Z}_1 \cdot \dot{Z}_2}{\Sigma} \ [\Omega] \qquad \Sigma = \dot{Z}_1 + \dot{Z}_2 + \dot{Z}_3$$

$$\dot{Z}b = \frac{\dot{Z}_2 \cdot \dot{Z}_3}{\Sigma} \ [\Omega] \qquad \dot{Z}_i = x_i + y_i$$

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

2). Y → Δ Conversion

$$\dot{Z}_1 = \frac{\triangle}{Zb} \quad [\Omega] \qquad \triangle = \dot{Z}a\ \dot{Z}b + \dot{Z}b\ \dot{Z}c + \dot{Z}c\ \dot{Z}a$$

$$\dot{Z}_2 = \frac{\triangle}{Zc} \quad [\Omega] \qquad \dot{Z}_i = x_i + y_i$$

$$\dot{Z}_3 = \frac{\triangle}{Za} \quad [\Omega]$$

[Key Operation Procedure] 1). Δ → Y Conversion

Step No.	Input	Display	Remarks
1	DEF A	Z1 R =_	
2	5 ENTER	Z1 X =_	
3	3 ENTER	Z2 R =_	
4	6 ENTER	Z2 X =_	
5	–2 ENTER	Z3 R =_	
6	9 ENTER	Z3 X =_	
7	5 ENTER	ZA	
8	ENTER	1.761 ··· –1.284 ··· E-01	Ra, Xa
9	ENTER	ZB	
10	ENTER	3.100 ··· –3.302 ··· E-01	Rb, Xb
11	ENTER	ZC	
12	ENTER	2.091 ··· 1.972 ···	Rc, Xc
13	ENTER	>	

[Key Operation Procedure] 2). Y → Δ Conversion

Step No.	Input	Display	Remarks
1	DEF B	ZA R =_	
2	8 ENTER	ZA X =_	
3	3 ENTER	ZB R =_	
4	9 ENTER	ZB X =_	
5	–5 ENTER	ZC R =_	
6	7 ENTER	ZC X =_	
7	6 ENTER	Z1	
8	ENTER	14.97 ··· 16.65 ···	R_1, X_1
9	ENTER	Z2	
10	ENTER	23.24 ··· –9.21 ···	R_2, X_2
11	ENTER	Z3	
12	ENTER	26.97 ··· –0.73 ···	R_3, X_3
13	ENTER	>	

PROGRAM TITLE	Δ↔Y CONVERSION	PROGRAM NO. P5–C–1	3

[Program List]

```
 10:"A":T=0:S=0:
    DEGREE
 20:INPUT "Z1 R=";
    X
 30:INPUT "Z1 X=";
    Y
 40:GOSUB 400
 50:GOSUB 350
 60:B=U:C=V
 70:INPUT "Z2 R=";
    X
 80:INPUT "Z2 X=";
    Y
 90:GOSUB 400
100:GOSUB 350
110:D=U:E=V
120:INPUT "Z3 R=";
    X
130:INPUT "Z3 X=";
    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 :WAIT :
    PRINT "ZA":
    PRINT J,K
290:PRINT "ZB":
    PRINT L,M
300:PRINT "ZC ":
    PRINT N,O
310:END
350:U=√(X*X+Y*Y)
360:V=ACS (X/U)
370:IF 0>YLET 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
520:INPUT "ZA X=";
    Y
530:GOSUB 350
540:B=U:C=V
550:INPUT "ZB R=";
    X
560:INPUT "ZB X=";
    Y
570:GOSUB 350
580:D=U:E=V
590:INPUT "ZC R=";
    X
600:INPUT "ZC X=";
    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:PRINT "Z1 ":
    PRINT L,M
870:PRINT "Z2 ":
    PRINT N,O
880:PRINT "Z3 ":
    PRINT J,K
890:END


STATUS 1
                977
```

[Memory Contents]

	Δ→Y	Y→Δ
A		
B	$R1 \rangle \dot{Z}1$	$Ra \rangle \dot{Z}a$
C	$X1 \rangle$	$Xa \rangle$
D	$R2 \rangle \dot{Z}2$	$Rb \rangle \dot{Z}b$
E	$X2 \rangle$	$Xb \rangle$
F	$R3 \rangle \dot{Z}3$	$Rc \rangle \dot{Z}c$
G	$X3 \rangle$	$Xc \rangle$
H	$\rangle \Sigma Z$	$\rangle \Delta$
I	$\rangle$	$\rangle$
J	$Ra \rangle \dot{Z}a$	$R3 \rangle \dot{Z}3$
K	$Xa \rangle$	$X3 \rangle$
L	$Rb \rangle \dot{Z}b$	$R1 \rangle \dot{Z}1$
M	$Xb \rangle$	$X1 \rangle$
N	$Rc \rangle \dot{Z}c$	$R2 \rangle \dot{Z}2$
O	$Xc \rangle$	$X2 \rangle$
P		
Q		
R		
S	√	
T	√	
U	√	$\lvert\dot{Z}\rvert$
V	√	θ
W		
X	√	√
Y	√	√
Z		

SHARP

PROGRAM TITLE	OPEN AND RADIATE TRAVERSE	PROGRAM NO. P5–C–5	1
		CE–150 required	

[Outline]

This program allows the azimuth and coordinates at individual points to be determined with the inputs of starting azimuth, starting coordinates, each included angles, and distances.

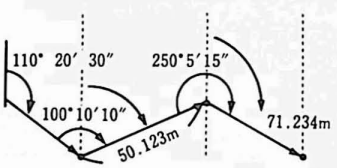
[Operating Guide]

[DEF] [A] : Open Traverse
The inputs of starting azimuth and coordinates are first made.
Next, key in the included angles at individual points and distances.
As a result, the azimuth and coordinates can be found.

[DEF] [B] : Radiate Traverse
Key in starting azimuth and coordinates.
Next, enter the included angles and distances from starting points.
As a results, the azimuth and coordinates can be found.

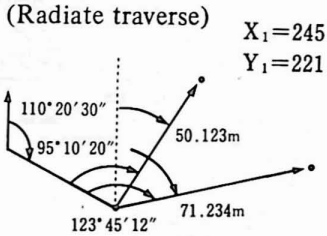
[Example]

(Open traverse)



X= 100 X_1=143.1825 X_2=130.0806
Y= 100 Y_1=125.4477 Y_2=195.4664

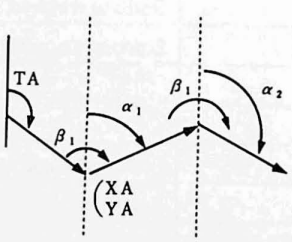
(Radiate traverse)



X= 200m
Y= 200m

[Contents] (Formulas)

A) Open traverse

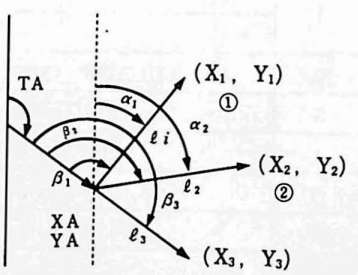


Azimuth $\alpha_i = \alpha_{i+1} + \beta i + 180° - (360°)$

Coordinates $X_i = X_{i-1} + \ell_i \cdot \cos \alpha_i$

$Y_i = Y_{i-1} + \ell_i \cdot \sin \alpha i$

B) Radiate traverse



PROGRAM TITLE	OPEN AND RADIATE TRAVERSE	PROGRAM NO. P5–C–5	2

[Printout]

```
OPEN                    RADIATE
*TA=      110.2030      *TA=      110.2030
*TX=      100.0000      *TX=      200.0000
*TY=      100.0000      *TY=      200.0000
--1--                   --1--
B=        100.1010      B=         95.1020
L=         50.1230      L=         50.1230
A=         30.3040      A=         25.3050
X=        143.1825      X=        245.2350
Y=        125.4477      Y=        221.5894
--2--                   --2--
B=        250.0515      B=        123.4512
L=         71.2340      L=         71.2340
A=        100.3555      A=         54.0542
X=        130.0806      X=        241.7746
Y=        195.4664      Y=        257.6988
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	OPEN	Open traverse
2		TA = _	Starting azimuth
3	110.2030 ENTER	TX = _	Coordinates
4	100 ENTER	TY = _	
5	100 ENTER	B = _	Included angle at each point
6	100.1010 ENTER	L = _	Distance
7	50.123 ENTER	B = _	
8	250.0515 ENTER	L = _	
9	71.234 ENTER	B = _	
10	ENTER	>	Processing is completed.
1	DEF B	RADIATE	Radiate traverse
		TA = _	Starting azimuth
2	110.2030 ENTER	TX = _	Coordinates
3	200 ENTER	TY = _	
4	200 ENTER	B = _	Included angle at each point
5	95.1020 ENTER	L = _	Distance
6	50.123 ENTER	B = _	
⋮	⋮		
9	ENTER	>	Processing is completed.

PROGRAM TITLE	OPEN AND RADIATE TRAVERSE	PROGRAM NO. P5– C–5	3

[Program List]

```
 10:"A":CLEAR
 20:PAUSE "OPEN":I
    =0
 30:LPRINT "OPEN":
    GOTO 70
 40:"B":CLEAR
 50:PAUSE "RADIATE
    ":I=1
 60:LPRINT "RADIAT
    E"
 70:DEGREE :INPUT
    "TA=";A,"TX=";
    B,"TY=";C
 75:LPRINT USING "
    ########.####"
    ;"*TA=";A
 76:LPRINT "*TX=";
    B
 77:LPRINT "*TY=";
    C
 80:IC=1
 90:INPUT "B=";D:
    GOTO 100
 95:END
100:INPUT "L=";E
110:F=DEG A+DEG D+
    180
120:IF DMS F>=360
    LET F=DEG (DMS
    F-360):GOTO 12
    0
130:G=B+E*COS F:H=
    C+E*SIN F
140:F=INT (DMS (F+
    0.000014)*10^4)
    /10^4
141:FO$="--"+STR$
    IC+"--"
142:LPRINT FO$
143:IC=IC+1
150:LPRINT USING "
    ########.####"
    ;"B= ";D
160:LPRINT "L= ";E
170:LPRINT "A= ";F
180:LPRINT "X= ";G
190:LPRINT "Y= ";H
200:IF I=0LET A=F:
    B=G:C=H
210:GOTO 90

STATUS 1
```

487

[Memory Contents]

A	TA
B	XA
C	YA
D	β
E	ℓ
F	αi
G	X_i
H	Y_i
I	Discriminant
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
FO$	Output message
IC	√

SHARP

PROGRAM TITLE	CALCULATION FOR LOAN LIMITS AND NUMBER OF INSTALLMENTS	PROGRAM NO. P5–D–1	1

CE–150 required

[Outline] (Financial Area)

If you can keep afloat, so much the better.
In need of a loan, however, you want to make it affordable. This program calculates the limits of a proper loan and the number of payments based on your solvency. Start saving with efficient payment plans even on loans.

[Operating Guide]

"A": The loan limits calculation is based on solvency (installment and No. of installments.)
(Fractions smaller than the unit are omitted.)

"B": Calculation for the number of installments is also based on the loan and solvency.

[Example]

"A": Find the loan limits on condition of monthly solvency at 150,000., 12 % annual interest and 8 years installment term.

Input: Annual repayment = 150,000 × 12
Installment term = 8
Annual interest = 12 %

"B": Assuming that a loan of 3 million is repaid with monthly solvency of 100,000 at 12% annual interest, a calculation is made on how many months are required for repayment.

Input: Loan = 3,000,000
Monthly installment = 100,000
Monthly interest = 12÷12%

[Contents] (Formulas)

	Loan Limit Calculation	Number of Installments Calculation
Input	Each Installment (amount: a) Number of installments (n times) Interest (r %)	Loan (Total amount: A) Each Installment (amount: a) Interest (r %)
Output	Loan limits	Number of installments

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

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

where $R = 1 + \frac{r}{100}$

PROGRAM TITLE	CALCULATION FOR LOAN LIMITS AND NUMBER OF INSTALLMENTS	PROGRAM NO. P5–D–1	2

[Printout]

```
NO. OF INST. =
                8.00
INSTALLMENT=
        1,800,000
INTEREST(%)=
           12.000
LIMITS=
        8,941,751

LIMITS=
        3,000,000
INSTALLMENT=
          100,000
INTEREST(%)=
            1.000
NO. OF INST.=
           35.84
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	NO. OF INST. ?_	
2	8 ENTER	INSTALLMENT ?_	
3	150000 * 12 ENTER	INTEREST (%) ?_	
4	12 ENTER	>	
5	DEF B	LIMITS ?_	
6	3000000 ENTER	INSTALLMENT ?_	
7	100000 ENTER	INTEREST (%)?_	
8	1 ENTER	>	

PROGRAM TITLE	CALCULATION FOR LOAN LIMITS AND NUMBER OF INSTALLMENTS	PROGRAM NO. P5–D–1	3

[Program List]

```
 10:"A"CLEAR :LF 2
 20:INPUT "NO. OF
    INST. ? ";A
 25:LPRINT "NO. OF
     INST. =",
    USING "###.##"
    ;A
 30:GOSUB 400
 35:J=(1+C/100)^A
 40:D=INT ((J-1)*B
    /(J*C/100))
 50:LPRINT "LIMITS
    ="
 55:LPRINT USING "
    ##############,#
    ##";D
 60:LF 3:END
200:"B"CLEAR :LF 2
210:INPUT "LIMITS?
     ";D
215:LPRINT "LIMITS
    =":LPRINT
    USING "########
    ##,###";D
220:GOSUB 400
230:K=B/(B-D*C/100
    )
240:A=LOG K/LOG (1
    +C/100)
250:LPRINT "NO. OF
     INST.=",USING
    "###.##";A
260:LF 3:END
400:INPUT "INSTALL
    MENT?";B
405:LPRINT "INSTAL
    LMENT=":LPRINT
    USING "########
    ######,###";B
410:INPUT "INTERES
    T(%)? ";C
415:LPRINT "INTERE
    ST(%)=",USING
    "####.###";C
430:RETURN
440:END
```

STATUS 1

478

[Memory Contents]

A$	No. of installments
B	Installment
C	Interest (%)
D	Loan limits
E	
F	
G	
H	
I	
J	Calculation Work
K	Calculation Work
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

SHARP

PROGRAM TITLE	COMPOUND ANNUITY RATE CALCULATION	PROGRAM NO. P5–D–4	1

[Outline]

This program calculates the current compound annuity rate at the end and beginning of a term, as well as the outstanding amount at each term end.

[Operating Guide]

DEF A : 1. Term end outstanding amount input
2. Interest input
3. Term input
4. No. of installments input
5. Interest calculation
6. Term calculation

DEF B : 1. Calculation of current price payable at term end
2. Calculation of current price payable at term beginning

DEF D : 1. Outstanding amount at term end

[Example]

1. Determine the current annuity payable in 9 years with 90,000 at the end of a 6-month term, and an interest rate of 5 %.
2. Determine the current annuity payable in 9 years with 90,000 at the beginning of a 6-month term, and an interest rate of 5 %.
3. Amount of five million loan is made at an interest rate of 8 % (two settlements per year), and repaid in 5 year installments at 6-month compound interest. What is an installment at term end?
   · Interest unit: 1 (Fractions are rounded-off.)

(Note: 1 and 2 are determined simultaneously so that they can be compared.)

[Contents] (Formulas)

Interest rate = Interest rate ÷ No. of installments ÷ 100

$Y = 1 - (\text{Interest rate} + 1)^{-n}$ n = Installment term

Current price at term end = Outstanding amount × Y ÷ Interest rate

Current price at term beginning = Outstanding amount × Y ÷ Interest rate × (Rate + 1)

Outstanding amount at term end = Outstanding amount × Interest rate ÷ Y

(Fractions of amounts are rounded-off.)

PROGRAM TITLE	COMPOUND ANNUITY RATE CALCULATION	PROGRAM NO. P5–D–4	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	OUTSTD. AMT.= _	Data input
2	90000 ENTER	RATE = _	
3	5 ENTER	TERM = _	
4	9 ENTER	NO. OF INSTL. = _	
5	2 ENTER	>	
6	DEF B	TERM-END CRNT. PR.	Term end current price displayed
7	ENTER	1291803	
8	ENTER	TERM-BEGNNG CRNT. PR.	Current price due displayed
9	ENTER	1324098	
10	DEF A	OUTSTD. AMT. = _	
11	5000000 ENTER	RATE = _	Data input
12	8 ENTER	TERM = _	
13	5 ENTER	NO. OF INSTL. = _	
14	2 ENTER	>	
15	DEF D	OUTSTDNG AMT AT TRM END	Display of outstanding amount at term end
16	ENTER	616455	

PROGRAM TITLE	COMPOUND ANNUITY RATE CALCULATION	PROGRAM NO. P5–D–4	3

[Program List]

```
 10:"A":CLEAR
 20:INPUT "OUTSTD.
    AMT.=";R
 30:INPUT "RATE=";
    I
 40:INPUT "TERM=";
    N
 50:INPUT "NO. OF
    INSTL.=";L
 60:I=(I/L)/100:N=
    N*L
 70:Y=1-(I+1)^(-N)
 80:END
100:"B":M=INT (R*Y
    /I+0.5)
110:WAIT :PRINT "T
    ERM-END CRNT.
    PR."
115:CLS :PRINT M
120:S=INT (R*Y/I*(
    I+1)+0.5)
125:WAIT :PRINT "T
    ERM-BEGNNG CRN
    T. PR."
130:CLS :PRINT S
135:END
140:"D":A=INT (R*I
    /Y+0.5)
150:WAIT :PRINT "O
    UTSTDNG AMT AT
     TERM END"
155:CLS :PRINT A
160:END
```

STATUS 1

342

[Memory Contents]

A	Outstanding amount at term end
B	
C	
D	
E	
F	
G	
H	
I	Interest rate
J	
K	
L	No. of installments
M	Current price at term end
N	Term
O	
P	
Q	
R	Outstnding amount at term end and beginning
S	Current price at term beginning
T	
U	
V	
W	
X	
Y	√
Z	

SHARP

PROGRAM TITLE	ESTIMATION ADDITION	PROGRAM NO. P5–D–5	1
		CE–150 required	

[Outline]

Product numbers and prices are first keyed-in and registered. Then, an estimate can be generated by only keying-in the quantities and discount rates or discount amounts of any desired products. Registrations can be up to 35 items.

[Operating]

[DEF] [A] : For registrations or modifications. To register, key-in all the items to be registered. Product names and prices can be modified.

[DEF] [B] : Prints all the registered product names and prices.

[DEF] [C] : Recall the required product names, then input quantities and discount rates or discount amounts. The estimation will be printed out.

[Example]

1. Register list:

Product name	Price
A–11	1,000
A–12	2,000
A–13	3,000
A–14	4,000
A–15	5,000
B–11	1,100
B–12	2,200
B–13	3,300
B–14	4,400
B–15	5,500

With product A–15, discount rate is 10 % for the quantity of 5.
With A–15, discount is 3,000 for the quantity of 15.
With these data, key-in them in accordance with the key operation Procedure for result printout.

2. If the total No. of items input exceeds that of preregistered, the display of "EXCEED REG. NO." appears. Therefore, retype the data.
3. The maximum number of characters is 16 for product name.

[Contents] (Formulas)

A···12····· (A)
@ 2,000 ··· (B)
* 10 ··· (C)
= 20,000··· (D)
−1,000 ··· (E)
19,000 ··· (F)

A = Product name
B = Price
C = Quantity
D = Price × Quantity

$$E = D \times \frac{\text{Discount rate}}{100}$$ or Discount amount

F = D − E

F is added to the total.

- Registration numbers in the register list are automatically allocated.

PROGRAM TITLE	ESTIMATION ADDITION	PROGRAM NO. P5–D–5	2

[Printout]

```
* DETAILS *
A-15
@      5,000
*          5
=     25,000
                -2,500
                22,500
A-12
@      2,000
*         15
=     30,000
                -3,000
                27,000
A-13
@      3,000
*         10
=     30,000
                30,000
TOTAL
                79,500
```

```
* REGISTER LIST *
   1 A-11
                1,000
   2 A-12
                2,000
   3 A-13
                3,000
   4 A-14
                4,000
   5 A-15
                5,000
   6 B-11
                1,100
   7 B-12
                2,200
   8 B-13
                3,300
   9 B-14
                4,400
  10 B-15
                5,500
```

PROGRAM TITLE	ESTIMATION ADDITION	PROGRAM NO. P5–D–5	3

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	REG. = 1, CHANGE = 2	With the input of 2, procedure follows step 24.
2	1 ENTER	MAX. NO. OF ITEMS = _	
3	10 ENTER	PROD. NAME = _	Repeat for No. of inputs
4	A–11 ENTER	PRICE = _	
5	1000 ENTER	PROD. NAME = _	
⋮	⋮	⋮	
23	5500 ENTER	REGISTER END	
24	2 ENTER	CHANGE NO. = _	
25	4 ENTER	A–44 =? _	Press only ENTER key when no change is made.
26	A–14 ENTER	4000 =? _	Press (content) ENTER for content change.
27	ENTER	CHANGE NO. = _	
28	ENTER	>	Key-in register No. if more changes needed.
29	DEF B	>	Register list printout
30	DEF C	REGISTER No. = _	
31	5 ENTER	QUANTITY = _	
32	5 ENTER	DISCOUNT RATE = _	Key-in discount rate in percent.
33	10 ENTER	REGISTER No. = _	
34	2 ENTER	QUANTITY = _	
35	15 ENTER	DISCOUNT RATE = _	
36	ENTER	DISCOUNT AMOUNT = _	When discount amount is keyed-in.
37	3000 ENTER	REGISTER NO. = _	
38	3 ENTER	QUANTITY = _	
39	10 ENTER	DISCOUNT RATE = _	No Discount
40	ENTER	DISCOUNT AMOUNT = _	
41	ENTER	REGISTER NO. = _	
42	ENTER	>	Upon execution completion, total printout

PROGRAM TITLE	ESTIMATION ADDITION	PROGRAM NO. P5–D–5	4

[Program List]

```
 10:"A":WAIT 0
 20:INPUT "REG.=1,
    CHANGE=2  ";X$
 30:IF (X$="1")+(X
    $="2")<>1GOTO
    20
 40:IF X$="2"GOTO
    150
 50:CLEAR :INPUT "
    MAX. NO. OF IT
    EMS=";N:DIM A$
    (N-1),A(N-1)
 60:FOR I=0TO N-1
 70:INPUT "PROD. N
    AME=";A$(I)
 80:INPUT "PRICE="
    ;A(I)
100:NEXT I
110:PAUSE "REGISTE
    R END"
120:END
150:CLS :INPUT "CH
    ANGE NO.=";C:
    GOTO 170
160:END
170:IF C>NPAUSE "E
    XCEED REG. NO.
    ":GOTO 150
180:PRINT A$(C-1);
    "=";
190:INPUT A$(C-1)
200:CLS :PRINT A(C
    -1);"=";
210:INPUT A(C-1)
215:GOTO 150
220:END
300:"B":WAIT 0
302:LF 2
304:USING :LPRINT
    "* REGISTER LI
    ST *"
306:FOR I=0TO N-1
310:IF A$(I)=""
    GOTO 330
320:USING :LPRINT
    USING "####";I
    +1;" ";A$(I)
325:USING :LPRINT
    USING "########
    ,###";A(I)
330:NEXT I
340:END
400:"C";WAIT 0:Z=0
403:LF 2
405:USING :LPRINT
    "* DETAILS *"
410:INPUT "REGISTE
    R NO.=";D:GOTO
    417
415:GOTO 620
417:IF D>NPAUSE "E
    XCEED REG. NO.
    ":GOTO 410
420:INPUT "QUANTIT
    Y=";E
430:INPUT "DISCOUN
    T RATE=";F:
    GOTO 450
440:INPUT "DISCOUN
    T AMOUNT=";G
450:J=D-1
475:V=A(J)*E
480:IF F<>0GOTO 51
    0
490:W=-G:GOTO 520
510:W=-(V*F/100)
520:Y=A(J)*E+W
530:USING :LPRINT
    A$(J)
540:USING :LPRINT
    "@";USING "###
    ####,###";A(J)
541:LPRINT "*";E
542:LPRINT "=";V
550:IF W<>0USING :
    LPRINT USING "
    ########,###";W
560:USING :LPRINT
    USING "########
    ,###";Y
600:Z=Y+Z:F=0:G=0
610:GOTO 410
620:USING :LPRINT
    "TOTAL"
630:USING :LPRINT
    USING "######,
    ###";Z
640:END


STATUS 1
                997
```

[Memory Contents]

A	
B	
C	Change No. input
D	Register No. input
E	Quantity
F	Discount Rate
G	Discount Amount
H	
I	√
J	√
K	
L	
M	
N	No. of Registers
O	
P	
Q	
R	
S	
T	
U	
V	Amount before Discount
W	Discount Amount
X	
Y	Total Amount after Discount
Z	Grand Total Amount after Discount
X$	Register and Change Acceptance
A$(N-1)	Product name
A(N-1)	Price

SHARP

PROGRAM TITLE	HISTOGRAM	PROGRAM NO. P5– D–7	1
		CE–150 required	

[Outline]

It is often necessary to obtain the frequency distribution of the data when it is grouped into a broader classification. This program generates histograms, making visual data assessmenst possible.

[Operation Guide]

1. Parameter inputs (No. of data, class initial value, class interval, and number of classes)
2. Setting the way of the data input (Key input or cassette input)
   Key input: Data to be keyed-in then to be output to the cassette tape.
   Cassette input: Data to be input from the cassette tape.
3. The variance and the standard deviation are calculated for printouts.
4. The histogram is printed out.

[Example]

No. of data = 10, Class initial value = 0, Class interval = 2, Class number = 5.

5	2	7	9	8	1	3	4	6	8

Variance: 6.81
Standard Deviation: 2.60959767

[Contents] [Formulas]

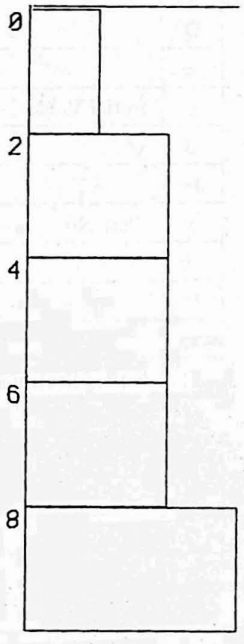
$$V=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2 \qquad \text{(Variance)}$$

$$S=\sqrt{V} \qquad \text{(Standard deviation)}$$

PROGRAM TITLE	HISTOGRAM	PROGRAM NO. P5– D–7	2

[Printout] The actual printout is colored. Refer to page 2 .

```
VARIANCE=
                6.81
STD. DEV.=
          2.60959767
```



[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	NO. OF DATA = _	
2	10 ENTER	INITIAL VALUE = _	
3	0 ENTER	SECTIONAL INTERVAL = _	
4	2 ENTER	NO. OF SECTIONS = _	
5	5 ENTER	KEY-IN? (Y,N)_	
6	Y ENTER	DATA = _	
7	5 ENTER	DATA = _	With Y input, key-in data.
8	0.2 ENTER	DATA = _	
⋮	⋮		
16	8 ENTER	>	Printout

PROGRAM TITLE	HISTOGRAM	PROGRAM NO. P5– D–7	3

[Program List]

```
 10:"A":CLEAR :
    TEXT :USING
 20:INPUT "NO. OF
    DATA=";N
 30:INPUT "INITIAL
    VALUE=";F
 40:INPUT "SECTION
    AL INTERVAL=";
    B
 50:INPUT "NO. OF
    SECTIONS=";M
 60:DIM A1(N-1),H(
    M-1)
 70:FOR C=0TO M-1
 80:H(C)=0
 90:NEXT C
100:INPUT "KEY-IN?
    (Y,N)";A$
110:IF A$="N"GOTO
    170
115:Z=F+B*M-1:X=0
120:FOR C=0TO N-1
130:INPUT "DATA=";
    A1(C):GOTO 150
140:GOTO 160
150:IF A1(C)>ZGOTO
    130
152:IF A1(C)<FGOTO
    130
153:X=X+1
155:NEXT C
160:PRINT #X,A1(*)
165:GOTO 180
170:INPUT #X,A1(*)
180:S=0:N=X
190:FOR C=0TO N-1
200:I=INT ((A1(C)-
    F)/B)
210:H(I)=H(I)+1
220:S=S+A1(C)
230:NEXT C
240:U=S/N:T=0
250:FOR C=0TO N-1
260:T=T+(A1(C)-U)^
    2
270:NEXT C
280:T=T/N:S=√T
290:COLOR 0:LPRINT
    "VARIANCE=",T
300:LPRINT "STD. D
    EV.=";S
310:N=-10^(98)
320:FOR C=0TO M-1
330:IF H(C)>NLET N
    =H(C)
340:NEXT C
350:GRAPH
360:GLCURSOR (50,0
    ):SORGN
370:COLOR 0
380:LINE (0,0)-(15
    0,0)
390:LINE (0,0)-(0,
    -450)
400:L=450/M:N=N/15
    0
410:W=0:Q=F
420:FOR C=0TO M-1
422:COLOR 2:
    GLCURSOR (-50,
    W-15)
424:LPRINT USING "
    ####";Q
430:COLOR 1
435:G=INT (H(C)/N)
440:LINE (0,W)-(G,
    W)-(G,W-L)-(0,
    W-L)
450:W=W-L
470:Q=Q+B
480:NEXT C
490:END

STATUS 1
                844
```

[Memory Contents]

A	
B	Sectional interval
C	√
D	
E	
F	Initial Value
G	√
H	
I	Class No.
J	
K	
L	
M	No. of Sections
N	No. of Data
O	
P	
Q	√
R	
S	$\Sigma A1(i), \sqrt{T}$
T	$\frac{1}{N}\Sigma(A1(i)-\bar{A})^2$
U	
V	Mean Value
W	√
X	No. of Effective Data
Y	
Z	Maximum Effective Value
A$	√
A1(N–1)	Data Table
H(M–1)	Data Table for Classes

SHARP

PROGRAM TITLE	GRAPH GENERATION I (BAND OR CIRCLE GRAPH)	PROGRAM NO. P5– D–8	1

CE–150 and CE–151 required

[Outline]

With this program, you can generate circle or band graph by keying in statistical data.

[Operating Guide]

Input: Item name (within 10 characters)
Item value
) Input of up to 10 items are possible.
Band graph or circle graph selection

Output: Item name, rate (% display)
Band or circle graph

[Example]

Key in statistical information by age, as follows:

(1) 20 people age 0 to 10
(2) 60 people age 11 to 20
(3) 45 people age 21 to 30
(4) 35 people age 31 to 40
(5) 25 people age 41 to 50
(6) 20 people age 51 to 60
(7) 15 people age 61 to 70

For the results, refer to the "Printout".

[Contents] (Formulas)

- The ratio of an item value to the total item value is displayed in percent (%) on the graph.

$D = A(J) \div H \times 100$

D : Ratio
A(J): An item value
H : Total item value

- Circle graph generation
With a circle sectioned in 12° increments from 0° to 360°, points (X1 and Y1) on a circular arc with a radius of 20mm are calculated for segmented connection.

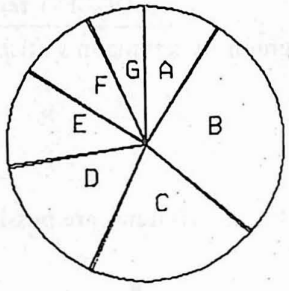
$X1 = R \times SIN\ C$
$Y1 = R \times COS\ C$

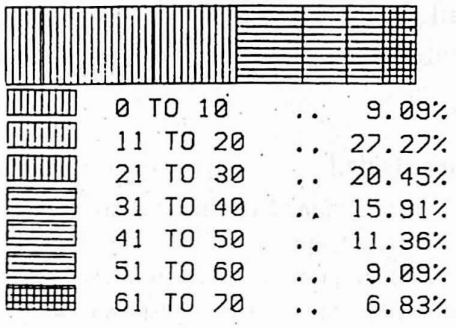
R: Radius
C: Angle

- The ratio is displayed with the value rounded off to two decimal places.

PROGRAM TITLE	GRAPH GENERATION I (BAND OR CIRCLE GRAPH)	PROGRAM NO. P5–D–8	2

[Printout] The actual printout is colored. Refer to page 2 .






```
A 0 TO 10
  ....... 9.09%
B 11 TO 20
  ....... 27.27%
C 21 TO 30
  ....... 20.45%
D 31 TO 40
  ....... 15.91%
E 41 TO 50
  ....... 11.36%
F 51 TO 60
  ....... 9.09%
G 61 TO 70
  ....... 6.83%
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	ITEM NAME (1)	
2	0 to 10 ENTER	VALUE (1)	
3	11 to 20 ENTER	ITEM NAME (2)	
⋮	⋮	⋮	
13	51 to 60 ENTER	ITEM NAME (7)	
14	61 to 70 ENTER	VALUE (7)	
15	15 ENTER	ITEM NAME (8)	
	ENTER	CIRCLE = 1 BAND = 2 _	Circle graph ..... 1 Band graph 2
16	1 ENTER		
			Graph printout

PROGRAM TITLE	GRAPH GENERATION I (BAND OR CIRCLE GRAPH)	PROGRAM NO. P5– D–8	3

[Program List]

```
 10:"A":WAIT 0:
    CLEAR :Q=9:DIM
    A$(Q)*10,B$(Q)
    *1,A(Q)
 20:B$(0)="A":B$(1
    )="B":B$(2)="C
    ":B$(3)="D":B$
    (4)="E":B$(5)=
    "F"
 25:B$(6)="G":B$(7
    )="H":B$(8)="I
    ":B$(9)="J"
 30:FOR I=0TO Q
 40:C$="ITEM NAME(
    "+STR$ (I+1)+"
    )":PRINT C$;
 50:INPUT A$(I):
    GOTO 70
 60:CLS :I=I-1:
    GOTO 100
 70:CLS :C$="VALUE
    ("+STR$ (I+1)+
    ")=":PRINT C$;
 80:INPUT A(I):CLS
    :H=H+A(I)
 90:NEXT I
100:I=I+1
110:INPUT "CIRCLE=
    1 BAND=2";C
120:IF (C=1)+(C=2)
    <>1GOTO 110
130:IF C=2GOTO 300
140:GRAPH :
    GLCURSOR (110,
    -125):SORGN
150:D=12:Y=100:R=1
    00:L=1:C=0
160:FOR J=1TO 31
170:GOSUB 600:LINE
    (X,Y)-(X1,Y1):
    X=X1:Y=Y1:C=C+
    D
180:NEXT J
190:FOR J=0TO I-1
195:R=100
200:F=360*A(J)/H:F
    =G+F:IF J=I-1
    LET F=360
210:FOR M=1TO 2
215:IF M=1LET C=G+
    .5:GOTO 225
220:C=F-.5
225:GOSUB 600:IF L
    >3LET L=1
230:LINE (0,0)-(X1
    ,Y1),0,L:NEXT
    M
235:R=50:C=(F-G)/2
    +G:GOSUB 600:X
    1=X1-3
260:G=F
261:GLCURSOR (X1,Y
    1):LPRINT B$(J
    ):L=L+1:NEXT J
262:GLCURSOR (-110
    ,-150):SORGN
264:Y=0:X=0:COLOR
    0
265:FOR J=0TO I-1
267:D=A(J)/H*100:D
    =INT ((D+.005)
    *100)/100:IF J
    =I-1LET D=100-
    N:GOTO 270
268:N=N+D
270:GLCURSOR (X,Y)
    :LPRINT B$(J)
275:GLCURSOR (18,Y
    ):LPRINT A$(J)
280:Y=Y-20
282:GLCURSOR (18,Y
    ):LPRINT "....
    ....";USING "##
    #.##";D;"%":
    USING
284:Y=Y-20
285:NEXT J
290:TEXT :LF 10:
    END
300:GRAPH :
    GLCURSOR (0,0)
    :SORGN :ROTATE
    1
312:K=1:L=1:S=160:
    V=215
315:FOR J=0TO I-1
320:D=INT (A(J)/H*
    100+.5):E=D*3
325:W=T-E:IF J=I-1
    LET W=-300
327:IF L>3LET L=1:
    K=K+1
330:LINE (160,T)-(
    215,W),0,0,B:
    GOSUB 650
332:T=W:L=L+1:NEXT
    J
335:K=1:L=1:W=-50:
    T=0
336:FOR J=0TO I-1
338:IF L>3LET L=1:
    K=K+1
340:F=160/I*(I-J-1
    ):LINE (F,0)-(
    (F-5+160/I),-5
    0),0,0,B
345:S=F:V=F-5+160/
    I:GOSUB 650
349:COLOR 0:
    GLCURSOR (F,-8
    0):LPRINT A$(J
    )
350:GLCURSOR (F,-2
    10):LPRINT "•
    "
351:D=A(J)/H*100:D
    =INT ((D+.005)
    *100)/100
352:IF J=I-1LET D=
    100-G:GOTO 355
353:G=G+D
355:GLCURSOR (F,-2
    40):LPRINT
    USING "###.##"
    ;D;"%":USING
368:L=L+1:NEXT J
370:TEXT :LF 10:
    END
```

(To be continued)

PROGRAM TITLE	GRAPH GENERATION I (BAND OR CIRCLE GRAPH)	PROGRAM NO. P5– D–8	4

[Program List]

```
600:X1=R*SIN C:Y1=
    R*COS C:RETURN
650:IF K>3LET K=1
655:IF K=1GOSUB 70
    0
660:IF K=2GOSUB 75
    0
665:IF K=3GOSUB 70
    0:GOSUB 750
690:RETURN
700:P=T:FOR O=1TO
    60
705:P=P-5
710:IF P<=WGOTO 74
    0
715:IF O-INT (O/2)
    *2=0LINE (S,P)
    -(U,P),0,L:
    GOTO 725
720:LINE (U,P)-(S,
    P),0,L
725:NEXT O
740:RETURN
750:P=S:FOR O=1TO
    50
755:P=P+5
760:IF P>=UGOTO 79
    0
765:IF O-INT (O/2)
    *2=0LINE (P,T)
    -(P,W),0,L:
    GOTO 775
770:LINE (P,W)-(P,
    T),0,L
775:NEXT O
790:RETURN
```

```
STATUS 1
                1772
```

[Memory Contents]

A	
B	
C	Circle and band graph selection code
D	
E	
F	Angle (1)
G	Angle (2)
H	Total item value
I	Loop counter
J	Loop counter
K	Pattern selection in graph
L	Pen color code
M	Loop counter
N	Total ratio
O	Loop counter
P	√
Q	√
R	√
S	Band graph X-axis (1)
T	Band graph Y-axis (1)
U	
V	Band graph X-axis (2)
W	Band graph Y-axis (2)
X	√
Y	√
Z	
D$	Display character editing
A$(Q)*10	Item name
B$(Q)*1	Alphabet
A(Q)	Item value
X1	X-axis
Y1	Y-axis

SHARP

PROGRAM TITLE	GRAPH GENERATION II (BAR OR BROKEN LINE GRAPH)	PROGRAM NO. P5–D–9	1
		CE–150 required	

[Outline]

With the input of statistical data, you can generate bar or broken line graphs.
(Vertical graphs are produced on roll paper.)

[Operating Guide]

Input: Title
Graph selection (Bar graph = 1, and broken line graph = 2)
Items (No. of items: up to 8 items.)
Item name (within 16 characters)
Item value

Output: Bar graph or broken line graph
For bar graph, No. 1 to 4 item are represented by horizontal lines in 4 different colors.
Differently colored horizontal doted lines represent No. 5 to 8 item.

[Example]

(1) Title: Sales chart
Graph selection: Bar graph = 1

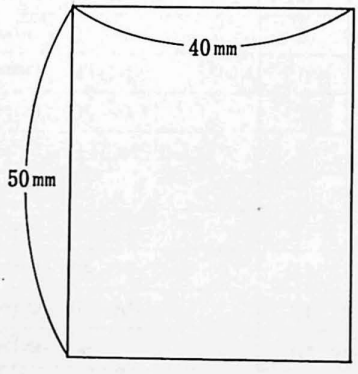
Item:	Item name	Item value	
(1)	Pen	10	Type in the items on the left.
(2)	Note	20	
(3)	Pencil	30	
(4)	Book	40	
(5)	Paper	50	

For the output, refer to the "Printout".
For the broken line graph, the order of items is different.

[Contents] (Formulas)

(1) Horizontal direction of the graph
- Bar graph
  Horizontal width of an item
  = Horizontal width (40mm)
  ÷ No. of items – space (1mm)
- Broken line
  Horizontal width of an item
  = Horizontal width
  ÷ (No. of items + 1)

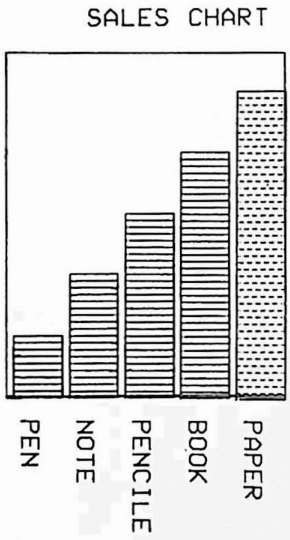
(2) Vertical direction of the graph

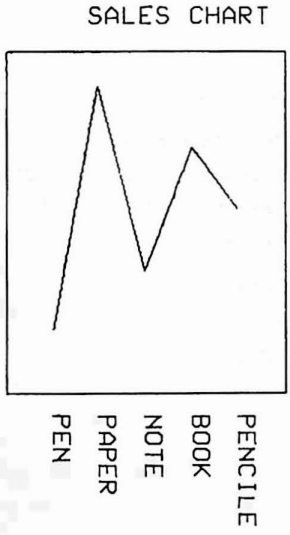


PROGRAM TITLE	GRAPH GENERATION II (BAR OR BROKEN LINE GRAPH)	PROGRAM NO. P5– D–9	2

Making the vertical length of the max. input item value 45mm, the vertical lengths of other item values are calculated.
Vertical length of an item = 45m/m ÷ Maximum item value × Item value.

[Printout] The actual printout is colored. Refer to page 2 .





[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	TITLE ? _	
2	SALES CHART ENTER	BAR=1, BROCKEN LINE=2 ?_	
3	1 ENTER	ITEM (1) =	The following also applies to the input of broken line graph.
4	PEN ENTER	VALUE (1) =	
5	10 ENTER	ITEM (2) =	
6	NOTE ENTER	VALUE (2) =	
7	20 ENTER	ITEM (3) =	
8	PENCILE ENTER	VALUE (3) =	
9	30 ENTER	ITEM (4) =	
10	BOOK ENTER	VALUE (4) =	
11	40 ENTER	ITEM (5) =	
12	PAPER ENTER	VALUE (5) =	
13	50 ENTER	ITEM (6) =	
14	ENTER	>	Bar graph printout

PROGRAM TITLE	GRAPH GENERATION II (BAR OR BROKEN LINE GRAPH)	PROGRAM NO. P5– D–9	3

[Program List]

```
 10:"A":WAIT 0:
    CLEAR :DIM A$(
    8),A(8)
 20:INPUT "TITLE?"
    ;A$(0)
 30:INPUT "BAR=1 ,
     BROKEN LINE=2
     ?";C
 40:IF (C=1)+(C=2)
    <>1GOTO 30
 50:FOR I=1TO 8
 60:B$="ITEM("+
    STR$ I+")=":
    PRINT B$;
 65:INPUT A$(I):
    CLS :GOTO 80
 70:CLS :I=I-1:
    GOTO 100
 80:B$="VALUE("+
    STR$ I+")=":
    PRINT B$;
 85:INPUT A(I):CLS
 87:IF D<A(I)LET D
    =A(I)
 90:NEXT I
100:LPRINT A$(0)
105:D=45/D
110:GRAPH
120:GLCURSOR (0,-2
    50):SORGN
130:IF C=2LET G=2
140:LINE (0,0)-(20
    0,250),0,G,B
150:IF C=2GOTO 400
160:G=5
170:E=(40-I)/I*5
180:FOR J=1TO I
190:H=G+E
200:F=D*A(J)*5
220:GOSUB 600:G=H+
    5:NEXT J:G=5
230:FOR J=1TO I:H=
    G+E
235:N=G+E/2-10:
    GOSUB 800
240:G=H+5:NEXT J
245:GLCURSOR (0,-2
    50)
250:TEXT :LF 5:END
400:E=40/(I+1)*5
410:FOR J=1TO I
420:H=E*J
430:F=D*A(J)*5
440:IF J=1GOTO 460
450:LINE (G,M)-(H,
    F),0,3
460:G=H:M=F
480:NEXT J
483:FOR J=1TO I:H=
    E*J
485:N=H:GOSUB 800:
    NEXT J
487:GLCURSOR (0,-2
    50)
490:TEXT :LF 5:END
600:M=M+1:L=L+1
610:IF L=4LET L=0
620:GLCURSOR (G,0)
    :LINE (G,0)-(H
    ,F),0,L,B
630:P=0:IF M>4LET
    P=2
700:O=0
705:FOR K=1TO 45
708:O=O+5
710:IF F<=OGOTO 72
    0
713:IF K-INT (K/2)
    *2=1LINE (G,O)
    -(H,O),P:GOTO
    718
715:LINE (H,O)-(G,
    O),P
718:NEXT K
720:RETURN
800:ROTATE 1
810:GLCURSOR (N,-1
    5):COLOR 0
820:LPRINT A$(J)
830:ROTATE 0
840:RETURN
```

STATUS 1

919

[Memory Contents]

A	
B	
C	Graph selection
D	Maximum item value
E	Graph horizontal width of an item
F	Y-Coordinate
G	X-Coordinate
H	X-Coordinate
I	
J	
K	
L	Pen color No.
M	
N	X-Coordinate
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
B$	Character editing
A$(8)	A$(0): Title A$(1) to (8) : Item name
A(8)	Item value

SHARP

PROGRAM TITLE	WORKING HOUR PROPORTIONAL PROCESSING	PROGRAM NO.	1
		P5–D–11	

[Outline]

Values of working hours can be determined with the inputs of starting and closing times.

Be noted that 24 hour system is employed here.

[Operating Guide]

(1) First clear the total by pressing the [DEF] [D] keys. Then set the proportional value using the [DEF] [C] keys. (After this, use the [DEF] [D] or [DEF] [C] keys as needed.)

(2) Press the [DEF] [A] to key-in the staring time and the closing time. The value for the working hours will be displayed.

(3) Repeat the [DEF] [C] and [DEF] [A] according to proportional value and number of data.

(4) The total value is displayed by using the [DEF] [B] keys.

[Example]

(1) Keyin proportional value 500 after the [DEF] [C] .
(This should be the proportional value to the working hours between 9:00 and 17:00)

(2) The [DEF] [D] key is used to clear the total area to zero.

(3) With the work-hour data 9:30 to 17:00, 14:00 to 16:00 and 17:00 to 23:10, input "9.30" "17.00", and "14.00", "16.00" after the [DEF] [A] operation, then "7.30 (T) *500 = 3750" and "2.00 (T) *500 = 1000" will be displayed respectively.
When the proportional value after 17:00 is 1000, replace 500 with 1000 after the [DEF] [C] operation, then key-in "17.00", "23.10" after [DEF] [A] . As a result, "6.10(T)*1000 = 6166" is displayed.

(4) "TOTAL = 10916" is displayed after [DEF] [B] operation.

[Contents] (Formulas)

"A" With the inputs of the starting time and the closing time (Minutes should be a decimal number), "Elapsed Time × Proportional Value = Work-hour Value" is displayed.
There is no limit to the number of data.
Pressing the [ENTER] ends processing.

"B" The total value for working hours is displayed.

"C" The proportional value is reset.

"D" The total area is cleared to zero.

NOTE: For the elapsed time display, "9.30 (T)" means 9 hours 30 minutes.

PROGRAM TITLE	WORKING HOUR PROPORTIONAL PROCESSING	PROGRAM NO. P5–D–11	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF D	TOTAL CLEAR	
		>	
2	DEF C	PROPORT. VAL. _	
	500 ENTER	>	
3	DEF A	START ?_	
4	9.30 ENTER	END ?_	Close/Input the minute as decimal number.
5	17.00 ENTER	7.30 (T)*500 =	
6	ENTER	3750	
7	ENTER	START ?_	
8	14.00 ENTER	END ?_	
9	16.00 ENTER	2.00(T)*500 =	
10	ENTER	1000	
11	ENTER	START ?_	
12	ENTER	>	
13	DEF C	PROPORT. VAL.	
	1000 ENTER	>	
14	DEF A	START ?_	
15	17.00 ENTER	END ?_	
16	23.10 ENTER	6.10 (T)*1000 =	
17	ENTER	6166	
18	ENTER	START ?_	
19	ENTER	>	
20	DEF B	TOTAL = 10916	
	ENTER	>	

PROGRAM TITLE	WORKING HOUR PROPORTIONAL PROCESSING	PROGRAM NO. P5–D–11	3

[Program List]

```
 15:"A"WAIT :INPUT
    "START?";O:
    GOTO 20
 18:END
 20:GOSUB 500:S=O
 30:INPUT "END?";O
 40:GOSUB 500:E=O
 50:M=0
 60:M=E-S
300:F=M*D
303:O=M:GOSUB 600:
    M=O
320:T=T+F
330:USING :PRINT
    USING "###.##"
    ;M;"(T) *";
    USING "#####";
    D;"="
335:USING :PRINT
    USING "#######
    ";F
340:GOTO 15
350:"B":USING :
    PRINT "TOTAL="
    ;USING "######
    ##";T
360:END
400:"C":INPUT "PRO
    PORT. VAL.";D
420:END
450:"D":T=0
460:USING :PAUSE "
    TOTAL CLEAR"
470:END
500:K=INT O:I=(O-K
    )*100
510:I=I/60:O=K+I
520:RETURN
600:K=INT O:I=(O-K
    )
610:I=(I*60)/100:O
    =K+I
620:RETURN
```

STATUS 1

402

[Memory Contents]

A	
B	
C	
D	Proportional Value
E	Closing Time (after calculation)
F	Value for Working Hours
G	
H	
I	√
J	
K	√
L	
M	Elapsed Time
N	
O	Starting Time/ Closing Time
P	
Q	
R	
S	Starting Time (after calculation)
T	Total of F
U	
V	
W	
X	
Y	
Z	

SHARP

PROGRAM TITLE	DEPRECIATION	PROGRAM NO. P5–D–12	1

[Outline]

Calculations of ordinary depreciation amounts and undepreciated remainders are possible with this program either in the fixed rate or fixed amount method.

[Operating Guide]

Calculation based on the fixed rate method

Press the [DEF] [A] to enter acquisition cost, remaining value, and the number of times. This displays depreciation amounts and undepreciated amounts disignated times. Finally, the total depreciation amount is also displayed.

Calculation based on the fixed amount method

Press the [DEF] [B] to input acquisition cost, years of life, depreciation month, and remaining value, then the depreciation amount and the undepreciated amount will be displayed. Finally, totals for individual items are also displayed.

[Example]

(1) Fixed rate method

Determines the depreciation amount, undepreciated amount and total depreciation amount per term for product A with the acquisition cost of 800,000, life of 6 years, and remaining rate of 10%. Two settlements per year.

(2) Fixed amount method

Determines the depreciation amounts and undepreciated remainders for both product A and product B with the following conditions.

Product A: 900,000 as an acquisition cost, 5 years of life, and 6 months as the depreciation term this year.

Product B : 720,000 as an acquisition cost, 25 years of life, and 8 months as the depreciation term this year.

For both of them, the remaining rate is 10%.

(For input/output, refer to the Key Operation Procedure.)

[Contents] (Formulas)

(Fixed rate method)

Depreciation amount = Acquistion costs × depreciation rate

Undepreciated remainder = Acquistion cost − Depreciation amount

$$\text{Depreciation rate} = 1 - \left(\frac{\text{remaining rate (\%)}}{100} \right)^{\frac{1}{n}} \qquad n = \text{Years of life}$$

(Fixed amount method)

$$\text{Depreciation amount} = \left(\begin{array}{l}\text{Acquisition}\\ \text{cost}\end{array}\right) \times \left(\frac{100 - \begin{array}{l}\text{Remaining}\\ \text{rate}\end{array}\ (\%)}{100}\right) \times \left(\frac{1}{\text{Years of life}}\right) \times \left(\frac{\begin{array}{l}\text{No. of Depreciation}\\ \text{months}\end{array}}{12}\right)$$

$$\text{Undepreciated remainder} = \left(\begin{array}{l}\text{Acquisition}\\ \text{cost}\end{array}\right) - \left(\begin{array}{l}\text{Depreciation}\\ \text{amount}\end{array}\right)$$

The remaining rate is at least 5%.

[Key Operation Procedure] : Fixed rate method.

Step No.	Input	Display	Remarks
1	DEF A	COST?_	
2	800000 ENTER	NO. OF TIMES ?_	
3	12 ENTER	REM. RATE (%) ?_	
4	10 ENTER	1 DEPR. = 139680	
5	ENTER	1 UNDEPR. = 660320	
⋮	⋮	⋮	
12	ENTER	5 DEPR. = 64832	
13	ENTER	5 UNDEP. = 306489	
⋮	⋮	⋮	
26	ENTER	12 DEPR. = 16922	
27	ENTER	12 UNDEP. = 79998	
28	ENTER	TOTAL DEPR.= 720002	
29	ENTER	COST? _	Processing can be repeated.
30	ENTER	>	Press this key to end processing.

PROGRAM TITLE	DEPRECIATION	PROGRAM NO. P5–D–12	3

[Key Operation Procedure] : Fixed amount method.

Step No.	Input	Display	Remarks
1	DEF B	COST ?	
2	900000 ENTER	YEAR OF LIFE? _	
3	5 ENTER	DEPR. MONTH? _	
4	6 ENTER	REM. RATE (%)? _	
5	10 ENTER	DEPR. = 81000	
6	ENTER	UNDEPR. = 819000	
7	ENTER	COST?	
8	720000 ENTER	YEAR OF LIFE? _	
9	25 ENTER	DEPR. MONTH? _	
10	8 ENTER	REM. RATE (%) ? _	
11	10 ENTER	DEPR.= 17280	
12	ENTER	UNDEPR. = 702720	
13	ENTER	COST? _	Press this key for the print-out of totals.
14	ENTER	TTL COST = 1620000	
15	ENTER	TTL DEPR. = 98280	
16	ENTER	TTL UNDEPR. = 1521720	
17	ENTER	>	

PROGRAM TITLE	DEPRECIATION	PROGRAM NO. P5– D–12	4

[Program List]

```
 10:"A":CLEAR :
    WAIT
 20:INPUT "COST?";
    A:GOTO 30
 25:END
 30:INPUT "NO. OF
    TIMES?";B
 40:INPUT "REM. RA
    TE(%)?";O
 50:IF (O<5)+(O>99
    )=1GOTO 40
 60:C=1-(O/100)^(1
    /B)
 70:D=INT (C*10^5+
    .5)/10^5
 80:E=0
 90:FOR I=1TO B
100:F=INT (D*A)
110:E=E+F
120:A=A-F
130:PRINT I;"   DE
    PR.=";F
150:PRINT I;"   UN
    DEPR.=";A
160:NEXT I
170:PRINT "TTL DEP
    R.=";E
200:GOTO 20
500:"B":CLEAR :
    WAIT
510:INPUT "COST?";
    E:GOTO 520
515:GOTO 610
520:INPUT "YEAR OF
     LIFE?";F
530:INPUT "DEPR. M
    ONTH?";G
535:INPUT "REM. RA
    TE(%)?";H
540:IF (H<5)+(H>99
    )=1GOTO 535
546:H=(100-H)/100
550:D=INT (E*H/F*G
    /12)
560:A=A+D:B=E+B
575:PRINT "DEPR.="
    ;D
580:PRINT "UNDEPR.
    =";E-D
590:GOTO 510
610:PRINT "TTL COS
    T=";B
615:PRINT "TTL DEP
    R.=";A
620:PRINT "TTL UND
    EPR.=";B-A
65279:END


STATUS 1
                586
```

[Memory Contents]

(Fixed rate method)

A	Acquisition cost
B	No. of times
C	Depreciation rate
D	
E	Total depreciation amount
F	Depreciation amount
G	
H	
I	√
J	
K	
L	
M	
N	
O	Remaining rate
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

(Fixed amount method)

A	Total depreciation amount
B	Total acquisition cost
C	
D	Depreciation amount
E	Total depreciation/ Acqusition cost
F	Years of life
G	Depreciation date
H	Remaining rate
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

SHARP

PROGRAM TITLE	ALLOTMENT CALCULATION	PROGRAM NO. P5– D–15	1

CE–150 required

[Outline]

With the indexes sequentially keyed-in, this program lets you proportion the value to be allotted. It also totals the indexes, as well as calculating the unit allotment value.

[Operating Guide]

For 8 items of data with 10 indexes already keyed-in, pressing only [ENTER] key when "Index 9?" is displayed enables you to process 8 data items.

(Note) The maximum number of indexes is 170.
The index printout is made to the first decimal.
Also, the allotment value of each index is printed out as an integer with round off.

[Example]

Input: Value to be allotted = 5000
Number of indexes = 3
Index (1) = 10.5
Index (2) = 120
Index (3) = 70

For the calculation result, refer to the "Printout".

[Contents] (Formulas)

Input: Value to be allotted
Number of indexes: n
Index

Output: Value to be allotted
Index total (Index 1 + Index 2 + --- + Index n)
Unit allotment value
(Value to be allotted ÷ Index total)
Index
Allotted value

(Note)
- The maximum number of digits for the input of the value to be allotted is 6 of integer.
  The maximum number of digits for the input of an index is 5 of integer.
- An error due to rounding off to the integer is adjusted by using the allotted value of the final index.

PROGRAM TITLE	ALLOTMENT CALCULATION	PROGRAM NO. P5– D–15	2

[Printout]

```
VALUE TO BE ALLOTD
  .......
                5000
INDEX TTL........
               200.5
UNIT ALLOTD VALUE
        24.93765586

       INDEX/ALLOTM
 1      10.5     262
 2     120.0    2993
 3      70.0    1745
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	VALUE TO BE ALLOTD?_	
2	5000 ENTER	NO. OF INDEXES ?–	
3	3 ENTER	INDEX 1 ?	
4	10.5 ENTER	INDEX 2 ?	
5	120 ENTER	INDEX 3 ?	
6	70 ENTER	>	

PROGRAM TITLE	ALLOTMENT CALCULATION	PROGRAM NO. P5– D–15	3

[Program List]

```
 10:"A":CLEAR
 20:INPUT "VALUE T
    O BE ALLOTD?";
    A
 30:INPUT "NO. OF
    INDEXES?";B
 40:C=B-1:DIM H(C)
 50:FOR D=0TO C
 60:E=D+1
 70:USING :PAUSE "
    INDEX ";E
 80:INPUT H(D):
    GOTO 150
 90:B=E-1:GOTO 200
150:F=H(D)+F
160:NEXT D
200:G=A/F
210:USING :LPRINT
    "VALUE TO BE A
    LLOTD  .......
    "
220:USING :LPRINT
    A
230:USING :LPRINT
    "INDEX TTL....
    ...."
240:USING :LPRINT
    F
250:USING :LPRINT
    "UNIT ALLOTD V
    ALUE"
260:USING :LPRINT
    G
270:LF 1
280:USING :LPRINT
    "      INDEX/A
    LLOTM"
290:C=B-1
300:FOR D=0TO C
310:E=D+1
315:I=INT (G*H(D)+
    .5)
316:IF D=CLET I=A-
    J:GOTO 320
317:J=J+I
320:USING :LPRINT
    USING "###";E;
    USING "#######.
    #";H(D);USING
    "########";I
330:NEXT D
340:END
STATUS 1
                483
```

[Memory Contents]

A	Value to be allotted
B	No. of Indexes
C	
D	
E	
F	Index Total
G	Unit Allotted Value
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
H (C)	Index

SHARP

PROGRAM TITLE	VOLUME AND WEIGHT UNIT CONVERSION	PROGRAM NO. P5– D–16	1

CE–150 required

[Outline]

This program performs unit conversions in volume or weight.

[Operating Guide]

DEF A : With these keys pressed, selection is made for either volume or weight, and prints out "Unit Item Table".

DEF B : Pressing the keys makes a unit conversion in weight or volume selected at A.

Input : Unit Code to be converted
Conversion Unit Code
Data to be converted

Output: Converted Data

[Example]

Volume		Weight	
CUBIC CENTIM	1000	GRAM	3750
CUBIC METER	0.001	TON	0.00375
LITER	1	GRAIN	57870.4
GALLON	0.26417	OUNCE	132.275
CUBIC INCH	61.0237	POUND	8.2672
CUBIC FEET	0.03532	USA. TON	0.00413

Ex.) How many gallons are equivalent to 10 liters?

How many cubic centimeters are equivalent to 1 gallon?

Ex.) How many grams are equivalent to one ounce?

How many grams are equivalent to one pound?

[Contents] (Formulas)

$$\text{Data after Conversion} = \frac{\text{Data before Conversion}}{\text{Unit Value before Conversion}} \times \text{Unit Value after Conversion}$$

PROGRAM TITLE	VOLUME AND WEIGHT UNIT CONVERSION	PROGRAM NO. P5–D–16	2

[Printout]

```
VOLUME                    WEIGHT
UNIT-------NUMBER         UNIT-------NUMBER

CUBIC METER               GRAM --------1
  (C.M) ------1           TON ---------2
CUBIC CENTIM.             GRAIN
  (C.CM) -----2             (GRN) -----3
LITER                     OUNCE
  (L) --------3             (ONC) -----4
GALLON                    POUND
  (GL) -------4             (PND) -----5
CUBIC INCH                USA.TON
  (C.I) ------5             (U.TN) ----6
CUBIC FEET
  (C.F) ------6


L     10                  ONS   1
GL    2.6417              GRAM 28.35002835

GL    1                   PND   1
C.CM 3785.441193          GRAM 453.5997678
```

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	VOLUME/WEIGHT(V/W)?	
2	V ENTER	>	Ends after the table print out.
3	DEF B	UNIT? – UNIT	
4	3 ENTER	UNIT3 – UNIT?	
5	4 ENTER	DATA=–	
6	10 ENTER	UNIT? – UNIT	
7	4 ENTER	UNIT4 – UNIT?	
8	2 ENTER	DATA=–	
9	1 ENTER	UNIT? – UNIT	
10	ENTER	>	Pressing this key ends processing.

PROGRAM TITLE	VOLUME AND WEIGHT UNIT CONVERSION	PROGRAM NO. P5– D–16	3

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	VOLUME/WEIGHT(V/W)?	
2	W ENTER	>	Table output
3	DEF B	UNIT? – UNIT	
4	4 ENTER	UNIT 4 – UNIT?	
5	1 ENTER	DATA=–	
6	1 ENTER	UNIT? – UNIT	
7	5 ENTER	UNIT5 – UNIT ?	
8	1 ENTER	DATA=–	
9	1 ENTER	UNIT? – UNIT	
10	ENTER	>	Pressing this key ends program.

[Program List]

```
 10:"A":CLEAR :DIM
    X(5),A$(5):CLS
 15:INPUT "VOLUME/
    WEIGHT?(V/W)";
    N$:GOTO 25
 20:GOTO 420
 25:IF (N$="V")+(N
    $="W")<>1GOTO
    15
 50:IF N$="V"GOTO
    250
 60:GOTO 340
250:LF 1:LFRINT "V
    OLUME"
255:LPRINT "UNIT--
    -----NUMBER"
260:LF 1
270:LPRINT "CUBIC
    METER ":X(0)=0
    .001
275:LPRINT "  (C.M
    ) ------1"
280:LPRINT "CUBIC
    CENTIM. ":X(1)
    =1000
285:LPRINT "  (C.C
    M) -----2"
290:LPRINT "LITER
    ":X(2)=1
295:LPRINT "  (L)
    --------3"
300:LPRINT "GALLON
     ":X(3)=0.2641
    7
305:LPRINT "  (GL)
     -------4"
310:LPRINT "CUBIC
    INCH ":X(4)=61
    .0237
315:LPRINT "  (C.I
    ) ------5"
320:LPRINT "CUBIC
    FEET ":X(5)=0.
    03532
325:LPRINT "  (C.F
    ) ------6"
326:A$(0)="C.M ":A
    $(1)="C.CM":A$
    (2)="L    "
327:A$(3)="GL  ":A
    $(4)="C.I ":A$
    (5)="C.F "
330:LF 8:END
```

(To be continued)

PROGRAM TITLE	VOLUME AND WEIGHT UNIT CONVERSION	PROGRAM NO. P5–D–16	4

[Program List]

```
340:LF 1:LPRINT "W
    EIGHT"
345:LPRINT "UNIT--
    -----NUMBER"
350:LF 1
360:LPRINT "GRAM -
    -------1":X(0)
    =3750
370:LPRINT "TON --
    -------2":X(1)
    =0.00375:A$(1)
    ="TON "
380:LPRINT "GRAIN
    ":X(2)=57870.4
385:LPRINT "  (GRN
    ) -----3"
390:LPRINT "OUNCE"
    :X(3)=132.275
395:LPRINT "  (ONC
    ) -----4"
400:LPRINT "POUND
    ":X(4)=8.26720
405:LPRINT "  (PND
    ) -----5"
410:LPRINT "USA.TO
    N ":X(5)=0.004
    13
415:LPRINT "  (U.T
    N) ----6"
417:A$(0)="GRAM":A
    $(2)="GRN"
418:A$(3)="ONC ":A
    $(4)="PND ":A$
    (5)="U.TN "
420:LF 8:END
470:"B":LF 1:WAIT
    0
480:CLS :LF 1:
    PRINT "UNIT
    -UNIT";
500:CURSOR 6:INPUT
    A:GOTO 510
505:CLS :END
510:IF (A<1)+(A>6)
    <>0GOTO 480
520:CURSOR 15:
    INPUT B
525:IF (B<1)+(B>6)
    <>0GOTO 520
530:CLS :INPUT "DA
    TA=";S
540:D=S/X(A-1)*X(B
    -1)
560:LPRINT A$(A-1)
    ;
570:LPRINT S
580:LPRINT A$(B-1)
    ;
590:LPRINT D
600:D=0:GOTO 480


STATUS 1
                1220
```

[Memory Contents]

A	Number before Unit Conversion
B	Number after Unit Conversion
C	
D	Value after Unit Conversion
E	
F	
G	
H	
I	
J	Weight
K	
L	
M	
N	
O	
P	
Q	
R	
S	Input Value before Unit Conversion
T	Volume
U	
V	
W	
X	
Y	
Z	
N$	Unit Name Selection Area
X(5)	Ratio of each unit
A$(5)	Names of Units

SHARP

PROGRAM TITLE	LENGTH AND AREA UNIT CONVERSION	PROGRAM NO. P5–D–17	1

CE–150 required

[Outline]

This program converts length or area units.

[Operating Guide]

DEF A : Press these keys to select either length or area for printout of "Unit Item Table".

DEF B : These convert the length or area unit selected by the A .
Input : Unit Code to be converted
Conversion Unit Code
Data to be converted
Output: Converted Data

[Example]

Length		Area	
METER	1	SQUARE METER	1
MILLI METER	1000	ARE	0.01
INCH	39.3701	SQUARE INCH	1550.00
FEET	3.28084	SQUARE FEET	10.7639
YARD	1.09361	ACRE	0.00025
MILE	0.00062	TUBO	0.30250

Ex.) How many inches are equivalent to 10 yards?
How many yards are equivalent to 3 meters?

Ex.) How many acres are equivalent to 7 ares?

[Contents] (Formulas)

$$\text{Data after Conversion} = \frac{\text{Data before Conversion}}{\text{Unit Value before Conversion}} \times \text{Unit Value after Conversion}$$

Remark: Be noted the area unit "TUBO" is used only in Japan.

PROGRAM TITLE	LENGTH AND AREA UNIT CONVERSION	PROGRAM NO. P5–D–17	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	LENGTH/AREA? (L/A)	
2	L ENTER	>	Ends after the table output
3	DEF B	UNIT? – UNIT	
4	5 ENTER	UNIT 5 – UNIT?	
5	3 ENTER	VALUE = ?	
6	10 ENTER	UNIT? – UNIT	Printout
7	1 ENTER	UNIT 1 – UNIT?	
8	5 ENTER	VALUE = ?	
9	3 ENTER	UNIT? – UNIT	Printout
10	ENTER	>	Processing is complete with this key pressed.

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	LENGTH/AREA? (L/A)	
2	A ENTER	>	Ends after the table output
3	DEF B	UNIT? – UNIT	
4	2 ENTER	UNIT2 – UNIT?	
5	5 ENTER	VALUE = ?	
6	7 ENTER	UNIT? – UNIT	
7	ENTER	>	Processing is complete with this key pressed.

PROGRAM TITLE	LENGTH AND AREA UNIT CONVERSION	PROGRAM NO. P5– D–17	3

[Printout]

```
LENGTH                          AREA
UNIT--------NUMBER              UNIT--------NUMBER

METER                           SQUARE METER
  (M) --------1                   (S.M) ------1
MILLIMETER                      ARE ----------2
  (M.M) ------2                 SQUARE INCH
INCH ---------3                   (S.I) ------3
FEET ---------4                 SQUARE FEET
YARD ---------5                   (S.F) ------4
MILE ---------6                 ACRE ---------5
                                TSUBO
                                  (TUBO) -----6


YARD  10                        ARE  7
INCH  360.0012802               ACRE  0.175

M     3
YARD  3.28083
```

PROGRAM TITLE	LENGTH AND AREA UNIT CONVERSION	PROGRAM NO. P5–D–17	4

[Program List]

```
 10:"A":CLEAR :DIM
    X(5),A$(5)
 15:INPUT "LENGTH/
    AREA?(L/A)";N$
    :GOTO 25
 20:END
 25:IF (N$="L")+(N
    $="A")<>1GOTO
    15
 30:IF N$="A"GOTO
    160
 70:LF 1
 75:LPRINT "LENGTH
    "
 77:LPRINT "UNIT--
    ------NUMBER"
 80:LF 1
 90:LPRINT "METER
    ":X(0)=1
 95:LPRINT "  (M)
    --------1"
100:LPRINT "MILLIM
    ETER ":X(1)=10
    00
105:LPRINT "  (M.M
    ) ------2"
110:A$(2)="INCH ":
    LPRINT A$(2)+"
    ---------3":X(
    2)=39.3701
120:A$(3)="FEET ":
    LPRINT A$(3)+"
    ---------4":X(
    3)=3.28084
130:A$(4)="YARD ":
    LPRINT A$(4)+"
    ---------5":X(
    4)=1.09361
140:A$(5)="MILE ":
    LPRINT A$(5)+"
    ---------6":X(
    5)=0.00062
145:A$(0)="M    ":
    A$(1)="M.M  "
150:LF 8:END
160:LF 1:LPRINT "A
    REA"
165:LPRINT "UNIT--
    ------NUMBER"
170:LF 1
180:LPRINT "SQUARE
    METER ":X(0)=
    1
185:LPRINT "  (S.M
    ) ------1"
190:A$(1)="ARE ":
    LPRINT A$(1)+"
    ----------2":X
    (1)=0.01
200:LPRINT "SQUARE
    INCH ":X(2)=1
    550.00
205:LPRINT "  (S.I
    ) ------3"
210:LPRINT "SQUARE
    FEET ":X(3)=1
    0.7639
215:LPRINT "  (S.F
    ) ------4"
220:A$(4)="ACRE ":
    LPRINT A$(4)+"
    ---------5":X(
    4)=0.00025
230:LPRINT "TSUBO
    ":X(5)=0.30250
235:LPRINT "  (TUB
    O) -----6"
237:A$(0)="S.M ":A
    $(2)="S.I ":A$
    (3)="S.F ":A$(
    5)="TUBO "
240:LF 8:END
470:"B":LF 1:WAIT
    0
480:CLS :LF 1:
    PRINT "UNIT
    -UNIT";
500:CURSOR 6:INPUT
    A:GOTO 510
505:CLS :END
510:IF (A<1)+(A>6)
    <>0GOTO 480
520:CURSOR 15:
    INPUT B
525:IF (B<1)+(B>6)
    <>0GOTO 520
530:CLS :INPUT "VA
    LUE=";Z
540:D=Z/X(A-1)*X(B
    -1)
560:LPRINT A$(A-1)
    ;
570:LPRINT Z
580:LPRINT A$(B-1)
    ;
590:LPRINT D
600:D=0:GOTO 480
```

STATUS 1

1159

[Memory Contents]

A	Code before unit conversion
B	Code after unit conversion
C	
D	Value after unit conversion
E	
F	
G	
H	
I	
J	
K	
L	
M	Area
N	Length
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	Input Value before unit conversion.
N$	Unit Name Selecting Area
X(5)	Ratio value for each unit
A$(5)	Unit name

SHARP

PROGRAM TITLE	CALCULATION OF HOUSEHOLD ACCOUNTS	PROGRAM NO. P5–D–22	1

CE–150, CE–151 and CTR required

[Outline]

Manage your budget at the beginning of each month. You input the daily expenses every day, then the total expenses up to the day and its ratio to the budget are displayed.

The monthly expenses are summed up for the year, and the yearly item by item expenditure list is also printed out.

[Operating Guide]

[DEF] [A] : Loads the sum total data up to the previous day into the machine through the cassette.
Input expenditure data (food expenses, utilities, etc.) for the day.
Prints the daily expenses, the sum total up to the day and its ratio to the budget then saves to the cassette tape.

[DEF] [B] : The monthly sum total is added to the yearly total.
The monthly budget and the sum total expenses are all cleared to zero on the cassette tape.

[DEF] [C] : Type-in the budget for the month.
The budget amounts are printed out and saved into the cassette tape.

[DEF] [D] : Prints the sum total for the year.

[DEF] [F] : Clears all areas.

Precautions : The [DEF] [B] and [C] should be operated only once a month.
The procedure of [DEF] [A] without [DEF] [C] after [DEF] [B] and [F] operations causes an error.

Note : Fifteen items are provided for expense items. To change the number of items, alter the contents of DATA statement on line Nos. 800 to 802 in the Program List.

[Example]

1. Input the budget for Nov., 1981, as follows:

Food expenses	50,000	Social expenses	5,000
Housing expenses	20,000	Transportation	5,600
Utilities	2,000	Communication expenses	2,500
Clothing expenses	1,000	Miscellaneous expenses	10,000
Insurance · Sanitation expenses	5,000	Repayment	5,000
		Tax	4,000
Educational expenses	70,000	Others	5,000
Entertainment expenses	4,000	Savings	10,000

Input the above items and budgets according to the Key Operation Procedure, and save them into the tape.

Expenses on 1981, Nov., 1 :

Food expense	2,500
Utilities	1,500
Clothing expenses	500
and so on.	

Input the above and save into the tape.

Expenses on 1981, Dec., 1 :

Food expenses	3,000
Housing expenses	15,000
and so on.	

Execute the procedure of [DEF] [B] and [DEF] [D] in succession to obtain the resulting list on the following page.

For better understandings, see the key Operation Procedure.

2. If there is no inputs into the displayed item, press only the [ENTER] key.

3. When "TAPE OUT/IN OK (Y,N)" is displayed:
Enter "Y" with the tape set to the save-in/load-to state, respectively.

4. When doing saving-in/loading-to operation, make sure to set the tape to the head of the file.

PROGRAM TITLE	CALCULATION OF HOUSEHOLD ACCOUNTS	PROGRAM NO. P5–D–22	3

[Printout]

```
* BUDGET *
 1981YEAR 11MONTH
FOOD EXP.
            50,000
HOUSING EXP.
            20,000
UTILITIES
             2,000
CLOTHING EXP.
             1,000
INS.&SANIT. EXP.
             5,000
EDUC. EXP.
            70,000
ENTTMNT EXP.
             4,000
SOCIAL EXP.
             5,000
TRANSPORTATION
             5,600
COMNCTN EXP.
             2,500
MISC. EXP.
            10,000
REPAYMENT
             5,000
TAX
             4,000
OTHERS
             5,000
SAVINGS
            10,000
TOTAL
           199,100

* DETAILS *
 11MONTH 1DAY
FOOD EXP.
             2,500
     2,500    5.0%
HOUSING EXP.
            15,000
    15,000   75.0%
UTILITIES
             1,500
     1,500   75.0%
CLOTHING EXP.
               500
       500   50.0%
INS.&SANIT. EXP.
             3,000
     3,000   60.0%
EDUC. EXP.
            30,000
    30,000   42.8%
ENTTMNT EXP.
               550
       550   13.7%
SOCIAL EXP.
             4,500
     4,500   90.0%
TRANSPORTATION
               130
       130    2.3%
COMNCTN EXP.
               300
       300   12.0%
MISC. EXP.
               500
       500    5.0%
REPAYMENT
             4,000
     4,000   80.0%
TAX
             3,500
     3,500   87.5%
OTHERS
             1,000
     1,000   20.0%
SAVINGS
            10,000
    10,000  100.0%
TOTAL
            76,980
    76,980  38.66%

* DETAILS *
 11MONTH 2DAY
FOOD EXP.
             2,500
     5,000   10.0%
TRANSPORTATION
               130
       260    4.6%
TOTAL
             2,630
    79,610  39.98%
```

```
* BUDGET *
 1981YEAR 12MONTH
FOOD EXP.
            50,000
HOUSING EXP.
            20,000
UTILITIES
             2,000
CLOTHING EXP.
             1,000
INS.&SANIT. EXP.
             5,000
EDUC. EXP.
            70,000
ENTTMNT EXP.
             4,000
SOCIAL EXP.
             5,000
TRANSPORTATION
             5,600
COMNCTN EXP.
             2,500
MISC. EXP.
            10,000
REPAYMENT
             5,000
TAX
             4,000
OTHERS
             5,000
SAVINGS
            10,000
TOTAL
           199,100

* DETAILS *
 12MONTH 1DAY
FOOD EXP.
             3,000
     3,000    6.0%
HOUSING EXP.
            15,000
    15,000   75.0%
UTILITIES
             1,500
     1,500   75.0%
ENTTMNT EXP.
               500
       500   12.5%
TAX
             3,000
     3,000   75.0%
OTHERS
             4,000
     4,000   80.0%
SAVINGS
            10,000
    10,000  100.0%
TOTAL
            37,000
    37,000  18.58%

*SUM TOTAL FOR THE
 YEAR*
FOOD EXP.
             8,000
HOUSING EXP.
            30,000
UTILITIES
             3,000
CLOTHING EXP.
               500
INS.&SANIT. EXP.
             3,000
EDUC. EXP.
            30,000
ENTTMNT EXP.
             1,050
SOCIAL EXP.
             4,500
TRANSPORTATION
               260
COMNCTN EXP.
               300
MISC. EXP.
             3,500
REPAYMENT
             8,000
TAX
             3,500
OTHERS
             1,000
SAVINGS
            20,000
TOTAL
           116,610
```

PROGRAM TITLE	CALCULATION OF HOUSEHOLD ACCOUNTS	PROGRAM NO. P5–D–22	4

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF F	TAPE OUT OK (Y, N)_	Set the cassette for saving in.
2	Y ENTER	>	Saving the data into the tape is over.
3	DEF C	TAPE IN OK (Y, N)_	Set the cassette for loading in.
4	Y ENTER	HOUSEHOLD	After a moment, the file name is displayed.
		YEAR=_	
5	1981 ENTER	MONTH=_	
6	11 ENTER	DAY=_	
7	1 ENTER	FOOD EXP.= ?	Input Nov. budget of each item.
8	50000 ENTER	HOUSING EXP. = ?	
9	20000 ENTER	UTILITIES= ?	
10	2000 ENTER	CLOTHING EXP.= ?	
11	1000 ENTER	INS. & SANIT. EXP.=?	
12	5000 ENTER	EDUC. EXP.=?	
13	70000 ENTER	ENTTMNT EXP.=?	
14	4000 ENTER	SOCIAL EXP.=?	
15	5000 ENTER	TRANSPORTATION=?	
16	5600 ENTER	COMNCTN EXP.=?	
17	2500 ENTER	MISC. EXP.=?	
18	10000 ENTER	REPAYMENT=?	
19	5000 ENTER	TAX=?	
20	4000 ENTER	OTHERS=?	
21	5000 ENTER	SAVINGS=?	
22	10000 ENTER	TAPE OUT OK (Y, N)_	Set the cassette to save-in.
23	Y ENTER	>	Saving into the cassette tape is over.

PROGRAM TITLE	CALCULATION OF HOUSEHOLD ACCOUNTS	PROGRAM NO. P5–D–22	5

Step No.	Input	Display	Remarks
24	DEF A	TAPE IN OK (Y, N)_	Set the cassette tape for loading.
25	Y ENTER	HOUSEHOLD	After a moment, the file name is displayed.
		YEAR=–	
26	1981 ENTER	MONTH=–	
27	11 ENTER	DAY=–	
28	1 ENTER	FOOD EXP.= ?	
29	2500 ENTER	HOUSING EXP.= ?	
30	15000 ENTER	UTILITIES= ?	
31	1500 ENTER	CLOTHING EXP.= ?	
32	500 ENTER	INS. & SANIT. EXP.=?	
33	3000 ENTER	EDUC. EXP.=?	
34	30000 ENTER	ENTTMNT EXP.=?	
35	550 ENTER	SOCIAL EXP.=?	
36	4500 ENTER	TRANSPORTATION=?	
37	130 ENTER	COMNCTN EXP.=?	
38	300 ENTER	MISC. EXP.=?	
39	500 ENTER	REPAYMENT=?	
40	4000 ENTER	TAX=?	
41	3500 ENTER	OTHERS=?	
42	1000 ENTER	SAVINGS=?	
43	10000 ENTER	TAPE OUT OK (Y, N)_	Set the cassette for saving-in.
44	Y ENTER	>	Saving into the cassette tape is over.

PROGRAM TITLE	CALCULATION OF HOUSEHOLD ACCOUNTS	PROGRAM NO. P5–D–22	6

Step No.	Input	Display	Remarks
45	DEF A	TAPE IN OK (Y, N)_	Set the cassette tape for loading.
46	Y ENTER	HOUSEHOLD	After a moment, the file name is displayed.
		YEAR=–	
47	1981 ENTER	MONTH=–	
48	11 ENTER	DAY=–	
49	2 ENTER	FOOD EXP.= ?	
50	2500 ENTER	HOUSING EXP.= ?	If no input
51	ENTER	UTILITIES= ?	If no input
52	ENTER	CLOTHING EXP. = ?	If no input
53	ENTER	INS. & SANIT. EXP.=?	If no input
54	ENTER	EDUC. EXP.=?	If no input
55	ENTER	ENTTMNT EXP.=?	If no input
56	ENTER	SOCIAL EXP.=?	If no input
57	ENTER	TRANSPORTATION=?	
58	130 ENTER	COMNCTN EXP.=?	If no input
⋮	⋮	⋮	⋮
⋮	⋮	SAVINGS=?	If no input
64	ENTER	TAPE OUT OK (Y, N)–	Set the cassette to save-in.
65	Y ENTER	>	Saving into the cassette tape is over.
66	DEF B	TAPE IN OK (Y, N)_	Set the cassette tape for loading.
67	Y ENTER	HOUSEHOLD	After a moment, the file name is displayed.
		TAPE OUT OK (Y, N)_	Set the cassette to save-in.
68	Y ENTER	>	Saving into the cassette tape is over.

PROGRAM TITLE	CALCULATION OF HOUSEHOLD ACCOUNTS	PROGRAM NO. P5–D–22	7

Step No.	Input	Display	Remarks
69	DEF C	TAPE IN OK (Y, N)_	Set the cassette tape for loading.
70	Y ENTER	HOUSEHOLD	After a moment, the file name is displayed.
		YEAR=_	
71	1981 ENTER	MONTH=_	
72	12 ENTER	DAY=_	
73	1 ENTER	FOOD EXP.= ?	
74	50000 ENTER	HOUSING EXP.= ?	Input the Dec. budget of each item.
⋮	⋮	⋮	
⋮	⋮	⋮	
⋮	⋮	⋮	
88	10000 ENTER	TAPE OUT OK (Y, N)_	Set the cassette to save-in.
89	Y ENTER	>	Saving into the cassette tape is over.
90	DEF A	TAPE IN OK (Y, N)_	Set the cassette tape for loading.
91	Y ENTER	HOUSEHOLD	After a moment, the file name is displayed.
		YEAR=_	
92	1981 ENTER	MONTH=_	
93	12 ENTER	DAY=_	
94	1 ENTER	FOOD EXP.= ?	Input the data to items required.
95	3000 ENTER	⋮	
⋮	⋮	⋮	
⋮	⋮	⋮	
110	10000 ENTER	TAPE OUT OK (Y, N)_	Set the cassette to save in.
111	Y ENTER	>	Saving into the cassette tape is over.

PROGRAM TITLE	CALCULATION OF HOUSEHOLD ACCOUNTS	PROGRAM NO. P5–D–22	8

Step No.	Input	Display	Remarks
112	DEF B	TAPE IN OK (Y,N)_	Set the cassette tape for loading.
113	Y ENTER	HOUSEHOLD	After a moment, the file name is displayed.
		TAPE OUT OK (Y, N)_	Set the cassette to save in.
114	Y ENTER	>	Saving into the cassette tape is over.
115	DEF D	TAPE IN OK (Y, N)_	Set the cassette tape for loading.
116	Y ENTER	HOUSEHOLD >	After a moment, the file name is displayed. Sum Total Print for the year.

PROGRAM TITLE	CALCULATION OF HOUSEHOLD ACCOUNTS	PROGRAM NO. P5–D–22	9

[Program List]

```
10:"C":CLEAR :
   WAIT 0
15:DIM B(50)
20:GOSUB 800
35:RESTORE
37:BEEP 3
39:INPUT "TAPE IN
    OK (Y,N) ";X$
41:IF X$<>"Y"GOTO
   39
43:INPUT #"HOUSEH
   OLD";B(*)
45:GOSUB 900
47:USING :LPRINT
   "* BUDGET *"
48:LPRINT B(0);"Y
   EAR";B(1);"MON
   TH"
50:FOR I=0TO 14
60:READ A$
70:PRINT A$;"=";
80:INPUT B(I+3):
   GOTO 90
85:GOTO 110
90:USING :LPRINT
   A$
95:USING :LPRINT
   USING "#######
   ,###";B(I+3)
100:B(18)=B(18)+B(
   I+3)
110:CLS :NEXT I
120:USING :LPRINT
   "TOTAL"
125:USING :LPRINT
   USING "#######
   ,###";B(18)
126:BEEP 3
127:INPUT "TAPE OU
   T OK (Y,N) ";X
   $
128:IF X$<>"Y"GOTO
   127
130:PRINT #"HOUSEH
   OLD";B(*)
140:END

200:"A":CLEAR :
   WAIT 0
205:DIM B(50)
210:GOSUB 800
225:RESTORE
230:BEEP 3
232:INPUT "TAPE IN
    OK (Y,N) ";X$
236:IF X$<>"Y"GOTO
   232
250:INPUT #"HOUSEH
   OLD";B(*)
251:GOSUB 900
252:LF 2
253:USING :LPRINT
   "* DETAILS *"
255:LPRINT B(1);"M
   ONTH";B(2);"DA
   Y"
260:FOR I=0TO 14
270:READ A$
280:PRINT A$;"=";
290:INPUT R:GOTO 3
   00
295:GOTO 340
300:B(I+19)=B(I+19
   )+R
310:B(34)=B(34)+R
320:USING :LPRINT
   A$
325:USING :LPRINT
   USING "#######
   ,###";R
328:USING :LPRINT
   USING "#######
   ,###";B(I+19);
   USING "#####.#
   ";B(I+19)/B(I+
   3)*100;"%"
330:S=S+R
340:CLS :NEXT I
350:USING :LPRINT
   "TOTAL"
351:USING :LPRINT
   USING "#######
   ,###";S
352:USING :LPRINT
   USING "#######
   ,###";B(34);
   USING "####.##
   ";B(34)/B(18)*
   100;"%"

355:BEEP 3
356:INPUT "TAPE OU
   T OK (Y,N) ";X
   $
358:IF X$<>"Y"GOTO
   356
360:PRINT #"HOUSEH
   OLD";B(*)
370:END
500:"B":CLEAR :
   WAIT 0
505:BEEP 3
506:CLS :INPUT "TA
   PE IN OK (Y,N)
    ";X$
508:IF X$<>"Y"GOTO
   506
510:DIM B(50)
520:INPUT #"HOUSEH
   OLD";B(*)
530:FOR I=19TO 34
540:B(I+16)=B(I+16
   )+B(I)
550:B(I)=0
560:NEXT I
570:FOR I=4TO 18
580:B(I)=0
590:NEXT I
592:BEEP 3
594:INPUT "TAPE OU
   T OK (Y,N) ";X
   $
596:IF X$<>"Y"GOTO
   594
600:PRINT #"HOUSEH
   OLD";B(*)
610:END
620:"F":CLEAR :
   WAIT 0
622:DIM B(50)
641:BEEP 3
642:INPUT "TAPE OU
   T OK (Y,N) ";X
   $
645:IF X$<>"Y"GOTO
   642
647:PRINT #"HOUSEH
   OLD";B(*)
650:END
```

(To be continued)

PROGRAM TITLE	CALCULATION OF HOUSEHOLD ACCOUNTS	PROGRAM NO. P5–D–22	10

[Program List]

```
700:"D":CLEAR :
    WAIT 0
701:DIM B(50)
710:GOSUB 800
720:BEEP 3
722:INPUT "TAPE IN
     OK(Y,N) ";X$
726:IF X$<>"Y"GOTO
    722
730:INPUT #"HOUSEH
    OLD";B(*)
731:LF 2
732:USING :LPRINT
    "*SUM TOTAL FO
    R THE YEAR*
735:RESTORE
740:FOR I=0TO 14
750:READ A$
760:USING :LPRINT
    A$
765:USING :LPRINT
    USING "########
    ,###";B(I+35)
770:NEXT I
780:USING :LPRINT
    "TOTAL"
785:USING :LPRINT
    USING "######,
    ###";B(50)
788:END
800:DATA "FOOD EXP
    .","HOUSING EX
    P.","UTILITIES
    ","CLOTHING EX
    P.","INS.&SANI
    T. EXP."
801:DATA "EDUC. EX
    P.","ENTTMNT E
    XP.","SOCIAL E
    XP.","TRANSPOR
    TATION"
802:DATA "COMNCTN
    EXP.","MISC. E
    XP.","REPAYMEN
    T","TAX","OTHE
    RS"
810:DATA "SAVINGS"
820:RETURN
900:INPUT "YEAR=";
    B(0)
910:INPUT "MONTH="
    ;B(1)
920:INPUT "DAY=";B
    (2)
960:RETURN
```

STATUS 1

1,971

[Memory Contents]

A	
B	
C	
D	
E	
F	
G	
H	
I	√
J	
K	
L	
M	
N	
O	
P	
Q	
R	Item by Item Amount for that day
S	Total Amount for that day
T	
U	
V	
W	
X	
Y	
Z	
A$	Item Name
X$	Receipt of Tape. OK?
B(50)	Item by Item Total Amount

SHARP

PROGRAM TITLE	INVENTORY CONTROL	PROGRAM NO. P5–D–23	1

CE–150, CE–151 and CTR required

[Outline]

All commodities are classified into blocks (up to 776 items per block) to control their stocks.

Commodity Table and Commodity List lower than the minimum stock level are made. Commodity identification is formed with 10 characters. Present stock, minimum stock and warehousing/delivery quantity is provided with up to 6 digits.

[Operating Guide]

(1) DEF F : Clears the memory, and secures data and stock file areas.

(2) DEF A : Makes and renews stock file, and DEF D makes data file.

(3) DEF B : Renews stock file according to the data file.

(4) DEF C : Displays the contents of stock file according to "Commodity Table" and "Commodity List" that are under the minimum stock level. Loading from or saving to the tape is determined at user's discretion.
However, unless DEF F is pressed again after the first DEF F operation, the contents of stock and data files in memory remain unchanged.

(5) DEF D : To input the warehousing and delivery of commodities.

[Example] : Stock control of upholsterer

(1)

Code	Item	Present Stock	MIN. Stock
1	Desk	500	250
2	Bed	100	200
3	Chair	500	350

Make a stock file and print out "Commodity Table".

(2) Add "Table 150, 100" as Code 4, and amend the item in Code 1 to "Bicycle" for the stock file.

(3)

Code	Delivery Q'tty	Q'tty in Warehouse
1	50	40
2	50	10

After making a data file and renewing the stock file, "Commodity Table" is printed out again.

Commodities less than the minimum stock quantity in the commodity table are printed out in red.

PROGRAM TITLE	INVENTORY CONTROL	PROGRAM NO. P5–D–23	2

[Contents] (Formulas)

(1) DEF A : To register a stock file (Commodity Code 1 to 75, Commodity name, stock quantity and min. stock quantity) and to renew (input commodity code, then amend and add commodity names, stocks, min. stocks). For renewal, make amendments by referring to the printed-out master table.

(2) DEF B : Collates stock and data files with the commodity code, and calculates the new stock quantity = old stocks + warehousing Q'tty – delivery Q'tty) to renew the stock file.

(3) DEF C : Prints out the commodity table and commodity list under the minimum stock level.
Enter 1 if you want to print out this, and 2 if you do not.

(4) DEF D : Makes a data file (Commodity code, warehousing quantity, delivery quantity) and prints out the data list. Also can make up to 75 data.

(5) DEF F : Clears the memory, and secures the stock file and data file areas.

[Printout] Items less than the min. Stock qtty are printed in red. Refer to page 3.

```
**  TABLE  **
  1 DESK
        500     250
  2 BED
        100     200
  3 CHAIR
        500     350


PRESENT STOCK LIST
  2 BED
        100     200


**DATA LIST**
  1      50      40
  2      50      10
```

```
**MASTER TABLE**
  1 DESK
        500     250
  2 BED
        100     200
  3 CHAIR
        500     350


**  TABLE  **
  1 DESK
        490     250
  2 BICYCLE
         60     200
  3 CHAIR
        500     350
  4 TABLE
        150     100


PRESENT STOCK LIST
  2 BICYCLE
         60     200
```

PROGRAM TITLE	INVENTORY CONTROL	PROGRAM NO. P5–D–23	3

[Key Operation Procedure] (1)

Step No.	Input	Display	Remarks
1	DEF F	MEMORY CLEAR >	

[Key Operation Procedure] (2)

Step No.	Input	Display	Remarks
1	DEF A	REGISTER=1 RENEWAL=2	
2	1 ENTER	CODE=–	
3	1 ENTER	COMMODITY NAME=–	
4	DESK ENTER	STOCK QTTY=–	
5	500 ENTER	MIN. STOCK=–	
6	250 ENTER	CODE=–	
7	2 ENTER	COMMODITY NAME=–	
8	BED ENTER	STOCK QTTY=–	
9	100 ENTER	MIN. STOCK=–	
10	200 ENTER	CODE=–	
11	3 ENTER	COMMODITY NAME=–	
12	CHAIR ENTER	STOCK QTTY=–	
13	500 ENTER	MIN. STOCK=–	
14	350 ENTER	CODE=–	
15	ENTER	MASTER-TAPE OUT OK=1 NO=2 ?	Pressing only this key ends registration.
16	1 ENTER	>	Set the tape to the cassette to secure the tape-saving state.
1	DEF C	MASTER-TAPE IN OK=1 NO=2 ?	
2	2 ENTER	TABLE YES=1 NO=2 ?	Printouts the table.
3	1 ENTER	STOCK LIST YES=1 NO=2 ?	Printouts the commodity list less than the minimum stocks.
4	1 ENTER	>	

[Key Operation Procedure] (3)

Step No.	Input	Display	Remarks
1	DEF D	CODE= _	
2	1 ENTER	DELIVERY= _	
3	50 ENTER	WAREHOUSING= _	
4	40 ENTER	CODE= _	
5	2 ENTER	DELIVERY= _	
6	50 ENTER	WAREHOUSING= _	
7	10 ENTER	CODE= _	
8	ENTER	DATA-TAPE OUT OK=1 NO=2	Set the tape to the cassette to secure the tape-saving state.
9	1 ENTER	>	
1	DEF A	REGISTER=1 RENEWAL=2	
2	2 ENTER	MASTER-TAPE IN OK=1 NO=2 ?	Set the master tape to the cassette for the tape-loading state.
3	1 ENTER	CODE= _	Prints out the master table.
4	4 ENTER	COMMODITY NAME= _	New data
5	TABLE ENTER	STOCK Q TTY= _	
6	150 ENTER	MIN. STOCK= _	
7	100 ENTER	CODE= _	
8	2 ENTER	COMMODITY NAME= _	Code to be amended.
9	BICYCLE ENTER	STOCK Q TTY= _	
10	ENTER	MIN. STOCK= _	Pressing only this key does not amend the data.
11	ENTER	CODE= _	
12	ENTER	MASTER-TAPE OUT OK=1 NO=2 ?	
13	2 ENTER	>	

[Key Operation Procedure] (4)

Step No.	Input	Display	Remarks
1	DEF B	MASTER-TAPE IN OK=1 NO=2 ?	
2	2 ENTER	DATA-TAPE IN OK=1 NO=2	Set the tape to the cassette to secure the tape-loading state.
3	1 ENTER	MASTER-TAPE OUT OK=1 NO=2 ?	Set the master tape to the cassette for the tape-saving state.
4	1 ENTER	>	

[Key Operation Procedure] (5)

Step No.	Input	Display	Remarks
1	DEF C	MASTER-TAPE IN OK=1 NO=2 ?	
2	2 ENTER	TABLE YES=1 NO=2 ?	Prints out the table.
3	1 ENTER	STOCK LIST YES=1 NO=2 ?	Prints out the commodity list less than the minimum stocks.
4	1 ENTER	>	

PROGRAM TITLE	INVENTORY CONTROL	PROGRAM NO. P5-D-23	6

[Program List]

```
 10:"A":INPUT "REG
    ISTER=1 RENEWA
    L=2";C
 20:IF (C=1)+(C=2)
    <>1GOTO 10
 30:IF C=1GOTO 110
 50:GOSUB 700:
    LPRINT "**MAST
    ER TABLE**"
 60:FOR I=0TO M
 70:IF A$(I)<>""
    GOSUB 800
 90:NEXT I:GOSUB 9
    00
110:INPUT "CODE=";
    B:GOTO 160
120:GOSUB 750:END
160:IF B<1GOTO 110
163:IF B>M+1GOTO 1
    10
165:INPUT "COMMODI
    TY NAME=";B$:A
    $(B-1)=B$
170:INPUT "STOCK Q
    TTY=";E:A(0,(B
    -1))=E
180:INPUT "MIN. ST
    OCK=";E:A(1,(B
    -1))=E
190:GOTO 110
200:"B":GOSUB 700
210:INPUT "DATA-TA
    PE IN OK=1 NO=
    2";C
220:IF (C=1)+(C=2)
    <>1GOTO 210
230:IF C=2GOTO 260
250:INPUT #"DATA";
    D(*)
260:FOR I=0TO N
265:IF D(2,I)=0
    GOTO 300
270:K=D(2,I)-1:IF
    K>MGOTO 300
280:A(0,K)=A(0,K)-
    D(0,I)+D(1,I)
300:NEXT I
310:GOSUB 750:END
400:"C":GOSUB 700
430:INPUT "TABLE Y
    ES=1 NO=2?";C
440:IF (C=1)+(C=2)
    <>1GOTO 430
450:IF C=2GOTO 540
460:LPRINT "**  TA
    BLE  **"
470:FOR I=0TO M
475:IF A$(I)=""
    GOTO 510
480:IF A(1,I)>A(0,
    I)COLOR 3
490:GOSUB 800
500:IF A(1,I)>A(0,
    I)COLOR 0
510:NEXT I:GOSUB 9
    00
540:INPUT "STOCK L
    IST YES=1 NO=2
    ?";C
550:IF (C=1)+(C=2)
    <>1GOTO 540
560:IF C=2GOTO 620
570:LPRINT "PRESEN
    T STOCK LIST"
580:FOR I=0TO M
590:IF A(1,I)<=A(0
    ,I)GOTO 610
600:GOSUB 800
610:NEXT I:GOSUB 9
    00
620:END
630:"D":USING :
    LPRINT "**DATA
     LIST**"
635:FOR I=0TO N
640:INPUT "CODE=";
    D(2,I):GOTO 65
    0
645:GOTO 670
650:IF D(2,I)<1
    GOTO 640
651:IF D(2,I)>M+1
    GOTO 640
653:INPUT "DELIVER
    Y=";D(0,I)
655:INPUT "WAREHOU
    SING=";D(1,I)
657:USING :LPRINT
    USING "###";D(
    2,I);USING "##
    #####";D(0,I);
    USING "#######
    ";D(1,I)
660:NEXT I
670:GOSUB 900:
    GOSUB 850:END
680:"F":CLEAR :M=7
    5:N=75:DIM A$(
    M),A(1,M),D(2,
    N):PAUSE "MEMO
    RY CLEAR" END
700:INPUT "MASTER-
    TAPE IN OK=1 N
    O=2";C
710:IF (C=1)+(C=2)
    <>1GOTO 700
715:IF C=2GOTO 740
730:INPUT #"MASTER
    ";A$(*),A(*)
740:RETURN
750:INPUT "MASTER-
    TAPE OUT OK=1
    NO=2";C
760:IF (C=1)+(C=2)
    <>1GOTO 750
765:IF C=2GOTO 780
770:USING :PRINT #
    "MASTER";A$(*)
    ,A(*)
780:RETURN
800:LPRINT USING "
    ###";I+1;" ";
    USING "&&&&&&&
    &&&";A$(I)
810:USING :LPRINT
    "    ";USING "#
    ######";A(0,I)
    ;USING "######
    #";A(1,I):
    USING :RETURN
850:INPUT "DATA-TA
    PE OUT OK=1 NO
    =2";C
860:IF (C=1)+(C=2)
    <>1GOTO 850
870:IF C=1PRINT #"
    DATA";D(*):
    RETURN
900:LF 2:RETURN
```

```
STATUS 1
                    1612
```

PROGRAM TITLE	INVENTORY CONTROL	PROGRAM NO. P5–D–23	7

[Memory Contents]

A		A$		A$(M)	Master Commodity Name
B	Master Code No.	B$	Commodity Name Input Area	A(I, M)	Master Present Stock Master min. Stock.
C	√	C$		D(Z, N)	Data Delivery Data No. Data Warehousing
D		D$			
E	Numerical Figure Input Area	E$			
F		F$			
G		G$			
H		H$			
I	Loop Counter	I$			
J	Loop Counter	J$			
K	√	K$			
L		L$			
M	Number of Master Commodities	M$			
N	Number of Data Commodities	N$			
O		O$			
P		P$			
Q		Q$			
R		R$			
S		S$			
T		T$			
U		U$			
V		V$			
W		W$			
X		X$			
Y		Y$			
Z		Z$			

SHARP

PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5– D–24	1

CE-150, CE-151 and CTR × 2 required

[Outline]

This program calculates individual total, individual average, class total and class average marks of tests in five subjects for each class (up to 45 students), and it arranges the marks in order from the highest to the lowest and prints it out.
The program also makes a frequency distribution table (histogram) of all students.

[Operating Guide]

[DEF] [D] : Clears all the memories, setting all school total to zero.
Input the interval and number of intervals, start point of the historgram.

[DEF] [A] : Registers and renews each name.
This key operation prints out the class table when renewing. With the printed out codes, make amendments or add names by using the codes.
The codes can be up to 45 (number of students per class).

[DEF] [B] : Input the mark for each subject by pressing [ENTER] key after the code and name are displayed.
In the case of amendment, enter only necessary subjects. Pressing only [ENTER] key skips a subject.

[DEF] [C] : Prints out the ranking list by class, the whole school average, variance, and frequency distribution upon completion of the class processing.

[Example]

(1) With the scores of two classes for five subjects, make the ranking list by class and frequency distribution table.
Class AAA 6 students
Class BBB 4 students

The [DEF] [D] clears the total area of the memory. Then, repeat the [DEF] [A], [DEF] [B], [DEF] [C] in this sequence by the number of classes.

(2) Load the tape made in the above (1) procedure to correct and add names and/or marks. Then, make class-by-calss ranking list and frequency distribution table again.
Class AAA Change of names
Class BBB Addition of one student

Clear the total area again, using the [DEF] [D], then [DEF] [A], [C] for class AAA, and [DEF] [A], [B], [C] for class BBB are used to correct and add, then to printout class-by-class ranking list.

PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5–D–24	2

[Contents] (Formulas)

(1) • The formula for variance is as follows:

$$\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}$$

where n=number of class students or total school students
i = number of subjects
$\bar{x}$ = class average mark or total school average
x_i = marks of subjects

* Variance is printed out by rounded off to three decimal place.

• Contents of print-outs
Class name
Code, Name, Marks by subjects, Individual total, Individual average, Class total, Class average, Class variance, All school total, average and variance of all school, Frequency distribution (shown by the average marks of five subjects)

• Up to 10 classes can be handled.

(2) • Input items necessary to make the frequency distribution are as follows:
Interval = 10
Start point = 0
Number of intervals = 5

Only when the start point begins with 0, the difference between the first and next start points is "Interval + 1".
The number of intervals is up to 20.

(Ex.)

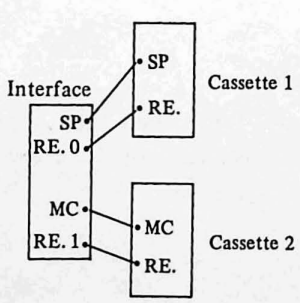
0
11
21
31
41

• Make the file by the class and save it into the tape.
The file name is identical to the class name.

• Student name should be less than 14 characters.

(3) • Load data into the machine using DEF A , B , or C , and save into the tape by DEF A , or B .

• To do this program, connect the cassettes as illustrated below.



PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5–D–24	3

[Printout]

The histogram is printed in color. Refer to Page 3.

```
** CLASS LIST **
  1 AB
  2 CD
  3 EF
  4 GH
  5 IJ
** MERIT ORDER **
AAA CLASS  LIST
  1 CD
N. LANG.   100
MATH       100
ENG.       100
HIST.      100
SCIENCE    100
TOTAL 500
AVG 100

  2 GH
N. LANG.   100
MATH       100
ENG.        80
HIST.       90
SCIENCE     60
TOTAL 430
AVG 86

  3 AB
N. LANG.    80
MATH        90
ENG.        40
HIST.       78
SCIENCE     80
TOTAL 368
AVG 74

  4 IJ
N. LANG.    50
MATH        45
ENG.        60
HIST.       70
SCIENCE     55
TOTAL 280
AVG 56

  5 EF
N. LANG.    10
MATH        25
ENG.        60
HIST.       35
SCIENCE     20
TOTAL 150
AVG 30

CLASS TTL 1728
CLASS AVERAGE 69
VARIANCE 741.25
```

```
** CLASS LIST **
  1 KL
  2 MN
  3 OP
  4 QR
  5 ST


** MERIT ORDER **
BBB CLASS  LIST
  1 MN
N. LANG.    90
MATH        95
ENG.        95
HIST.      100
SCIENCE     95
TOTAL 475
AVG 95

  2 ST
N. LANG.    45
MATH        60
ENG.        85
HIST.       75
SCIENCE     95
TOTAL 360
AVG 72

  3 QR
N. LANG.    65
MATH        85
ENG.        75
HIST.       95
SCIENCE     35
TOTAL 355
AVG 71

  4 KL
N. LANG.    50
MATH        50
ENG.        55
HIST.       45
SCIENCE     60
TOTAL 260
AVG 52

  5 OP
N. LANG.    10
MATH        25
ENG.        35
HIST.       50
SCIENCE     65
TOTAL 185
AVG 37

CLASS TTL 1635
CLASS AVERAGE 65
VARIANCE 484.5
```

```
AVG. OF ALL= 67

VARIANCE 8

HISTOGRAM
 0
 6
11
16
21
26  1
31
36  1
41
46
51  1
56  1
61
66
71  3
76
81
86  1
91  1
96  1
```

PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5– D–24	4

[Key Operation Procedure] (1)

Step No.	Input	Display	Remarks
1	DEF D	INTERVAL OF HISTOGRAM?–	
2	5 ENTER	START POINT?–	
3	0 ENTER	NO. OF INTERVALS?–	
4	20 ENTER	>	

[Key Operation Procedure] (2)

Step No.	Input	Display	Remarks
1	DEF A	REGISTER=1/CHANGE=2 ?	
2	1 ENTER	CLASS NAME=–	
3	AAA ENTER	NAME=–	
4	AB ENTER	NAME=–	
	⋮	⋮	(Repeat)
9	IJ ENTER	NAME=–	
10	ENTER	TAPE-OUT OK-1/NO=2 ?–	Processing is over with this key. If OK (1), then will be saved into the tape.
11	2 ENTER	>	

PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5– D–24	5

[Key Operation Procedure] (2)

Step No.	Input	Display	Remarks
1	DEF B	TAPE-IN OK-1/NO=2 ?–	
2	2 ENTER	1 AB	When a name is on display, start to enter a score of each subject with this key.
3	ENTER	N. LANG. 0?==> _	
4	80 ENTER	MATH. 0?==> _	When it's first time, 0 mark is displayed.
5	90 ENTER	ENG. 0?==> _	
6	40 ENTER	HIST. 0?==>_	
7	78 ENTER	SCIENCE 0?==>_	
8	80 ENTER	2 CD	
9	ENTER	N. LANG. 0?==>_	
10	100 ENTER	MATH. 0?==>	
⋮	⋮	(Repeat) ⋮	
37	70 ENTER	SCIENCE 0?==> _	
38	55 ENTER	TAPE-OUT OK-1/NO=2 ?–	Set the cassette to save in.
39	1 ENTER	>	

PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5– D–24	6

[Key Operation Procedure] (3)

Step No.	Input	Display	Remarks
1	DEF C	TAPE-IN OK-1/NO=2 ? _	
2	2 ENTER	WHOLE OK=1/NO=2 ? _	Prints the list by class.
3	2 ENTER	>	Input 2 since not all classes are over.
	(Repeat DEF A to DEF C by the number of classes.)		
1	DEF C	TAPE-IN OK-1/NO=2 ? _	
2	2 ENTER	WHOLE OK=1/NO=2 ? _	Prints the list by class.
3	1 ENTER	>	Upon completion of all classes, input 1. The average mark of all and frequency distribution are printed out.

PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5– D–24	7

[Key Operation Procedure] (4)

Step No.	Input	Display	Remarks
1	DEF A	REGISTER=1/CHANGE=2 ?–	(Modification process)
2	2 ENTER	TAPE-IN OK=1/NO=2 ? _	Set Class BBB tape to the cassette for data loading-in.
3	1 ENTER	CLASS NAME=–	
4	BBB ENTER	CODE=–	Class List is printed out.
5	5 ENTER	NAME=–	(New)
6	KL ENTER	CODE=–	
7	ENTER	TAPE-OUT OK-1/NO=2 ?–	Processing is over with this key.
8	2 ENTER	>	
1	DEF B	TAPE-IN OK-1/NO=2 ?–	(Correction of Marks)
2	2 ENTER	1 KL	
3	ENTER	N. LANG. 90?==> _	Mark before correction is displayed.
4	ENTER	MATH. 95?==> _	Enter new score, if needed to be modified, and press this key with no input if no correction is necessary.
5	90 ENTER		
	⋮	(Repeat) ⋮	
31	75 ENTER	SCIENCE 0?==> _	
32	95 ENTER	TAPE-OUT OK-1/NO=2 ?–	Set Class BBB tape to the cassette to save.
33	1 ENTER	>	

PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5– D–24	8

[Program List]

```
10:"A":INPUT "REG
   ISTER=1/CHANGE
   =2 ?";E
15:IF (E=1)+(E=2)
   <>1GOTO 10
18:IF E=2GOTO 60
21:A=0:FOR I=0TO
   P
22:B$(I)=""
23:FOR J=0TO Q+1
24:B(J,I)=0
25:NEXT J
26:NEXT I
28:INPUT "CLASS N
   AME=";A$
30:FOR I=0TO P
35:INPUT "NAME=";
   B$(I):GOTO 45
40:I=I-1:GOTO 100
45:A=A+1
50:NEXT I:GOTO 10
   0
60:GOSUB 700
65:LPRINT "** CLA
   SS LIST **":
   GOSUB 800
70:INPUT "CODE=";
   I:GOTO 80
75:GOTO 100
80:IF (I<1)+(I>P+
   1)=1GOTO 70
88:IF B$(I-1)=""
   LET A=A+1
90:INPUT "NAME=";
   B$(I-1)
95:GOTO 70
100:GOSUB 750:END
110:"B":GOSUB 700:
   CLS :WAIT :FOR
   I=0TO A-1
120:CLS :Z$=STR$ (
   I+1)+" "+B$(I)
125:PRINT Z$
130:WAIT 0:FOR J=0
   TO Q
140:CLS :PRINT D$(
   J);" ";
145:PRINT B(J,I);
150:INPUT " ? ==>"
   ;B(J,I)
170:NEXT J:WAIT :
   CLS
190:NEXT I
200:GOSUB 750
210:END
220:"C":F=0:D=D+1
230:GOSUB 700:FOR
   I=0TO A-1:FOR
   J=0TO Q
235:B(Q+1,I)=B(Q+1
   ,I)+B(J,I):
   NEXT J
240:F=F+B(Q+1,I):
   NEXT I
280:GOSUB 600
320:LPRINT "** MER
   IT ORDER **"
323:LPRINT A$;" CL
   ASS  LIST"
325:M=INT (F/A/(Q+
   1)+.5)
330:G=1:GOSUB 800
335:INPUT "WHOLE O
   K=1/NO=2?";E
340:IF (E=1)+(E=2)
   <>1GOTO 335
345:IF E=2GOTO 365
350:S=INT (C/D+.5)
   :LPRINT "AVG.
   OF ALL=";S:LF
   1
352:R=0:FOR I=0TO
   D-1:R=INT (D(I
   )-S)^2+R:NEXT
   I
353:IF D=1LF 2:
   GOTO 360
354:N=R/(D-1):N=
   INT (N*10^3+.5
   )/10^3
355:LPRINT "VARIAN
   CE";N:LF 2
360:GOSUB 900
365:END
400:"D":CLEAR :P=4
   4:Q=4:K=9:DIM
   B$(P),B(Q+1,P)
   ,D$(Q)*9,D(K)
410:D$(0)="N. LANG
   .":D$(1)="MATH
   ":D$(2)="ENG."
   :D$(3)="HIST."
   :D$(4)="SCIENC
   E"
430:INPUT "INTERVA
   L OF HISTOGRAM
   ?";T
435:IF (T<1)+(T>10
   0)=1GOTO 430
440:INPUT "START P
   OINT?";U
445:IF (U<0)+(U>10
   0)=1GOTO 440
450:INPUT "NO. OF
   INTERVALS?";V
455:IF (V<1)+(V>20
   )=1GOTO 450
458:DIM E(V-1),F(V
   -1)
460:FOR I=0TO V-1:
   F(I)=U:IF U=0
   LET U=U+1
465:U=U+T:NEXT I
470:END
500:FOR Z=0TO V-1
505:IF F(Z)>WGOTO
   550
510:IF Z=V-1GOTO 5
   25
515:IF F(Z+1)<=W
   GOTO 550
520:E(Z)=E(Z)+1:
   GOTO 550
525:IF F(Z)+T>W
   GOTO 520
550:NEXT Z
555:RETURN
600:FOR I=0TO A-2:
   L=I+1
610:FOR J=LTO A-1
620:IF B(Q+1,I)>=B
   (Q+1,J)GOTO 62
   7
623:C$=B$(I):B$(I)
   =B$(J):B$(J)=C
   $
625:FOR O=0TO Q+1:
   H=B(O,I):B(O,I
   )=B(O,J):B(O,J
   )=H:NEXT O
627:NEXT J
629:NEXT I
630:RETURN
700:INPUT "TAPE-IN
    OK=1 / NO=2?"
   ;H
705:IF (H=1)+(H=2)
   <>1GOTO 700
710:IF H=2GOTO 725
715:INPUT "CLASS N
   AME?";A$
720:INPUT #A$;A,B$
   (*),B(*)
725:RETURN
```

(To be continued)

PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5-D-24	9

[Program List]

```
750:INPUT "TAPE-OU
    T OK=1 / NO=2?
    ";H
755:IF (H=1)+(H=2)
    <>1GOTO 750
760:IF H=2GOTO 770
765:PRINT #-1,A$;A
    ,B$(*),B(*)
770:RETURN
800:FOR I=0TO A-1
805:LPRINT USING "
    ###";I+1;" ";B
    $(I):USING
810:IF G=0GOTO 840
815:FOR J=0TO Q
820:LPRINT USING "
    &&&&&&&&&";D$(
    J);USING "####
    ";B(J,I)
825:USING :NEXT J
830:LPRINT "TOTAL"
    ;B(Q+1,I)
833:W=INT (B(Q+1,I
    )/(Q+1)+.5)
835:LPRINT "AVG";W
    :GOSUB 500:LF
    1
840:NEXT I
845:IF G=0GOTO 860
850:LPRINT "CLASS
    TTL";F
855:R=0:LPRINT "CL
    ASS AVERAGE";M
    :C=C+M:IF K>=D
    -1LET D(D-1)=M
857:FOR O=0TO A-1:
    S=INT (B(Q+1,O
    )/(Q+1)+.5):R=
    (S-M)^2+R:NEXT
    O
858:IF A=1GOTO 860
859:N=R/(A-1):N=
    INT (N*10^3+.5
    )/10^3:LPRINT
    "VARIANCE";N
860:G=0:LF 2:
    RETURN
900:LPRINT "HISTOG
    RAM"
903:GRAPH :
    GLCURSOR (0,0)
    :SORGN :LINE (
    50,0)-(215,0):
    LINE (50,0)-(5
    0,-450)
905:S=E(0):FOR I=1
    TO U-1:IF S<E(
    I)LET S=E(I)
908:NEXT I
910:Y=0:FOR I=0TO
    U-1:X=115/S*E(
    I)
913:IF X=0GOTO 930
915:LINE (50,Y)-(X
    +50,Y-450/U),0
    ,2,B
920:COLOR 0:
    GLCURSOR (X+55
    ,Y-20):LPRINT
    E(I)
930:GLCURSOR (0,Y-
    13):LPRINT F(I
    )
940:Y=Y-450/U:NEXT
    I:TEXT :COLOR
    0:LF 2
960:RETURN


STATUS 1
                2541
```

PROGRAM TITLE	MANAGEMENT OF STUDENTS' ACHIEVEMENTS	PROGRAM NO. P5– D–24	10

[Memory Contents]

A	No. of students in Class	A$	Class name	B(Q+1, P)	Achievements
B	√	B$		B$(P)	Student names
C	Class average total	C$	√	D$(P)	Subject names
D	No. of Classes	D$		D(K)	Class averages
E	√	E$		E(V–1)	Counting the number of students in frequency distribution
F	Class Total	F$		F(V–1)	Figure of start point at each interval.
G	√	G$			
H	√	H$			
I	√	I$			
J	√	J$			
K	No. of Classes	K$			
L	√	L$			
M	Class average mark	M$			
N		N$			
O	√	O$			
P	MAX. no. of students in Class	P$			
Q	No. of subjects	Q$			
R	√	R$			
S	√	S$			
T	Interval	T$			
U	Start point	U$			
V	No. of intervals	V$			
W	Individual average marks	W$			
X	√	X$			
Y	√	Y$			
Z	√	Z$	√		

SHARP

PROGRAM TITLE	POCKET COMPUTER SCHEDULE PLANNER	PROGRAM NO. P5–D–25	1

CE–150, CE–151 and CTR required

[Outline]

Preset date, time, contents and alarm time then machine will let you know what the schedule is when the alarm time comes. An alarm sounds at the alarm time. The contents of each schedule can be up to 40 characters and number of schedules up to 30 items.

[Operating Guide]

[DEF] [N] : Used to clear all schedule contents.

[DEF] [Z] : Pocket computer schedule starts.
An alarm sound at the alarm time.
The alarm continues for one minute, and can be stopped by pressing the [!] key. The contents of the schedule will then be printed out.

[A] : Used to set current time.

[B] : Used to register schedules.

[C] : Used to print the schedules within the designated period.

[D] : Used to print all schedules for the day.

[F] : Used to print all the registered schedules.

[G] : Used to print the first schedule after the designated date.

[H] : Used to print the locked or unlocked schedules.

[K] : Used to delete the designated schedule.

[M] : Used to delete the schedules before the designated period except for the locked ones.

[L] : Used to load the schedules from tape.

[S] : Used to save the schedules to tape.

[Cautions]

- The program stops when pressing the [BREAK] key.
- Press the keys slowly.
- Connect the AC adapter to the CE–150 for program run.
- Key-in the start and end times in a 24-hour format.
- Key-in the alarm time some minutes prior to the start time.
  With no input, the minutes become 0.
- Use the [K] or [DEF] [N] to delete the locked schedules.
- With 0 minute when registering the schedule, key-in 0 and press the [ENTER].
- Key-in each of month, day, hour and minute in 2 digits.
- When the register area runs out for schedule registrations, schedules not locked and before the current time will be deleted for new registrations. With no schedules to be deleted, there displays "THERE IS NO AREA", and program run continues.

PROGRAM TITLE	POCKET COMPUTER SCHEDULE PLANNER	PROGRAM NO. P5–D–25	2

[Example 1]

[DEF] [N] : Clears all schedules. Be careful!

[DEF] [Z] : Starts the pocket computer schedule.

[A] : Set the time to 10 hours, 35 minutes on November 9.

[B] : Register the schedules.

- Conference from 9:30, November 15 to 12:00, November 15, with an alarm 20 minutes prior. Make this locked.
- Visitor from 13:00, November 20 to 17:30, November 20, with an alarm 30 minutes prior. Make this unlocked.
- Concert from 15:00, November 13 to 16:30, November 13, with an alarm 30 minutes prior. Make this locked.
- Gymnastics from 6:30, November 30 to 6:50, November 30, with an alarm 0 minute prior. Make this unlocked.

[C] : Prints all the schedules from November 15 to 12:00, November 20.

[D] : Prints the schedules for the day (November 20).

[F] : Prints all the registered schedules.

[S] : Saves the schedules on cassette tape.

[Example 2]

1. Pressing the [BREAK] key to stop the program.
2. Clearing all schedules by the [DEF] [N] operation.
3. Pressing the [DEF] [Z] keys to start the program.
4. Pressing the [B] to register the schedules.
   - Visitor from 10:00, December 10 to 12:00, December 10, with an alarm 30 minutes prior. Unlocked.
   - Party from 18:00, December 24 to 23:00, December 24, with an alarm 60 minutes prior. Locked.
5. Pressing the [G] keys to print the first schedule after December 15.
6. Pressing the [H] keys to print the locked schedules.
7. Pressing the [H] keys to print the unlocked schedules.
8. Pressing the [K] keys to delete the schedules before 10:00, December 10.
9. Pressing the [S] keys to write the schedules on cassette tape.

[Example 3]

1. Pressing the [BREAK] key to stop the program.
2. Pressing the [DEF] [N] keys to clear all schedules.
3. Pressing the [DEF] [Z] keys to start the program.
4. Pressing the [L] keys to read the schedules out written in Example 1 shown before.
5. Pressing the [M] keys to clear the schedules other than the locked schedules before November 25.
6. Pressing the [F] keys to printout all the schedules presently registered.

PROGRAM TITLE	POCKET COMPUTER SCHEDULE PLANNER	PROGRAM NO. P5–D–25	3

[Printout]

```
11/ 15FROM
11/ 20UNTILL LIST

* CONFERENCE *
START 11/15  9:30
END   11/15  12:00
ALARM  20MIN.PRIOR

* VISITOR *
START 11/20  13:00
END   11/20  17:30
ALARM  30MIN.PRIOR
```

```
ALL LIST

* CONFERENCE *
START 11/15  9:30
END   11/15  12:00
ALARM  20MIN.PRIOR

* VISITOR *
START 11/20  13:00
END   11/20  17:30
ALARM  30MIN.PRIOR

* CONCERT *
START 11/13  15:00
END   11/13  16:30
ALARM  30MIN.PRIOR

* GYMNASTICS *
START 11/30  6:30
END   11/30  6:50
ALARM  0MIN.PRIOR
```

```
12/15  0:00 ON

* PARTY *
START 12/24  18:00
END   12/24  23:00
ALARM  60MIN.PRIOR


LOCK LIST

* PARTY *
START 12/24  18:00
END   12/24  23:00
ALARM  60MIN.PRIOR


UNLOCK LIST

* VISITOR *
START 12/10  10:00
END   12/10  12:00
ALARM  30MIN.PRIOR
```

```
ALL LIST

* CONFERENCE *
START 11/15  9:30
END   11/15  12:00
ALARM  20MIN.PRIOR

* CONCERT *
START 11/13  15:00
END   11/13  16:30
ALARM  30MIN.PRIOR

* GYMNASTICS *
START 11/30  6:30
END   11/30  6:50
ALARM  0MIN.PRIOR
```

[Key Operation Procedure] (1)

Step No.	Input	Display	Remarks
1	DEF N		All schedules cleared.
		DELETION END	
2	DEF Z	11/5 16:03	Program starts.
3	A	11/5 16:03 CHANGE=1 NO CHANGE= 2	Current time displayed. If the time is correct, enter 2 to continue the program.
4	1 ENTER	?—/ ; :	
5	11 ENTER	11/? ; :	Month input
6	09 ENTER	11/09 ; ? :	Day input
7	10 ENTER	11/09 ; 10 : ?	Hour input
8	35 ENTER	11/09 : 10 : 35_	Minute input Returns to the display at the step 3 above.
9	B	? —/ ; : START	Schedule registration
10	11 ENTER	11/? ; : START	Start month input. Press the ENTER to continue the program.
11	15 ENTER	11—/ 15 ; ? : START	Start day input
12	09 ENTER	11—/ 15—; 09: ? START	Start hour input

[Key Operation Procedure] (1)

Step No.	Input	Display	Remarks
13	30 [ENTER]	/ ; : END	Start minute input
14	11 [ENTER]	11–/ ; : END	End month input
15	15 [ENTER]	11–/ 15–; : END	End day input
16	12 [ENTER]	11–/ 15–; 12: END	End hour input
17	00 [ENTER]	CONTENTS=	End minute input
18	CONFERENCE [ENTER]	ALARM TIME=_	Schedule contents input
19	20 [ENTER]	LOCK=1, UNLOCK=2?_	Alarm time input (minutes prior to)
20	1 [ENTER]		Selection Returns to the display at the step 9.
21	[C]	LIST START DATE=	Month and day inputs in 4 digits
22	1115 [ENTER]	LIST END DATE=	Month and day inputs in 4 digits
23	1120 [ENTER]		Prints out the schedules registered, then continue program.
24	[D]	11/10 9:30	Prints out the schedules for the day and continues program.
25	[F]	11/10 9:31	Prints out all registered program and continues program.
26	[S]	TAPE OUT OK (Y/N)?_	Saves schedules on cassette tape and continues program.
27	Y [ENTER]		

PROGRAM TITLE	POCKET COMPUTER SCHEDULE PLANNER	PROGRAM NO. P5–D–25	5

[Key Operation Procedure] (2)

Step No.	Input	Display	Remarks
1	BREAK		Stops program.
2	DEF N	DELETION END	Clears all schedules.
3	DEF Z	11/09 11:30	Starts program and current time displayed.
4	B		To register schedules.
5 ⋮	⋮	⋮	⋮
14	60 ENTER	LOCK=1, UNLOCK=2 ?_	LOCK UNLOCK Selection
15	1 ENTER		Returns to the display at the step 4.
16	G 12150000 ENTER	DATE, TIME=_	Month, day, hour and minute inputs in 8 digits. 2 digits each. Prints all the schedules after the input date and continues program.
17	H	LOCK=1, UNLOCK=2 ?_	LOCK UNLOCK Selection
18	1 ENTER		Prints all the locked schedules and continue program.
19	H	LOCK=1, UNLOCK=2 ?_	LOCK UNLOCK Selection
20	2 ENTER		Prints all the unlocked schedule and continues program.
21	K	DATE, TIME=_	Month, day, hour and minute inputs in 8 digits.
22	12101000 ENTER	DELETION END	After deletion, continues program.

Step No.	Input	Display	Remarks
23	[S]	TAPE OUT OK (Y/N) ?_	Save schedule contents on cassette tape and re-runs program.
24	Y [ENTER]		

[Key Operation Procedure] (3)

Step No.	Input	Display	Remarks
1	[BREAK]		Stops program.
2	[DEF] [N]	DELETION END	Clears all schedules.
3	[DEF] [Z]	11/10 9:05	Reads schedule saved in Example 1.
4	[L]	TAPE IN OK (Y/N) ?_	Month, day, hour and minute inputs in 8 digits.
5	Y [ENTER]	SCHUDELE 11/10 9:10	Display of file name
6	[M]	DATE, TIME=_	Month, day, hour and minute inputs in 8 digits.
7	11250000 [ENTER]	DELETION END	Re-runs program after deleting the schedule other than the locked before designated time.
8	[F]	11/10 ; 9:12	Prints all schedules registered and re-runs program.

PROGRAM TITLE	POCKET COMPUTER SCHEDULE PLANNER	PROGRAM NO. P5–D–25	7

[Program List]

```
  2:"S":INPUT "TAP
    E OUT OK(Y/N)?
     ";Y$:IF Y$="Y
    "GOTO 5
  4:GOTO 2
  5:PRINT #"SCHEDU
    LE";P(*),N$(*)
    :CLS :RETURN
  8:"L":INPUT "TAP
    E IN OK(Y/N)?
    ";Y$:IF Y$="Y"
    GOTO 11
 10:GOTO 8
 11:INPUT #"SCHEDU
    LE";P(*),N$(*)
    :CLS :RETURN
 20:"A":A=TIME :
    GOSUB 950:CLS
    :WAIT 130:
    PRINT A$:WAIT
    0
 50:INPUT "CHANGE=
    1/NO CHANGE=2?
     ";Z$:IF (Z$="
    1")+(Z$="2")<>
    1GOTO 50
 60:IF Z$="2"
    RETURN
 63:PRINT "   /
     ;    :    ":
    GOSUB 925:A=B*
    10000+C*100+D+
    E/100:TIME =A:
    GOTO 20
100:A=B*10000+C*10
    0+D+E/100
105:TIME =A:GOTO 2
    0
170:"B":FOR I=0TO
    28:IF P(I,0)<>
    0GOTO 240
175:CLS :PRINT "
     /    ;    :
        START":
    GOSUB 925
178:IF U=1GOTO 245
180:X=B*10000+C*10
    0+D+E/100:IF X
    <TIME GOTO 175
200:CLS :PRINT "
     /    ;    :
         END":
    GOSUB 925:IF U
    =1GOTO 200
207:Y=B*10000+C*10
    0+D+E/100:IF Y
    <XGOTO 200
212:FOR J=0TO 25:
    IF X<P(J,0)
    GOTO 220
216:IF X>P(J,1)
    GOTO 222
218:Z=1:J=26:GOTO
    222
220:IF Y>P(J,0)LET
    Z=1:J=26
222:NEXT J
225:IF Z=1LET Z=0:
    GOTO 175
226:P(I,0)=X:P(I,1
    )=Y:CLS :INPUT
    "CONTENTS=";N$
    (I):CLS :INPUT
    "ALARM TIME=";
    P(I,2)
235:CLS :INPUT "LO
    CK=1/UNLOCK=2
    ?";P(I,3):IF (
    P(I,3)=1)+(P(I
    ,3)=2)<>1GOTO
    235
240:NEXT I
245:IF U=1LET U=0:
    GOTO 290
250:H=0:K=0
255:FOR J=0TO 25:
    IF P(J,3)=1
    GOTO 275
265:IF P(J,3)=1
    GOTO 275
267:IF H=0LET H=P(
    J,0):K=J+1
270:IF H>P(J,0)LET
    H=P(J,0):K=J+1
275:NEXT J
277:IF K=0WAIT 150
    :PRINT "THERE
    IS NO AREA":
    WAIT 0:GOTO 29
    0
280:I=K-1:GOSUB 90
    0:GOTO 170
290:CLS :RETURN
300:"C":WAIT 0:CLS
    :INPUT "LIST S
    TART DATE=";G:
    GOTO 330
305:G=0:H=9999
330:CLS :INPUT "LI
    ST END DATE=";
    H:GOTO 350
350:IF (G=0)+(H=99
    99)+(G>H)=1
    GOTO 300
365:A=G*100:GOSUB
    950:LPRINT B;"
    /";C;"FROM"
366:A=H*100:GOSUB
    950:LPRINT B;"
    /";C;"UNTILL L
    IST"
370:FOR I=0TO 28:
    IF P(I,0)=0
    GOTO 410
380:IF G>INT (P(I,
    0)/100)GOTO 41
    0
390:IF H<INT (P(I,
    0)/100)GOTO 41
    0
400:GOSUB 990
410:NEXT I:LF 3:
    CLS :RETURN
```

(To be continued)

PROGRAM TITLE	POCKET COMPUTER SCHEDULE PLANNER	PROGRAM NO. P5–D–25	8

[Program List]

```
450:"D":G=INT (
    TIME /100):P=
    INT (G/100):
    LPRINT P;"MONT
    H";G-P*100;"DA
    Y"
470:FOR I=0TO 28:
    IF G<>INT (P(I
    ,0)/100)GOTO 5
    00
490:GOSUB 990
500:NEXT I:LF 3:
    CLS :RETURN
550:"F":LPRINT "AL
    L LIST":FOR I=
    0TO 28:IF P(I,
    0)=0GOTO 580
570:GOSUB 990
580:NEXT I:LF 3:
    CLS :RETURN
600:"G":G=0:H=0:
    INPUT "DATE,TI
    ME=";G
608:A=G/100:GOSUB
    950:LPRINT A$;
    "ON"
610:FOR I=0TO 28:
    IF G<INT (P(I,
    0)*100)GOTO 62
    3
620:GOTO 630
623:IF H=0LET H=P(
    I,0):K=I
625:IF H>INT P(I,0
    )LET H=INT P(I
    ,0):K=I
630:NEXT I
635:IF H=0GOTO 649
640:I=K:GOSUB 990
649:LF 3:CLS :
    RETURN
660:"H" INPUT "LOCK
    =1, UNLOCK=2";T
665:IF (T=1)+(T=2)
    <>1GOTO 660
666:IF T=1LET B$="
    LOCK ":GOTO 66
    8
667:B$="UNLOCK "
668:LPRINT B$;"LIS
    T"
670:FOR I=0TO 28:
    IF P(I,0)=0
    GOTO 680
672:IF P(I,0)=0
    GOTO 680
675:IF P(I,3)=T
    GOSUB 990
680:NEXT I:LF 3:
    CLS :RETURN
700:"K":INPUT "DAT
    E,TIME=";G:FOR
    I=0TO 28
715:IF G=INT (P(I,
    0)*100)LET I=2
    6:NEXT I:GOSUB
    900:GOTO 725
720:NEXT I
725:GOSUB 920:CLS
    :RETURN
750:"M";G=0:INPUT
    "DATE,TIME=";G
    :FOR I=0TO 28:
    IF G>INT (P(I,
    0)*100)GOTO 77
    5
770:GOTO 780
775:IF P(I,3)=2
    GOSUB 900
780:NEXT I:GOSUB 9
    20:CLS :RETURN
800:"N":CLEAR :DIM
    P(29,4),N$(29)
    *40:GOSUB 920:
    END
830:"Z":WAIT 0
836:FOR R=0TO 28
837:A=TIME :GOSUB
    950:PRINT A$
838:B$=INKEY$ :IF
    (B$="B")+(B$="
    C")+(B$="D")+(
    B$="F")+(B$="G
    ")+(B$="H")=1
    GOTO 842
839:IF (B$="M")+(B
    $="K")+(B$="A"
    )+(B$="S")+(B$
    ="L")=1GOTO 84
    2
840:GOTO 843
842:GOSUB B$
843:IF (P(R,4)=1)+
    (P(R,0)=0)=1
    GOTO 872
845:U=P(R,0)-P(R,2
    )/100:W=(U-INT
    U)*100:IF INT
    W>59LET U=P(R,
    0)+1-0.6
847:IF INT (TIME *
    100)<INT (U*10
    0)GOTO 872
855:P(R,4)=1:M=
    TIME +0.01:N=(
    M-INT M)*100
859:IF INT N>59LET
    M=M+1-0.6
861:IF TIME >MGOTO
    870
865:B$=INKEY$ :IF
    B$<>CHR$ &11
    BEEP 2:GOTO 86
    1
870:I=R:GOSUB 990:
    LF 3
872:NEXT R:GOTO 83
    6
900:P(I,0)=0:P(I,1
    )=0:P(I,2)=0:P
    (I,3)=0:P(I,4)
    =0:N$(I)="":
    RETURN
920:CLS :WAIT 150:
    PRINT "DELETIO
    N END":WAIT 0:
    RETURN
925:CURSOR 0:INPUT
    B:GOTO 927
926:U=1:GOTO 949
927:IF B>12GOTO 92
    5
928:IF B=0GOTO 925
929:CURSOR 5:INPUT
    C:GOTO 931
930:GOTO 929
931:IF C=0GOTO 929
932:IF (B=4)+(B=6)
    +(B=9)+(B=11)=
    1GOTO 938
933:IF B=2GOTO 936
934:IF C>31GOTO 92
    9
935:GOTO 940
936:IF C>29GOTO 92
    9
937:GOTO 940
```

(To be continued)

PROGRAM TITLE	POCKET COMPUTER SCHEDULE PLANNER	PROGRAM NO. P5– D–25	9

[Program List]

```
938:IF C>30GOTO 92
    9
940:CURSOR 10:
    INPUT D:GOTO 9
    44
941:GOTO 940
944:IF D>23GOTO 94
    0
945:CURSOR 15:
    INPUT E:GOTO 9
    48
946:GOTO 945
948:IF E>59GOTO 94
    5
949:RETURN
950:B=INT (A/10000
    ):C=INT ((A-B*
    10000)/100):D=
    INT (A-B*10000
    -C*100)
955:E=INT ((A-B*10
    000-C*100-D)*1
    00)
975:IF E=0LET E$="
    00":GOTO 980
976:E$=STR$ E
980:A$=STR$ B+"/"+
    STR$ C+"  "+
    STR$ D+":"+E$
985:RETURN
990:LF 1:LPRINT "*
     ";N$(I);" *":
    A=P(I,0):GOSUB
    950:LPRINT "ST
    ART ";A$:A=P(I
    ,1)
993:GOSUB 950:
    LPRINT "END
    ";A$:LPRINT "A
    LARM ";P(I,2);
    "MIN.PRIOR":
    RETURN
```

```
STATUS 1
                3375
```

[Memory Contents]

A	Time
B	Month
C	Day
D	Hour
E	Minute
F	
G	√
H	√
I	√
J	√
K	√
L	
M	√
N	√
O	
P	√
Q	
R	√
S	
T	
U	√
V	√
W	√
X	√
Y	√
Z	√

A$	Month, day, hour, minute
B$	√
S$	√
Y$	√
Z$	√
N$(i)	Contents
P(i, 0)	Start time
P(i, 1)	End time
P(i, 2)	Alarm time
P(i, 3)	Lock, unlock
P(i, 4)	Before or after of the current time

SHARP

PROGRAM TITLE	PURCHASE LEDGER GENERATION	PROGRAM NO. P5–D–26	1

CE–150, CE–151 and CTR required

[Outline]

Product numbers, quantities and prices for each supplier are to be entered on each occurence of a purchase slip. The purchase list generated gives you the total for each supplier; and with this clear picture, you can manage your purchase control more efficiently.

[Operating Guide]

[DEF] [A] : These keys are used to enter the contents of each purchese slip. A list of the input data is printed out.

[DEF] [B] : Press these keys for a list generation of the Products to each supplier now stored on the tape.

Note : Make sure that only one supplier is recorded on each tape.

[Example]

1. Purchase ledger (New) :

Supplier	"A-123"		
Product name	"A-11"	"C-33"	"D–44"
Price	1,000	5,000	1,000
Quantity	15	5	1

Key in the above according to the Key Operation Procedure. With "END (Y/N)" displayed, type in "Y". Using the [DEF] [A] keys, enter the next data. At this time, replace the tape with a new one.

(New) :

Supplier	"J-963"
Product name	"J-77"
Price	6,200
Quantity	3

Key-in the above in that order.

With the display of "END(Y/N)", enter "N" and replace the tape with the previous one. Then key in the following to complete the key operation..

Supplier	"A-123"		
Product name	"C-33"	"D-44"	"R-55"
Price	–	–	4,000
Quantity	2	1	2

A list generation for the readouts of the above two tapes in sequence will produce the printout as shown on the next page.

2. With the display of TAPE IN/OUT OK (Y/N), at the key Operation Procedure, make sure that the supplier's name is the same as that on the tape. To set the tape for saving / loading key-in "Y".

With the input of anything other than"Y/N","TAPE OK (Y/N)" is displayed again.

3. For tape input/output, make sure to set to the head of the file.

[Contents] (Formulas)

- The purchase ledger list is only the inputs given this time
- A list covers the product names, prices and quantities now stored on the tape. The quantities for the same product name are summed up in the list.
- Registrations can be up to 140 product names per supplier.

[Printout]

```
*PURCHASE LEDGER*

* A-123 *
A-11
@          1,000
*             15
=         15,000

C-33
@          5,000
*              5
=         25,000

D-44
@          1,000
*              1
=          1,000

TOTAL
              41,000

* GRAND TOTAL *
              41,000
```

```
*PURCHASE LEDGER*

* J-963 *
T-77
@          6,200
*              3
=         18,600

TOTAL
              18,600

* A-123 *
C-33
@          5,000
*              2
=         10,000

D-44
@          1,000
*              1
=          1,000

R-55
@          4,000
*              2
=          8,000

TOTAL
              19,000

* GRAND TOTAL *
              37,600
```

```
** LIST **
* A-123 *
A-11
@          1,000
*             15
=         15,000
C-33
@          5,000
*              7
=         35,000
D-44
@          1,000
*              2
=          2,000
R-55
@          4,000
*              2
=          8,000
TOTAL
              60,000

* J-963 *
T-77
@          6,200
*              3
=         18,600
TOTAL
              18,600

* GRAND TOTAL *
              78,600
```

PROGRAM TITLE	PURCHASE LEDGER GENERATION	PROGRAM NO. P5– D–26	3

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	END (Y/N)	
2	N ENTER	NEW (Y/N)	No tape for this supplier is available yet.
3	Y ENTER	SUPPLIER=	
4	A–123 ENTER	PRODUCT NAME=	Repeat
5	A–11 ENTER	PRICE=	
6	1000 ENTER	QUANTITY=	
7	15 ENTER	PRODUCT NAME=	
⋮	⋮	⋮	
13	1 ENTER	PRODUCT NAME=	Input is completed for this supplier.
14	ENTER	TAPE OUT OK (Y/N)	Set cassette tape to save.
15	Y ENTER	END (Y/N)	
16	Y ENTER	>	Total by supplier is printed.
17	DEF A	END (Y/N)	
18	N ENTER	NEW (Y/N)	
19	Y ENTER	SUPPLIER=	
20	J–963 ENTER	PRODUCT NAME=	
21	T–77 ENTER	PRICE=	
22	6200 ENTER	QUANTITY =	
23	3 ENTER	PRODUCT NAME=	
24	ENTER	TAPE OUT OK (Y/N)	Set cassette tape to save.
25	Y ENTER	END (Y/N)	
26	N ENTER	NEW (Y/N)	Tape for this supplier is available already.
27	N ENTER	SUPPLIER=	
28	A–123 ENTER	TAPE IN OK (Y/N)	Set cassette tape to load.
29	Y ENTER	A–11	Product name display.
		QUANTITY=	

[Key Operation Procedure]

Step No.	Input	Display	Remarks
30	[ENTER]	C–33	
		QUANTITY=	
31	2 [ENTER]	D–44	
		QUANTITY=	
32	1 [ENTER]	PRODUCT NAME=	New product to be registered.
33	R--55 [ENTER]	PRICE=	
34	4000 [ENTER]	QUANTITY=	
35	2 [ENTER]	PRODUCT NAME=	
36	[ENTER]	TAPE OUT OK (Y/N)	Set cassette tape to save.
37	Y [ENTER]	END (Y/N)	
38	Y [ENTER]	>	
39	[DEF] [B]	SUPPLIER=	Repeat
40	A–123 [ENTER]	TAPE IN OK (Y/N)	Set cassette tape to load.
41	Y [ENTER]	⋮	
	⋮	SUPPLIER =	
44	[ENTER]	>	End

[Program List]

```
 5:"A":CLEAR :
   WAIT 0
10:DIM B$(139),D(
   139),B(139)
11:LF 2
12:USING :LPRINT
   "*PURCHASE LED
   GER*"
13:INPUT "END (Y/
   N) ";W$
14:IF W$="Y"GOTO
   390
15:IF W$<>"N"GOTO
   13
16:INPUT "NEW (Y/
   N) ";Y$
17:IF (Y$="Y")+(Y
   $="N")<>1GOTO
   16
20:CLS :INPUT "SU
   PPLIER=";A$:
   GOTO 27
25:GOTO 245
27:IF Y$="Y"GOTO
   80
30:INPUT "TAPE IN
    OK (Y/N) ";X$
40:IF X$<>"Y"GOTO
   30
50:INPUT #A$;B$(*
   ),D(*),B(*)
80:LF 1
140:USING :LPRINT
   "* ";A$;" *"
143:FOR I=0TO 139
145:IF Y$="Y"GOTO
   150
```

(To be continued)

PROGRAM TITLE	PURCHASE LEDGER GENERATION	PROGRAM NO. P5– D–26	5

[Program List]

```
146:IF B$(I)<>""
    LET Z=1:PAUSE
    B$(I):INPUT "Q
    UANTITY=";S:E=
    S*D(I):GOTO 19
    5
147:IF B$(I)=""
    GOTO 150
148:GOTO 240
150:INPUT "PRODUCT
     NAME=";B$(I):
    Z=0:GOTO 165
160:GOTO 245
165:T=0:INPUT "PRI
    CE=";T
170:S=0:INPUT "QUA
    NTITY=";S
190:E=S*T
195:LPRINT B$(I)
197:IF Z=1LPRINT "
    @";USING "####
    ##########,###
    ";D(I):GOTO 21
    0
200:LPRINT "@";
    USING "#######
    #######,###";T
210:LPRINT "*";S
215:LPRINT "=";E
216:LF 1
220:F=F+E
225:IF Z=1LET D(I)
    =D(I):B(I)=B(I
    )+S:GOTO 240
227:D(I)=T:B(I)=S
240:NEXT I
245:INPUT "TAPE OU
    T OK (Y/N) ";X
    $
246:IF X$<>"Y"GOTO
    245
250:PRINT #A$;B$(*
    ),D(*),B(*)
260:FOR I=0TO 139
265:B$(I)="":D(I)=
    0:B(I)=0
270:NEXT I
300:GOSUB 900
320:G=G+F
325:F=0
330:GOTO 13
390:GOSUB 950
400:END
500:"B":CLEAR
510:DIM B$(139),D(
    139),B(139)
515:LF 2
520:LPRINT "** LIS
    T **"
530:INPUT "SUPPLIE
    R=";A$:GOTO 54
    0
535:GOTO 720
540:INPUT "TAPE IN
     OK (Y/N) ";X$
545:IF X$<>"Y"GOTO
    540
550:INPUT #A$;B$(*
    ),D(*),B(*)
610:LPRINT "* ";A$
    ;" *"
630:FOR I=0TO 139
632:IF B$(I)=""
    GOTO 660
635:E=B(I)*D(I)
640:LPRINT B$(I)
645:LPRINT "@";
    USING "#######
    #######,###";D
    (I)
650:LPRINT "*";B(I
    )
652:LPRINT "=";E
655:F=F+E
660:NEXT I
680:GOSUB 900
685:G=G+F:F=0
700:GOTO 530
720:GOSUB 950
750:END
900:LPRINT "TOTAL"
910:LPRINT USING "
    ##############
    ,###";F
915:LF 1
920:RETURN
950:LPRINT "* ";"G
    RAND TOTAL";"
    *"
960:LPRINT USING "
    ############,
    ###";G
970:RETURN

STATUS 1
                1,399
```

[Memory Contents]

A	
B	
C	
D	
E	Total (for this time)
F	Total by supplier
G	Grand total
H	
I	√
J	
K	
L	
M	
N	√
O	
P	
Q	
R	
S	Quantity (for this time)
T	Price (for this time)
U	
V	
W	
X	
Y	
Z	√
A$	Supplier
W$	√
X$	√
Y$	√
B$(N–1)	Product name
B(N–1)	Quantity
D(N–1)	Price

SHARP

PROGRAM TITLE	BILLING LEDGER AND LIST	PROGRAM NO. P5–D–27	1

CE–150, CE–151 and CTR required

[Outline]

Product numbers, prices, and quantities for each customer are to be entered every time you bill. This billing ledger generation also gives you the total of each product for each customer; and with this clear picture, you can manage your billing control more efficiently, 16 digits are provided for each product number, up to 6 digits for each quantity, price, amount and up to 10 digits for total amount.

[Operating Guide]

[DEF] [A] : These keys are used to make each billing ledger.
A list of the input data is printed out.

[DEF] [B] : Press these keys to generate a list of all products recorded in the tape for each customer.

Note : Make sure that each tape has only one customer.

[Example]

1. Billing ledger (new customer) :

Customer code	"G–55"	
Product number	"K–33"	"H–66"
Price	2,500	1,000
Quantity	6	5

Type in the above data according to the Key Operation Procedure shown later.
When "END ? (Y/N)" displayed, type in "Y".
Use the [DEF] [A] keys again to enter the data of another customer.
Again, make sure to replace the tape with a new one for the new customer.

(New customer) :

Customer code	"Z–99"
Product number	"K–33"
Price	2,500
Quantity	4

Key-in the above in that order.
With the display of "END ? (Y/N)", enter "N" and replace the tape with the costomer code "G–55". Then type in the following to complete the key operation.

Customer code	"G–55"	
Product number	"H–66"	"J–77"
Price	–	3,500
Quantity	6	2

A list generation for each transaction in sequence are as shown in the "Printout" column.

2. When TAPE IN/OUT OK? (Y/N); is displayed, make sure the customer code is the same as that of the tape.
3. For tape saving/loading, make sure to set the tape to the head of the fill.

PROGRAM TITLE	BILLING LEDGER AND LIST	PROGRAM NO. P5–D–27	2

[Contents] (Formulas)

- The billing ledger shows only the inputs given this time.
- A billing list shows the product numbers, prices and quantities now saved in the tape. The quantities for the same product are summed up in the list.
- Up to 140 products per customer can be handled.

[Printout]

```
* BILLING LEDGER *

* G-55 *
K-33
@          2,500
*              6
=         15,000

H-66
@          1,000
*              5
=          5,000

TTL
              20,000

*  GTTL  *
              20,000


* BILLING LEDGER *

* Z-99 *
K-33
@          2,500
*              4
=         10,000

TTL
              10,000
```

```
* G-55 *
H-66
@          1,000
*              6
=          6,000

J-77
@          3,500
*              2
=          7,000

TTL
              13,000

*  GTTL  *
              23,000


** BILLING LIST **
* G-55 *
K-33
@          2,500
*              6
=         15,000
H-66
@          1,000
*             11
=         11,000
J-77
@          3,500
*              2
=          7,000
TTL
              33,000
```

```
* Z-99 *
K-33
@          2,500
*              4
=         10,000
TTL
              10,000

*  GTTL  *
              43,000
```

PROGRAM TITLE	BILLING LEDGER AND LIST	PROGRAM NO. P5–D–27	3

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	END ? (Y/N) _	
2	N ENTER	NEW ? (Y/N) _	
3	Y ENTER	CUSTOMER CODE=_	No tape for this customer is available yet.
4	G–55 ENTER	PRODUCT CODE=_	
5	K–33 ENTER	PRICE=_	Repeat
6	2500 ENTER	QTTY=_	
7	6 ENTER	PRODUCT CODE=_	
⋮	⋮	⋮	
10	5 ENTER	PRODUCT CODE=_	Input is completed for this customer.
11	ENTER	TAPE OUT OK (Y/N)_	Set cassette tape for saving.
12	Y ENTER	END ? (Y/N)_	All inputs are completed.
13	Y ENTER	>	"Total by customer is printed."
14	DEF A	END ? (Y/N)_	
15	N ENTER	NEW ? (Y/N)_	
16	Y ENTER	CUSTOMER CODE=_	
17	Z–99 ENTER	PRODUCT CODE=_	
18	K–33 ENTER	PRICE=_	
19	2500 ENTER	QTTY=_	
20	4 ENTER	PRODUCT CODE=_	Input is completed for this customer.
21	ENTER	TAPE OUT OK (Y/N)_	Set cassette tape for saving.
22	Y ENTER	END ? (Y/N)_	
23	N ENTER	NEW ? (Y/N)_	Tape for this customer is available for inputs.
24	N ENTER	CUSTOMER CODE=_	
25	G–55 ENTER	TAPE IN OK ? (Y/N)=_	Set cassette tape for loading.
26	Y ENTER	K–33	Product code displayed.
		QTTY=_	

PROGRAM TITLE	BILLING LEDGER AND LIST	PROGRAM NO. P5–D–27	4

[Key Operation Procedure]

Step No.	Input	Display	Remarks
27	ENTER	H–66	No input this time.
		QTTY=–	
28	6 ENTER	PRODUCT CODE=–	New product to be registered.
29	J–77 ENTER	PRICE=–	
30	3500 ENTER	QTTY=–	
31	2 ENTER	PRODUCT CODE=–	
32	ENTER	TAPE OUT OK (Y/N)=–	Set cassette tape for saving.
33	Y ENTER	END ? (Y/N)_	
34	Y ENTER	>	
35	DEF B	CUSTOMER CODE=–	Repeat
36	G–55 ENTER	TAPE IN OK ? (Y/N)=–	Set cassette tape for loading data.
37	Y ENTER	G–55	
⋮	⋮	⋮	
40		CUSTOMER CODE=–	
41	ENTER	>	End

[Program List]

```
 5:"A":CLEAR :
   WAIT 0
10:DIM B$(139),D(
   139),B(139)
11:LF 2
12:USING :LPRINT
   "* BILLING LED
   GER *"
13:INPUT "END ?(Y
   /N) ";W$
14:IF W$="Y"GOTO
   390
15:INPUT "NEW ?(Y
   /N) ";Y$
20:INPUT "CUSTOME
   R CODE=";A$:
   GOTO 27
25:GOTO 245
27:IF Y$="Y"GOTO
   80
30:INPUT "TAPE IN
    OK ?(Y/N) ";X
   $
40:IF X$<>"Y"GOTO
   30
50:INPUT #A$;B$(*
   ),D(*),B(*)
80:LF 1
140:USING :LPRINT
   "* ";A$;" *"
143:FOR I=0TO 139
145:IF Y$="Y"GOTO
   150
146:IF B$(I)<>""
   LET Z=1:PAUSE
   B$(I):INPUT "Q
   TTY=";S:E=S*D(
   I):GOTO 195
147:IF B$(I)=""
   GOTO 150
148:GOTO 240
150:INPUT "PROD. C
   ODE=";B$(I):Z=
   0:GOTO 165
160:GOTO 245
165:INPUT "PRICE="
   ;T
170:INPUT "QTTY=";
   S
190:E=S*T
195:LPRINT B$(I)
197:IF Z=1LPRINT "
   @";USING "####
   ##########,###
   ";D(I):GOTO 21
   0
```

(To be continued)

PROGRAM TITLE	BILLING LEDGER AND LIST	PROGRAM NO. P5–D–27	5

[Program List]

```
200:LPRINT "@";
    USING "#######
    #######,###";T
210:LPRINT "*";S
215:LPRINT "=";E
216:LF 1
220:F=F+E
225:IF Z=1LET D(I)
    =D(I):B(I)=B(I
    )+S:GOTO 240
227:D(I)=T:B(I)=S
240:NEXT I
245:INPUT "TAPE OU
    T OK ?(Y/N) ";
    X$
246:IF X$<>"Y"GOTO
    245
250:PRINT #A$;B$(*
    ),D(*),B(*)
260:FOR I=0TO N-1
265:B$(I)="":D(I)=
    0:B(I)=0
270:NEXT I
300:GOSUB 900
320:G=G+F
325:F=0
330:GOTO 13
390:GOSUB 950
400:END
500:"B":CLEAR
510:DIM B$(139),D(
    139),B(139)
515:LF 2
520:LPRINT "** BIL
    LING LIST **"
530:INPUT "CUSTOME
    R CODE=";A$:
    GOTO 540
535:GOTO 720
540:INPUT "TAPE IN
     OK ?(Y/N) ";X
    $
545:IF X$<>"Y"GOTO
    540
550:INPUT #A$;B$(*
    ),D(*),B(*)
610:LPRINT "* ";A$
    ;" *"
630:FOR I=0TO 139
632:IF B$(I)=""
    GOTO 660
635:E=B(I)*D(I)
640:LPRINT B$(I)
645:LPRINT "@";
    USING "#######
    #######,###";D
    (I)
650:LPRINT "*";B(I
    )
652:LPRINT "=";E
655:F=F+E
660:NEXT I
680:GOSUB 900
685:G=G+F:F=0
700:GOTO 530
720:GOSUB 950
750:END
900:LPRINT "TTL"
910:LPRINT USING "
    ##############
    ,###";F
915:LF 1
920:RETURN
950:LPRINT "*  GTT
    L  *"
960:LPRINT G
970:RETURN


STATUS 1
                1,320
```

[Memory Contents]

A	
B	
C	
D	
E	Total (for this time)
F	Total by customer
G	Grand total
H	
I	√
J	
K	
L	
M	
N	√
O	
P	
Q	
R	
S	Quantity (for this time)
T	Price (for this time)
U	
V	
W	
X	
Y	
Z	√
A$	Customer code
W$	√
X$	√
Y$	√
B$(139)	Product number
B(139)	Quantity
D(139)	Price

SHARP

PROGRAM TITLE	BIORHYTHM	PROGRAM NO. P5–E–1	1

CE–150 required

[Outline]

Your mental and physical conditions are a barometer of health, which greatly affect your day. Yes, biorhythm—you can get your monthly biorhythm in advance. Just type in your name and birthday for a printout of your biorhythm graph for any desired month. The curves for the physical (green), emotional (red) and intellectual (blue) provide you with a good indication of your total condition.

[Operating Guide]

- Type in any desired month, your name (up to 16 characters) and your date of birth.
- A biorhythm for your desired month is printed out in different colors for individual factors.

[Example]

Type in the followings:
Desired month: 1981, July
Name: SHARP
Date of birth: 1952, 1 (January), 28th

[Contents] (Formulas)

Input: Desired month, Name, and Birthday

Output: Printout of the biorhythm curves for the desired month (1st to 31st) in different colors for individual factors.

Calculation is made for the X-axis values of the curves as follows:

Physical $X = \mathrm{Sin}\ ((B+Y)/23 \times 360) \times 80$
Emotional $X = \mathrm{Sin}\ ((C+Y)/28 \times 360) \times 80$
Intellectual $X = \mathrm{Sin}\ ((D+Y)/33 \times 360) \times 80$

Where B, C, and D represent the remainders after the total number of days from the birthday to desired time has been divided by the individual cycles.

Y is the number of days (0 to 31).
The maximum length is 16mm in the positive (+) and negative (–) directions.

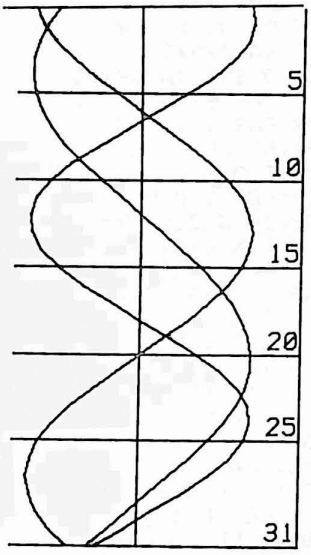
Cycle: Physical: 23 days
Emotional: 28 days
Intellectual: 33 days

PROGRAM TITLE	BIORHYTHM	PROGRAM NO. P5– E–1	2

[Printout]

The actual printout is colored. Refer to page 4.

```
DATE 1981,  7
NAME SHARP
BIRTH 1952,  1, 28

 -- PHYSICAL
 -- EMOTIONAL
 -- INTELLECTUAL

  (-)       (+)
```



[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	DATE? YEAR=–	
2	1981 ENTER	MONTH =–	4 digit input
3	7 ENTER	NAME?_	
4	SHARP ENTER	BIRTH? YEAR =–	
5	1952 ENTER	MONTH =–	4 digit input
6	1 ENTER 28 ENTER	DAY =– >	Printout

PROGRAM TITLE	BIORHYTHM	PROGRAM NO. P5– E–1	3

[Program List]

```
10:"A":CLEAR :
   INPUT "DATE?
   YEAR=";L,"MONT
   H=";M
15:TEXT :COLOR 0
20:LPRINT "DATE";
   USING "######";
   L;",";USING "#
   ##";M
30:N=0
35:GOSUB 700
40:GOSUB 600:O=A
50:INPUT "NAME? "
   ;A$
60:LPRINT "NAME "
   ;A$
70:INPUT "BIRTH?
   YEAR=";L,"MONT
   H=";M,"DAY=";N
80:LPRINT "BIRTH"
   ;USING "######"
   ;L;",";USING "
   ###";M;",";
   USING "###";N
90:GOSUB 600:P=A
100:A=O-P:O=0:P=0
110:LF 2
120:COLOR 2
130:LPRINT " -- PH
   YSICAL"
140:COLOR 3
150:LPRINT " -- EM
   OTIONAL"
160:COLOR 1
170:LPRINT " -- IN
   TELLECTUAL"
180:LF 1
190:COLOR 0
200:LPRINT "  (-)
        (+)"
205:GRAPH
210:GLCURSOR (100,
   0):SORGN
215:Y=I*2.5*5*(-1)
220:LINE (-100,0)-
   (115,0)
230:LINE (0,0)-(0,
   Y)
235:LINE (115,Y)-(
   115,0)
240:FOR Q=5TO 30
   STEP 5
243:R=Q
245:IF Q=30LET R=I
250:Y=R*2.5*(-1)*5
260:LINE (-90,Y)-(
   115,Y)
270:X=80
290:Z=Y+5
300:LINE (115,Z)-(
   X,Z),9
310:LPRINT R
320:NEXT Q
330:B=INT (A/23):B
   =A-(23*B)
340:C=INT (A/28):C
   =A-(28*C)
350:D=INT (A/33):D
   =A-(33*D)
360:FOR J=1TO 3
395:COLOR J
400:E=0
410:FOR Y=0TO I
420:IF J=2LET X=
   SIN ((B+Y)/23*
   360)*80
430:IF J=3LET X=
   SIN ((C+Y)/28*
   360)*80
440:IF J=1LET X=
   SIN ((D+Y)/33*
   360)*80
450:Z=Y*(-1)*2.5*5
460:F=0
470:IF E=0LET F=9:
   LET E=1
480:LINE (O,P)-(X,
   Z),F
490:O=X:P=Z
500:NEXT Y
510:NEXT J
515:TEXT :LF 5:
   COLOR 0
520:END
600:IF M-3>=0LET M
   =M+1:GOTO 620
610:L=L-1:M=13+M
620:A=INT (365.25*
   L)+INT (30.6*M
   )+N
625:A=A-INT (L/100
   )+INT (L/400)
630:RETURN
640:END
700:IF M=2GOTO 790
710:IF M=4GOTO 770
720:IF M=6GOTO 770
730:IF M=9GOTO 770
740:IF M=11GOTO 77
   0
750:I=31:GOTO 900
770:I=30:GOTO 900
790:K=INT (L/4):K=
   L-K*4
800:IF K=0GOTO 840
820:I=28:GOTO 900
840:K=INT (L/100):
   K=L-K*100
845:IF K=0GOTO 850
847:GOTO 890
850:K=INT (L/400):
   K=L-K*400
860:IF K=0GOTO 890
870:GOTO 820
890:I=29
900:RETURN
910:END

STATUS 1
                1327
```

PROGRAM TITLE	BIORHYTHM	PROGRAM NO. P5–E–1	4

[Memory Contents]

A	The total number of days from birthday to the desired month.
B	Set the remainders after division of the total number of days by the cycles. (Physical)
C	Set the remainders after division of the total number of days by the cycles. (Emotional)
D	Set the remainders after division of the total number of days by the cycles. (Intellectual)
E	First judgment
F	Pen-up/down code
G	
H	
I	Corresponding month and the number of days.
J	Loop counter
K	Corresponding year calculation
L	Birthyear/Research year
M	Birthmonth/Research month
N	Birthday
O	The number of days from the year to research time
P	The number of days from the year to birthday
Q	Loop counter
R	Index No. of days
S	
T	
U	
V	
W	
X	Biorhythm curve X-axis
Y	No. of days-per-month counter
Z	Biorhythm curve Y-axis
A$	Name

SHARP

PROGRAM TITLE	BOAT RACE	PROGRAM NO. P5–E–2	1
		CE–150 required	

[Outline]

This is a boat race game in which game players bet points in the double forecasting system on the arrival order of the boats.

[Operating Guide]

The display section is used as a boat race course where 7 boats, represented by the tips of the dots, compete. The game is played by the n number of people who bet their points in the double forecasting system. One player can bet his points on up to 5 combinations of boats, and 1 to 9 bet points on each combination. The bet points are pooled if nobody wins the game, and the pooled points are allocated to the winner in the succeeding game.

[Example]

1. The boat race game is played by two people:

One named JAMES:	Boat combination of 1–2	5 points
The other named FRANK:	Boat combination of 3–5	7 points

[Contents] (Formulas)

1. Boats move by means of random numbers from 1 to 7.
2. Score calculation formulas:

$$\text{Competition Rate} = \frac{(\text{Total bet points} + \text{Carryover points})}{\text{Winning points}}$$

$$\text{Score} = \text{Competition rate} \times \text{No. of winning points} - \text{Winners' bet}$$

[Printout]

```
JAMES
SCORE  =        -5
FRANK
SCORE  =        -7
```

*Both lost.

PROGRAM TITLE	BOAT RACE	PROGRAM NO. P5–E–2	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	NO. OF PLAYERS = _	
2	2 ENTER	NAME? _	Input the number of players.
3	JAMES ENTER	NAME? _	Input the name.
5	FRANK ENTER	>	
6	DEF B	JAMES _	Input the data by player.
		DO YOU BET? (Y, N) _	Do you bet any points?
7	Y ENTER	COMBINATION = _	
8	12 ENTER	BET POINTS (1–9) _	Inputs the combination of 1–2.
9	5 ENTER	COMBINATION = _	If no more bets, press only ENTER.
10	ENTER	FRANK _	Next player
		DO YOU BET? (Y, N) _	
11	Y ENTER	COMBINATION = _	
12	35 ENTER	BET POINTS (1–9) _	
13	7 ENTER	COMBINATION = _	
14	ENTER	>	Starts a game.

PROGRAM TITLE	BOAT RACE	PROGRAM NO. P5–E–2	3

[Program List]

```
 10:"A":CLEAR :DIM
    Z$(1)*16
 15:U=0:Z$(1)="NO
    WINNERS"
 20:INPUT "NO. OF
    PLAYERS=";N
 30:DIM B$(N-1),X1
    (N-1,4),C1(6)
 40:FOR A=1TO N
 50:INPUT "NAME?";
    B$(A-1):GOTO 7
    0
 60:N=A-1:END
 70:NEXT A
 75:END
 90:"B":FOR A=1TO
    N
100:FOR B=1TO 5
110:X1(A-1,B-1)=0
120:NEXT B
130:NEXT A
140:FOR A=1TO N
150:PAUSE USING "&
    &&&&&&";B$(A-1
    )
160:INPUT "DO YOU
    BET?(Y,N)";A$:
    GOTO 180
170:GOTO 280
180:IF A$="N"GOTO
    260
190:FOR B=1TO 5
200:INPUT "COMBINA
    TION=";D:GOTO
    220
210:GOTO 260
220:INPUT "BET POI
    NT(1-9)";E
230:X1(A-1,B-1)=D+
    E/10
240:NEXT B
260:NEXT A
280:WAIT 0:CLS
285:FOR L=1TO 7
290:C1(L-1)=0
300:NEXT L
320:GCURSOR 80:
    GPRINT &FF;&03
    ;&01
340:X=999
350:L=RND 7-1:IF L
    =XGOTO 350
353:E=2^L:P=RND 2+
    1
355:GCURSOR (C1(L)
    +1)
360:FOR I=C1(L)+1
    TO C1(L)+P:Q1=
    POINT IOR E:
    GPRINT Q1;:
    NEXT I
365:C1(L)=C1(L)+P
390:IF C1(L)<80
    GOTO 350
400:BEEP 1,90,50:
    BEEP 1,70,50
402:BEEP 1,150,90:
    BEEP 1,150,100
404:BEEP 1,50,60:
    BEEP 1,250,150
410:IF X=999LET X=
    L:GOTO 350
420:Y=L+1:X=X+1
490:WAIT 100:USING
    :CURSOR 15:
    PRINT X;"-";
    STR$ Y
500:S=10*X+Y:J=10*
    Y+X
510:Z=0:G1=0:D1=0
520:FOR A=1TO N
530:D=0:G=0
540:FOR B=1TO 5
550:L=INT (X1(A-1,
    B-1))
560:Q=(X1(A-1,B-1)
    *10-L*10)
570:IF L=SGOTO 610
580:IF L=JGOTO 610
590:G=G+Q:G1=G1+Q
600:GOTO 620
610:D1=D1+Q:D=D+Q
620:NEXT B
630:X1(A-1,1)=G:X1
    (A-1,2)=D
640:Z=Z+G+D
650:NEXT A
660:IF D1=0GOTO 69
    0
670:K=(U+Z)/D1:U=0
675:LPRINT "ALLCTN
     RATE";USING "
    ####";K
680:GOTO 710
690:PAUSE USING "&
    &&&&&&&&&&&&&&
    ";Z$(1)
700:K=0:U=U+Z
710:FOR F=1TO N
720:A=F-1
730:D=X1(A,2)*K-X1
    (A,1)-X1(A,2)
740:LPRINT USING "
    &&&&&&&";B$(A)
750:LPRINT "SCORE"
    ;"=";USING "##
    ##";D
780:NEXT F
790:LF 2:END


STATUS 1
                1257
```

PROGRAM TITLE	BOAT RACE	PROGRAM NO. P5–E–2	4

[Memory Contents]

A	√	A$		B$(N–1)	Name Table
B	√	B$			
C		C$		X1(N–1,4)	Combinations and Bet Points Table by player
D	Individual Winning Points	D$			
E	√	E$		C1(6)	Boat Position
F	√	F$			
G	Individual Losing Points	G$		G1	Total Losing Points
H		H$			
I	√	I$		D1	Total Winning Points
J	2nd–1st Combination	J$			
K	Competition Rate	K$		Z$(1)	Used in letter string
L	√	L$			
M	√	M$			
N	No. of Players	N$			
O		O$			
P	Boat Speed	P$			
Q	√	Q$			
R		R$			
S	1st-2nd Place Combination	S$			
T		T$			
U	Carryover Point	U$			
V	√	V$			
W		W$			
X	Winning Boat No.	X$			
Y	2nd Place Boat No.	Y$			
Z	Total Bet Points	Z$			

SHARP

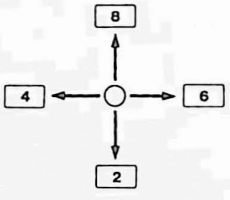
PROGRAM TITLE	LABYRINTH ESCAPE	PROGRAM NO. P5–E–3	1
		CE–151 required	

[Outline]

There's no way out if you keep going ahead!
The labyrinth is first displayed on the computer display, and is then covered with a masking pattern. A street appears if you successfully pass through a passageway. Bumping into the wall causes an alarm to sound. This game competes for the shortest time to reach at the goal. The elapsed time is displayed.

[Operating Guide]

1. With the [DEF] [A] keys pressed, the labyrinth is displayed on the display. It is then covered with the mask.
2. With your present position (Dot) flashing, advance by key operation.
3. Key operation



The flashing dot moves in the Designated direction.

4. Upon reaching the goal, the "cheers" mark and elapsed time are displayed. The instructions for "Replay" are displayed after few seconds.
With the elapsed time on display, the time for the present game and the shortest time up to now are indicated.
Press the [Y] (Yes) to restart a game, and the [N] (No) to end the game.

[Contents] (Formula)

1. Selects three labyrinth patterns (105 dots) by using random numbers (1 to 12) for display.
After a few seconds, the masking pattern (All are & 7F) begins being displayed.
2. Following the passage correctly causes a white-on-black passage to appear. Advancing against a wall results in an alarm that sounds 3 times.
3. The moving dot is flashed to distinguish itself from the labyrinth pattern.
4. Upon arrival of the dot at the goal, the "Cheers" pattern is displayed, then the shortest time up to now and the elapsed time for the present game are indicated.
5. Replay and Program End:
There appears "REPLAY (Y or N)?" after the time displayed, waiting for the next designation. With [Y] pressed, "REPLAY" begins from Step 1. With [N] pressed, the program is completed.

PROGRAM TITLE	LABYRINTH ESCAPE	PROGRAM NO. P5–E–3	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	After once displayed, the labyrinth pattern is masked.	The masking pattern is displayed in columns, one by one, from the left. The moving dot flashes, and the time counter dashs for time display.
	2		The movement designation moves the dot.
	4		
	6		
	8		
	⋮		The white-on-black passage, "Cheers" mark, and the elapsed time are displayed.
		YOUR TIME: ■ : ■	Stays for 2 or 3 seconds.
		SHORTEST TIME: ■ : ■	The shortest time is displayed. Stays for 2 or 3 seconds.
		REPLAY (Y or N)? –	Replay or completion?
OR	Y ENTER		To step 2 for replay.
	N ENTER	>	The game is over.

PROGRAM TITLE	LABYRINTH ESCAPE	PROGRAM NO. P5–E–3	3

[Program List]

```
10:"A":CLEAR :
   RANDOM :M=&FF:
   S=&FF
20:CLS :WAIT 0:
   DIM T$(11)*68,
   PT(102)
30:T$(0)="087F427
   90F7957554C770
   977555D427F550
   47F10775545700
   977525E55655F5
   55B6D"
31:T$(1)="086F217
   F027F514C575D5
   5457D296D533E4
   55E515D5577097
   D55556F507F524
   77A4A"
32:T$(2)="087F494
   B714D5F5115755
   4475D4575455D4
   575457D017F096
   B217F207F25712
   F692B"
33:T$(3)="087B427
   F482F715D47725
   D57552D75594F7
   01F70475D4570D0
   17F417759087F4
   27B49"
34:T$(4)="087F4A0
   97F097A477D415
   7147F207F285F6
   A77445D117F116
   F297E246F157D4
   57B2A"
35:T$(5)="087F047
   F207F017D55575
   05F407F486B2A5
   F52553F517D047
   F415D7506735E4
   27B4A"
40:T$(6)="5B6D557
   D53557D2577485
   F515577047F105
   57F215D5577487
   71955754F784F2
   17F08"
41:T$(7)="292F712
   57F057B55555F4
   877555D453D513
   E655B4A5F51555
   D7519457F207F4
   27B08"
42:T$(8)="6A4B7A4
   7527F027F426B4
   87F405F5157515
   D5157515D711557
   54457D5947694
   97F08"
43:T$(9)="496F217
   F084D77417F405
   F515D71077C077
   94D575A55755D2
   7715D477A097F2
   16F08"
44:T$(10)="2A6F51
   5F547B123F4A7B
   447F445D11772B
   7D0A7F027F1475
   415F712F487F48
   297F08"
45:T$(11)="296F21
   3D67305757D417F
   105F457E55257D
   2A6B097F017D05
   75555F407F027F
   107F08"
50:A=RND 12:B=RND
   12:C=RND 12
60:IF A=BGOTO 50
61:IF A=CGOTO 50
62:IF B=CGOTO 50
70:CLS :A=A-1:B=B
   -1:C=C-1
80:GCURSOR 0:
   GPRINT T$(A);T
   $(B);T$(C);
81:BEEP 10,10,10
100:FOR CP=0TO 101
101:BEEP 1,1,1
110:A=POINT CP:PT(
   CP)=A
120:GCURSOR CP:
   GPRINT "7F"
130:NEXT CP
140:PT(102)=&08
150:D=8:CP=0:Z=0
160:TIME =0
170:WAIT 0:X=POINT
   CP
180:CURSOR 23:
   PRINT Z:Y=DOR
   X
190:GCURSOR CP:
   GPRINT Y:D1=D
200:A$=INKEY$
210:IF A$<>""GOTO
   300
220:A=&7F-D:A=AAND
   X
230:GCURSOR CP:
   GPRINT A
240:D=0
250:A$=INKEY$
260:IF A$<>""GOTO
   300
270:D=D1
280:Z=Z+1:IF Z>99
   LET Z=0:CURSOR
   24:PRINT "0 "
290:GOTO 180
300:BEEP 1,10,10
310:IF A$="8"LET D
   W=INT ((D1+1)/
   2):GOTO 400
320:IF A$="2"LET D
   W=D1*2:GOTO 40
   0
330:IF A$="6"LET P
   W=CP+1:GOTO 50
   0
340:IF A$="4"LET P
   W=CP-1:GOTO 50
   0
350:BEEP 2,10,20
360:D=D1
370:GOTO 180
400:IF DW>64LET DW
   =64
410:A=PT(CP):A=A
   AND DW
420:IF A=0BEEP 3,1
   0,30:DW=D1:
   GOTO 440
430:A=&7F-D1:X=A
   AND X
440:GCURSOR CP:
   GPRINT X
450:D=DW
460:GOTO 170
```

(To be continued)

PROGRAM TITLE	LABYRINTH ESCAPE	PROGRAM NO. P5–E–3	4

[Program List]

```
500:IF PW<0LET PW=
    0:BEEP 3,10,30
    :GOTO 570
510:IF PW>101GOTO
    600
520:A=PT(PW):A=A
    AND D1
530:IF A=0BEEP 3,1
    0,30:GOTO 570
540:A=&7F-D1:X=A
    AND X
550:GCURSOR CP:
    GPRINT X
560:CP=PW
570:D=D1
580:GOTO 170
600:A=PT(PW):A=A
    AND D1
610:IF A=0BEEP 3,1
    0,30:D=D1:GOTO
    170
620:GCURSOR 105:
    GPRINT "04087B
    3F7B0804"
621:BEEP 1,90,50
622:BEEP 1,70,50
623:BEEP 1,150,90
624:BEEP 1,150,100
625:BEEP 1,60,60
626:BEEP 1,200,200
630:T=TIME :T=T-
    INT T:T=T*1000
    0
640:MM=INT (T/100)
    :SS=T-(MM*100)
645:NP$=STR$ MM+":
    "+STR$ SS
650:WAIT 150:
    CURSOR 21:
    PRINT NP$
660:IF M>MMLET M=M
    M:S=SS:GOTO 70
    0
670:IF M<>MMGOTO 7
    00
680:IF S>SSLET S=S
    S
700:CLS :CURSOR 0:
    WAIT 0
710:PRINT "YOUR TI
    ME:"
720:WAIT 150:
    CURSOR 12:
    PRINT NP$
730:CLS :CURSOR 0:
    WAIT 0
740:PRINT "SHORTES
    T TIME:"
745:HP$=STR$ M+":"
    +STR$ S
750:WAIT 150:
    CURSOR 16:
    PRINT HP$
760:CLS :WAIT 0:
    CURSOR 0
770:PRINT "REPLAY(
    YorN)"
780:CURSOR 13:
    INPUT A$
790:IF A$="Y"GOTO
    50
800:IF A$="N"GOTO
    850
810:GOTO 760
850:CLS :CURSOR 0:
    END
```

```
STATUS 1
                    2413
```

PROGRAM TITLE	LABYRINTH ESCAPE	PROGRAM NO. P5–E–3	5

[Memory Contents]

A	√	A$	Key input data	PT(102)	Pattern Table
B	√	B$		CP	Cursor Point
C	√	C$		D1	Your Dot Position
D	Your Dot Position	D$		DW	(Work) Vertical Shift – Dot Position
E		E$		PW	Horizontal Shift – Dot Position
F		F$		MM	Elapsed Time (Minute)
G		G$		SS	Elapsed Time (Second)
H		H$		NP$	Elapsed Time Editing Data
I		I$		HP$	Shortest Time Editing Data
J		J$		T$(11)	Dot Pattern Table
K		K$			
L		L$			
M	Shortest Time (Minute)	M$			
N		N$			
O		O$			
P		P$			
Q		Q$			
R		R$			
S	Shortest Time (Second)	S$			
T	Time Calculation Value (Min. Sec.)	T$			
U		U$			
V		V$			
W		W$			
X	Present Point Pattern	X$			
Y	Present Point + Your Pattern	Y$			
Z	Display Counter (Work)	Z$			

SHARP

PROGRAM TITLE	DOUBLE ROTATION	PROGRAM NO. P5–E–4	1

[Outline]

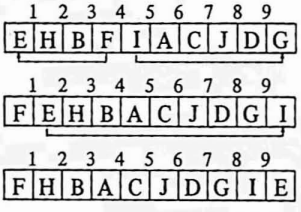
This is a brain game to rearrange alphabetical letters (A to J) put at random.
Can you succeed in the first attempt? Perhaps not. Try it.

[Operating Guide]

1) DEF A "DOUBLE ROTATION" is displayed. Then, the alphabetical letters A, B, C . . . . . J, are on display in irregular order. With the inputs of the breakpoints (1 to 9), the displayed alphabet is rotated.
Your score is determined by the frequency of key operations. The less, the better.

2) DEF B The display becomes the same as that already shown in DEF A . It is a lot of fun to compete with others for the most efficient and quick alphabetical rearrangement.

[Example]

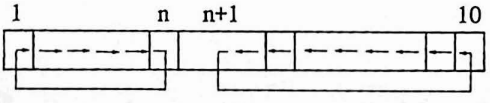


For example, if the breakpoint 4 is input in this letter string,
the alphabetical letters are rotated as shown .
Next, shown on the left when the breakpoint 1 is pressed.

In this manner, try to make efficient rearrangement.

[Contents] (Formulas)

Your score depends on the frequency of key operations.
Therefore, the less, the greater your are.



$(1 \leqq n \leqq 9)$

PROGRAM TITLE	DOUBLE ROTATION	PROGRAM NO. P5– E–4	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	DOUBLE ROTATION A to J	This is displayed until the alphabet is stored. (Random order)
2	1 ~ 9	(Repeat)	Press any one of the break-points 1 to 9.
3	ENTER	A to J GAME END YOUR SCORE	(Rotated alphabets on display) Displayed when the alphabet is rearranged in correct sequence. The score is displayed.
	DEF B	A to J (Random display)	The letter string first displayed in the DEF A appears and the procedure returns to step 2.

PROGRAM TITLE	DOUBLE ROTATION	PROGRAM NO. P5– E–4	3

[Program List]

```
 10:"A":CLEAR :
    WAIT 0:USING
 20:PAUSE "DOUBLE
    ROTATION"
 30:D$="ABCDEFGHIJ
    "
 40:Y$=""
 50:A=0
 60:FOR I=1TO 10
 70:R=RND 10
 80:S=2^(R-1)
 85:B=SAND A
 90:IF B<>0GOTO 70
100:A=AOR S
110:Y$=Y$+MID$ (D$
    ,R,1):NEXT I
120:S$=Y$
130:N=0
140:USING
150:BEEP 1:CLS :
    PRINT USING "&
    &&&&&&&&&";S$;
    " *POINT=  "
160:CURSOR 20
165:C=0
170:INPUT C
190:IF C<1GOTO 140
200:IF C>9GOTO 140
210:K$=LEFT$ (S$,C
    )
220:L$=RIGHT$ (S$,
    10-C)
240:IF C=1GOTO 260
250:K$=RIGHT$ (K$,
    1)+LEFT$ (K$,C
    -1)
260:IF C=9GOTO 280
270:L$=RIGHT$ (L$,
    9-C)+LEFT$ (L$
    ,1)
280:S$=K$+L$
290:N=N+1
300:IF S$<>D$GOTO
    140
310:BEEP 5:CLS :
    USING :PAUSE "
    GAME END"
320:WAIT :USING :
    PRINT USING "#
    ###";"YOUR SCO
    RE";N
330:END
400:"B":CLS :WAIT
    0:GOTO 120
```

STATUS 1

523

[Memory Contents]

A	√
B	√
C	Input Key
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	Score
O	
P	
Q	
R	Random number
S	√
T	
U	
V	
W	
X	
Y	
Z	
D$	ABCDEFGHIJ
K$	Randomly ordered
L$	alphabet after key operation
S$	Randomly orderd alphabet
Y$	Randomly orderd alphabet (for Saving)

SHARP

PROGRAM TITLE	MOLE BANGING	PROGRAM NO. P5–E–7	1

[Outline]

Strike a fleeing mole on the head!
With this game, key operation timing is essential to bang the mole when it comes out of its tunnel.
The mole raises its head in three stages. If you can strike its head in the first or second stage, you can get a score. When you miss the mole of coming to the final stage four times, the game is over.

[Operating Guide]

1. With the [DEF] and [B] pressed, the mole appears.
   Press the corresponding software key.
2. You can get 2 points if you bang the mole in the first stage, 1 point in the second, and no points in the third.
   You lose 2 points if you strike where there is no mole.
3. As the game continues, the mole moves slightly faster.
4. When you fail to strike the mole four times, the game is over.

Software Keys; [!] ["] [#] [$] [%] [&]

[Contents]

Finding the mole display positions (1 to 6) randomly, raise the display stages (1 to 3) in constant cycles (with sound).
The score is added when the key at the corresponding position is pressed. But the score is subtracted when any key other than the corresponding positions is pressed.
A successfully banged mole is displayed upside down, and shrieks.
If you miss the mole four times, the game is over.

PROGRAM TITLE	MOLE BANGING	PROGRAM NO. P5–E–7	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF B	GAME START	
	!	Mole display in 3 stages	Press the nearest soft ware key immediately after a look.
	"	⋮	
	#	⋮	
	$	⋮	
	%	⋮	
	&	⋮	
		Score Display	
		GAME OVER SCORE : Score Display	The game is over.

PROGRAM TITLE	MOLE BANGING	PROGRAM NO. P5–E–7	3

[Program List]

```
 10:"B":CLEAR
 20:WAIT 150:PRINT
    "GAME START"
 30:WAIT 0:CLS :
    RANDOM
 40:PRINT T
 50:W=4:E=24:F=40
 60:P=RND 6
 70:FOR R=1TO 3
 80:A$="":A=&FF:
    GOSUB 200:
    GOSUB 300
 90:IF A=&FFGOTO 1
    20
100:IF P=AGOTO 150
110:GOSUB 500
120:IF R=3LET X=X+
    1
130:NEXT R
140:GOTO 160
150:GOSUB 400
160:WAIT 0:GCURSOR
    C:GPRINT "0000
    000000000000"
170:E=E-1:F=F-1
172:IF X>3GOTO 600
174:IF E=0LET W=1:
    GOTO 60
176:IF F=0GOTO 600
180:GOTO 60
200:C=10+((P-1)*24
    ):GCURSOR C
210:BEEP 1,10,10
220:IF R=1WAIT W:
    GPRINT "204060
    7070604020":
    GOTO 250
230:IF R=2WAIT W:
    GPRINT "081078
    7C7C781008":
    GOTO 250
240:IF R=3WAIT W:
    GPRINT "02647E
    3F3F7E6402"
250:RETURN
300:A$=INKEY$
340:IF A$=CHR$ &11
    LET A=1:GOTO 3
    95
350:IF A$=CHR$ &12
    LET A=2:GOTO 3
    95
360:IF A$=CHR$ &13
    LET A=3:GOTO 3
    95
370:IF A$=CHR$ &14
    LET A=4:GOTO 3
    95
380:IF A$=CHR$ &15
    LET A=5:GOTO 3
    95
390:IF A$=CHR$ &16
    LET A=6
395:RETURN
400:BEEP 1,30,30:
    GCURSOR C:WAIT
    10
410:IF R=1GPRINT "
    20103070703010
    20":T=T+2:GOTO
    440
420:IF R=2GPRINT "
    20103C7C7C3C10
    20":T=T+1:GOTO
    440
430:IF R=3GPRINT "
    20133F7E7E3F13
    20"
440:PRINT T
450:RETURN
500:IF R=1LET T=T-
    2:GOTO 530
510:IF R=2LET T=T-
    1:GOTO 530
530:PRINT T
540:RETURN
600:WAIT 150:PRINT
    "GAME OVER  SC
    ORE: ";T;
610:GCURSOR 0:CLS
    :END


STATUS 1
                865
```

[Memory Contents]

A	√
B	
C	Display Cursor Position
D	
E	1st Loop Counter
F	2nd Loop Counter
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	Mole Display Positions (1 to 6)
Q	
R	Mole Display Stages (1 to 3)
S	
T	Score
U	
V	
W	Waiting time
X	No. of missed moles
Y	
Z	
A$	Area for IN KEY $

SHARP

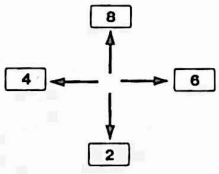
PROGRAM TITLE	SPACE EVADER GAME	PROGRAM NO. P5–E–9	1

[Outline]

Can the spaceship escape from a cluster of meteorites?
This program is a game to drive the spaceship to the goal through a cluster of meteorites on the display. Operation is performed using the UP, DOWN, LEFT and RIGHT keys. The max. score is 100.
The point to increase your sore is how often you can aboid collision.

[Operating Guide]

1. Key Operation



As illustrated, the spaceship moves in the directions corresponding to the keys.
The spaceship keeps flashing.

2. Score
   - 2.1 Vertical key operation has nothing to do with the score.
   - 2.2 Returning the spaceship to the left counts down by one point.
   - 2.3 Advancing the spaceship to the right counts up by one point.
   - 2.4 Hitting the spaceship against a meteorite counts down by 5 points.
3. When the spaceship hits a meteorite, an explosion is displayed, and an alarm sounds.
   The game is, however, restarted.

[Contents]

1. The randomly selected one to two-dotted meteorite pattern per row is stored in the meteorite display pattern table. One to 100 rows are to be housed, with an alarm sounding for each.
2. After the display of the housed meteorite pattern table contents, the spaceship appears in the first row, thus starting the game. The spaceship moves, while flashing.
3. The spaceship goes straight on to the right at a constant speed. Operate the appropriate key to prevent the spaceship from hitting a meterorite. When the spaceship collide with a meteorite, the explosion pattern is displayed. This decreases 5 points from the score.
4. One point decreases from the score when the spaceship returns, and one point increases when it advances.
   Moving in other directions does not affect the score.
5. The checkered flag will be displayed when the spaceship arrives at the goal.

PROGRAM TITLE	SPACE EVADER GAME	PROGRAM NO. P5–E–9	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	GAME START (Meteorite Pattern) (Score)	The alarm sounds during meteorite patten generated.
	2 4 6 8	The spaceship treks flashing.	These keys are to operate the spaceship.
		(Score)	When the spaceship arrives at the goal, a checkered flag appears.
		GAME OVER SCORE: (Score)	The game is over.

[Program List]

```
10:"A":CLEAR
20:DIM T(101):
   RANDOM
30:WAIT 100:PRINT
   "GAME START"
40:FOR N=1TO 100
   STEP 2
50:A=RND 7:B=2^(A
   -1)
60:IF (A=1)+(A=7)
   <>1LET T(N)=B:
   BEEP 1,1,1:
   GOTO 110
70:C=RND 7
80:IF A=CGOTO 70
90:C=2^(C-1)
100:T(N)=BOR C:
   BEEP 1,1,1
110:NEXT N
120:T(0)=&7F:T(1)=
   0:T(101)=&7F
130:WAIT 0:FOR P=0
   TO 101
140:GCURSOR P:
   GPRINT T(P)
150:NEXT P
160:CURSOR 22:
   PRINT S
170:P=0:D=1
180:FOR I=1TO 2
181:IF P>100GOTO 5
   00
190:A$="":X=0:WAIT
   2
200:A=T(P):B=AOR D
210:GCURSOR P:
   GPRINT B
220:A$=INKEY$
230:IF A$<>""GOTO
   300
250:GCURSOR P:
   GPRINT A
260:A$=INKEY$
270:IF A$<>""GOTO
   300
290:NEXT I
295:P=P+1:S=S+1:
   GOTO 350
300:GCURSOR P:
   GPRINT A
305:IF A$="8"LET D
   =INT ((D+1)/2)
   :GOTO 350
```

(To be continued)

PROGRAM TITLE	SPACE EVADER GAME	PROGRAM NO. P5–E–9	3

[Program List]

```
310:IF A$="2"LET D
    =D*2:IF D>64
    LET D=64:GOTO
    350
320:IF A$="6"LET P
    =P+1:S=S+1:
    GOTO 350
330:IF A$="4"LET P
    =P-1:S=S-1:IF
    P<1LET P=1:
    GOTO 350
340:GOTO 290
350:A=DAND T(P)
351:IF P>100GOTO 5
    00
360:IF A=0BEEP 1,3
    0,30:CURSOR 22
    :PRINT S:GOTO
    180
370:A=P-4:IF A<1
    LET A=1
380:BEEP 5,10,10
390:WAIT 70:
    GCURSOR A:
    GPRINT "00082A
    1C7F1C2A0800"
400:WAIT 0:S=S-6:
    CURSOR 22:
    PRINT S
410:FOR E=ATO A+10
415:IF E>101GOTO 4
    40
420:GCURSOR E:
    GPRINT T(E)
430:NEXT E
440:P=P+1:GOTO 180
500:WAIT 150:
    GCURSOR 105:
    GPRINT "7F556B
    556B556B557F"
501:CLS :WAIT 150
502:S=S-1
510:PRINT "GAME OV
    ER  SCORE: ";S
    ;
520:CLS :END


STATUS 1
                927
```

[Memory Contents]

A	√
B	√
C	√
D	Spaceship Dot Position
E	√
F	
G	
H	
I	√
J	
K	
L	
M	
N	Loop Counter for meteorite pattern storage
O	
P	Display Position
Q	
R	
S	Score
T	Meteorite Pattern Table
U	
V	
W	
X	
Y	
Z	
A$	Input Data
T(101)	Dot pattern storage

SHARP

PROGRAM TITLE	TYPING EXERCISES	PROGRAM NO. P5–F–1	1

[Outline]

Quick key operation adds up to substantial savings.
How fast and accurately can you type in on the keyboard?
This program helps you improve your typing speed for better key operation. The result is prompt program input to the machine with increased efficiency.

[Operating Guide]

When the buzzer sounds, a typing exercise in 3 to 6 letters is displayed. Now type in the same letters by using the keyboard within the predetermined time limit. You get 10 points when your typing is perfect, and 5 points when it is more than 50% correct. If typing exceeds the predetermined time limit, another exercise will come out.
The time limit depends on the number of letters displaged and the exercise grades (1, 2, 3). Grade 1 is the shortest, and Grade 3 is the longest. Ten typing exercises in each grade.
Challenge to the perfect score of 100.

[Contents]

The number of letters (3 to 6) is determined by using random-number-generating function.
The letter string (A to Z) is also extracted by using same function.

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF Z	GRADE (1, 2, 3)?	This step is used to initiate the game or to alter the grade.
2	DEF A	HIGH–SCORE =	This operation is not necessary if DEF Z is operated already.
3		(6 letter string)	3 alarms
4	Type in the letters shown on the left of the display.		An exercise changes when all the letters are typed or when the predetermined time elapses.
			Repeated 10 times.
5		YOUR–SCORE =	After 3 alarms, the score is displayed.
6		YOUR SCORE IS BEST	This is displayed only when you got the highest marks.

PROGRAM TITLE	TYPING EXERCISES	PROGRAM NO. P5–F–1	2

[Program List]

```
 10:"Z":CLEAR :CLS
    :DIM A$(5),B$(
    5):RANDOM
 15:INPUT "GRADE(1
    ,2,3)?";L
 17:IF (L=1)+(L=2)
    +(L=3)<>1THEN
    15
 20:"A":WAIT 0:P=0
    :PAUSE "HIGH-S
    CORE=";X
 30:FOR S=1TO 10
 40:B=RND 4+2:Y$="
     ":R=INT (B/2)
 50:FOR C=0TO B-1:
    B$(C)=""
 60:D=RND 26:A$(C)
    =CHR$ (D+&40):
    Y$=Y$+CHR$ (D+
    &40):NEXT C
 70:CLS :BEEP 3:
    PRINT Y$:
    CURSOR 10:E=0
 80:FOR W=1TO B*10
    *L:B$(E)=
    INKEY$ :IF B$(
    E)=""THEN 100
 85:PRINT B$(E);
 90:E=E+1:IF E=B
    LET W=400
100:NEXT W:Q=0
110:FOR W=0TO B-1:
    IF A$(W)=B$(W)
    LET Q=Q+1
120:NEXT W:IF Q<=R
    THEN 150
130:IF Q=BLET P=P+
    10:GOTO 150
140:P=P+5
150:NEXT S:CLS :
    BEEP 3:PAUSE "
    YOUR-SCORE=";P
160:IF P>XLET X=P:
    PRINT "YOUR SC
    ORE IS BEST"
170:WAIT :PRINT :
    END
```

STATUS 1

505

[Memory Contents]

A	
B	No. of typed leeters
C	
D	
E	No. of typed letters
F	
G	
H	
I	
J	
K	
L	Grade
M	
N	
O	
P	Score
Q	No. of correctly typed letters
R	
S	No. of exercises
T	
U	
V	
W	Time
X	Highest score
Y	Letter string of an exercise
Z	
Y$	Letter string of an exercise
A$(5)	Randomly generated letters
B$(5)	Typed letters

SHARP

PROGRAM TITLE	STOP WATCH, TIMER, AND ALARM CLOCK	PROGRAM NO. P5–F–2	1

[Outline]

This program shows you how convenient and versatile the PC–1500 built-in clock is. Three funcions–stop watch, timer, and alarm are included.

[Operating Guide]

DEF S : Press the SPACE key to input "a start and an end" instructions. The elapsed time is continuously displayed.

DEF D : Set the timer time and press the SPACE key to start the timer. When the specified time has elapsed, a melody will let you know that time is up.
The elapsed time is also displayed then.

DEF A : With the input of alarm time (0 to 23 hour, 0 to 59 minute, and 0 to 59 second), the preset time is indicated by a melody.
The time is also displayed.

(Caution) Before using this program, make sure to set the built-in clock (TIME) correct.

[Example]

DEF S : With the SPACE key pressed, the elapsed time is displayed in the form of OM OS 2SS .
Pressing again the SPACE key displays the elapsed time in the form of STOP 10M 59S 4SS to complete processing.

DEF D : Type in "003000" for the time when set to 0 hour, 30 minutes, 0 second. Pressing the SPACE key displays the elapsed time in the form of TIME LAPSE OH OM 1S .
When the preset time has elapsed, a melody sounds.

DEF A : Type in "105700" for the alarm time when set to 10:57:00. Current time is displayed as NOW-TIME 10H 54M 5S .
When the preset time has come, a melody sounds.

[Contents]

DEF S : Stop watch function
This starts and ends with the SPACE key.
The elapsed time is displayed by 1/5 second increments.

DEF D : Timer function
With the time input to the timer (hour, minute and second), the operation starts with the SPACE key.
When time is over, a melody sounds.
The elapsed time is displayed by one second increment.

[DEF] [A] : Alarm clock function
With the input of the alarm time (hour, minute and second), when the preset time has come, a melody sounds to announce and display the time.

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	TIME=10.5350 [ENTER]	10.5350	Type in the current time. (10:53:50)
2	[DEF] [A]	ALARM–TIME?	Designate the alarm time. (10:57:00)
3	105700 [ENTER]	NOW–TIME 10H53M56S	Current time being displayed.
		⋮	
4		NOW–TIME 10H57M00S	Alarm sounds 20 times
1	[DEF] [D]	TIMER?	Designate the time. Set the time after 00 : 30 : 00.
2	003000 [ENTER]	TIMER 003000	
3	[SPACE]	TIME LAPSE 0H 0M 1S	
		⋮	
4		TIME LAPSE 0H 30M 0S	Alarm sounds 20 times.
1	[DEF] [S]	STOP WATCH	
2	[SPACE]	0M 0S 0SS	Stop watch starts.
		⋮	
3	[SPACE]	STOP 1M 0S 0SS	One minute elapsed.

PROGRAM TITLE	STOP WATCH, TIMER, AND ALARM CLOCK	PROGRAM NO. P5–F–2	3

[Program List]

```
 10:"A":CLEAR :
    WAIT 0
 20:INPUT "ALARM-T
    IME?";T
 30:O=TIME
 40:K=INT (O/100)
 50:O=(O-K*100)*10
    000
 60:N=TIME
 70:K=INT (N/100)
 80:N=(N-K*100)*10
    000
 90:IF O=NGOTO 200
 95:GOSUB 900
100:PRINT "NOW-TIM
    E";USING "###"
    ;H;"H";USING "
    ###";M;"M";
    USING "###";S;
    "S"
120:IF T=NBEEP 20:
    GOTO 300
200:O=N
210:GOTO 60
300:END
400:"D":CLEAR :
    WAIT 0
410:INPUT "TIMER?"
    ;N
415:GOSUB 900
420:U=(H*60^2)+(M*
    60)+S
440:S=0
445:A$=""
450:A$=INKEY$
460:IF A$<>" "GOTO
    445
470:O=TIME :K=INT
    (O/100):O=(O-K
    *100)*10000
480:N=TIME :K=INT
    (N/100):N=(N-K
    *100)*10000
490:IF O=NGOTO 480
500:S=S+1
505:U=U-1
510:Z=S
520:IF Z<60GOTO 55
    5
530:Y=INT (Z/60):Z
    =Z-Y*60
540:IF Y<60GOTO 55
    5
550:X=INT (Y/60):Y
    =Y-X*60
555:PRINT "TIME LA
    PSE";USING "##
    ";X;"H";USING
    "###";Y;"M";
    USING "###";Z;
    "S"
560:IF U=0GOTO 590
570:O=N:GOTO 480
590:BEEP 20
600:END
650:"S":CLEAR :
    WAIT 0
655:PRINT "STOP WA
    TCH"
660:H=0:M=0:S=0:U=
    0
670:A$=""
680:A$=INKEY$
690:IF A$<>" "GOTO
    670
696:V=TIME
730:U=U+2
735:A=0:A=0
740:IF U<10LET S=S
    +0:A=0
750:IF U=10LET S=S
    +1:U=0
760:IF S<60LET M=M
    +0:A=0
770:IF S=60LET M=M
    +1:S=0
810:PRINT M;"M";
    USING "###";S;
    "S";USING "##"
    ;U;"SS"
815:A$=""
820:A$=INKEY$
830:IF A$<>" "GOTO
    730
840:WAIT :USING :
    PRINT "STOP";M
    ;"M";S;"S";U;"
    SS"
850:END
900:H=INT (N/10000
    )
910:M=INT ((N-H*10
    000)/100)
920:S=INT (N/100):
    S=N-S*100
930:RETURN
940:END

STATUS 1
                1037
```

[Memory Contents]

: Timer Function

A	
B	
C	
D	
E	
F	
G	
H	Timer Time (Hour)
I	
J	
K	Calculation
L	
M	Timer Time (Minute)
N	Timer Time: Elapsed Time (Now)
O	Elapsed Time (Old)
P	
Q	
R	
S	Timer Time (Second)
T	
U	Timer time conversion to seconds
V	
W	
X	Elapsed Time (Hour)
Y	Elapsed Time (Minute)
Z	Elapsed Time (Second)
A$	INKEY$

[Memory Contents]

: Alarm Clock Function

A	
B	
C	
D	
E	
F	
G	
H	Current Time (Hour)
I	
J	
K	Calculation
L	
M	Current Time (Minute)
N	Elapsed Time (Now)
O	Elapsed Time (Old)
P	
Q	
R	
S	Current Time (Second)
T	Alarm Time
U	
V	
W	
X	
Y	
Z	

: Stop Watch Function

A	WORK
B	
C	
D	
E	
F	
G	
H	Elapsed Time (Hour)
I	
J	
K	
L	
M	Elasped Time (Minute)
N	
O	
P	
Q	
R	
S	Elapsed Time (Second)
T	
U	Elapsed Time (1/10 second)
V	
W	
X	
Y	
Z	
A$	INKEY$

SHARP

PROGRAM TITLE	COMPUTER-DESIGNED FLOWER	PROGRAM NO. P5–F–3	1
		CE–150 required	

[Outline]

You can enjoy your own various designs by using the graphic printer. Let's see how to draw a flower design.

[Operating Guide]

Pressing the [DEF] [A] enables the printout of a cute flower design.

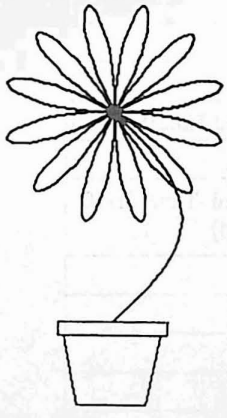
[Contents] (Formulas)

$X(i) = \mathrm{Sin}\ (6 \times i) \times \mathrm{Cos}\ (i + A) \times 80$
$Y(i) = \mathrm{Sin}\ (6 \times i) \times \mathrm{Sin}\ (i + A) \times 80$

Changing value of i from 1 to 30 per petal, 30 coordinates are connected with lines. Changing value A from 0° to 330° twelve times in 30° increment finds the coordinates of 12 varied petals.

[Printout]

The actual printout is colored.
Refer to page 4 .



[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	[DEF] [A]	>	printout

PROGRAM TITLE	COMPUTER-DESIGNED FLOWER	PROGRAM NO. P5–F–3	2

[Program List]

```
 10:"A":CLEAR :DIM
    X(30),Y(30):X(
    0)=0:Y(0)=0
 20:GRAPH
 30:GLCURSOR (100,
    -100):SORGN
 35:COLOR 3
 40:FOR A=0TO 60
    STEP 30
 50:FOR I=1TO 30
 60:X(I)=SIN (6*I)
    *COS (I+A)*80
 70:Y(I)=SIN (6*I)
    *SIN (I+A)*80
 80:NEXT I
 90:GOSUB "Q"
100:NEXT A
105:COLOR 2
110:FOR I=1TO 30
120:X(I)=SIN (6*I)
    *50
130:Y(I)=-I*5
140:NEXT I
150:GOSUB "P"
155:X=X(30):Y=Y(30
    )
160:LINE (X+40,Y)-
    (X-40,Y-10),0,
    0,B
170:LINE (X-35,Y-1
    0)-(X-25,Y-60)
    -(X+25,Y-60)-(
    X+35,Y-10)
180:TEXT :LF 5:END
200:"Q"GOSUB "P"
210:FOR I=0TO 30:X
    (I)=-X(I):NEXT
    I
220:GOSUB "P"
230:FOR I=0TO 30:Y
    (I)=-Y(I):NEXT
    I
240:GOSUB "P"
250:FOR I=0TO 30:X
    (I)=-X(I):NEXT
    I
260:GOSUB "P"
270:RETURN
300:"P"FOR I=0TO 2
    7STEP 3
310:LINE (X(I),Y(I
    ))-(X(I+1),Y(I
    +1))-(X(I+2),Y
    (I+2))-(X(I+3)
    ,Y(I+3))
320:NEXT I
330:RETURN

STATUS 1
                589
```

[Memory Contents]

A	Variables of FOR statement
B	
C	
D	
E	
F	
G	
H	
I	Variables of FOR statement
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	replacement of X(30)
Y	replacement of Y(30)
Z	
X(0~30)	X–coordinate
Y(0~30)	Y–coordinate

SHARP

PROGRAM TITLE	COMPUTER GRAPHICS	PROGRAM NO. P5–F–4	1
		CE–150 required	

[Outline]

It is great fun to generate a program that analyzes the extent of changes in a geometrical pattern. The array of triangles looks like ammonite in growth.

[Operating Guide]

Enjoy pattern change by inputting a variety of angles, increments and number of triangles.

[Example]

(Ex. 1) Geometrical pattern with 10 degrees, 3.5 increment and 30 triangles. (Refer to "Printout.")

(Ex. 2) Pattern with 20 degrees, 3 increment, and 35 triangles.

[Contents] (Formula)

$R = R + K$

(R is sum of increments, and its initial value is 5. Value K is added to each pattern.)

$T = T + S$

(T is sum of angles, and its initial value is S. Value S is added to each pattern.)

$X1 = R \times \mathrm{Sint}$

$Y1 = R \times \mathrm{Cost}$

$X2 = R \times \mathrm{Sin}(T + 60)$

$Y2 = R \times \mathrm{Cos}(T + 60)$

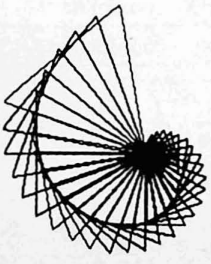
(0, 0) – (X1, Y1) – (X2, Y2) – (0, 0) are connected with straight lines.

The above pattern is repeated N times as the number of input.

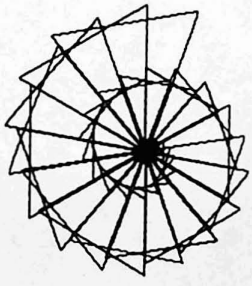
[Printout]

The actual printouts are colored. Refer to page 4.

(Ex. 1)



(Ex. 2)



PROGRAM TITLE	COMPUTER GRAPHICS	PROGRAM NO. P5–F–4	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	NO. OF TRIANGLES =_	
2	30 ENTER	ANGLE =_	
3	10 ENTER	INCREMENT =_	
4	3.5 ENTER	>	Printout
1	DEF A	NO. OF TRIANGLES =_	
2	35 ENTER	ANGLE =_	
3	20 ENTER	INCREMENT =_	
4	3 ENTER	>	Printout

[Program List]

```
10:"A":GRAPH :
   RANDOM
20:GLCURSOR (120,
   -200):SORGN
30:INPUT "NO. OF
   TRIANGLES=";N
40:INPUT "ANGLE="
   ;S
50:INPUT "INCREME
   NT=";K
60:T=-S:R=5
70:FOR I=1TO N
75:COLOR (RND 4-1
   )
80:R=R+K:T=T+S
90:X1=R*SIN T:Y1=
   R*COS T
100:X2=R*SIN (T+60
   ):Y2=R*COS (T+
   60)
110:LINE (0,0)-(X1
   ,Y1)-(X2,Y2)-(
   0,0)
120:NEXT I
130:END
```

[Memory Contents]

A	
K	Increment (input)
L	
M	
N	No. of Triangles (input)
O	
P	
Q	
R	Increment (Calculated value)
S	Angle (input)
T	Angle (Calculated value)
X1	Graphic X-coordinate 1
Y1	Graphic Y-coorcinate 1
X2	Graphic X-coordinate 2
Y2	Graphic Y-coordinate 2

```
STATUS 1
```

SHARP

PROGRAM TITLE	WORLD CLOCK	PROGRAM NO. P5-F-5 (Expanded, Standard)	1

CE–151 required in the expanded version.

[Outline]

What time is it in London? In New York?
In any other major cities in the world.
With this program, no cumbersome calculation is necessary.
A single-touch key operation gives you an instant indication of time in 30 major cities worldwide.
(The capacity in standard PC–1500 covers the world's 8 major cities.)
24 hour system is employed.

[Operating Guide]

Before program execution, Set Japan time as follows:

TIME = OO OO OO . OO OO [ENTER]
Month Day Hour Minute Second

REMARK:

Japan time can be easily found as follows:
Suppose you live in New York. Japan time is 14 hours ahead as derived from the time difference table shown on the next page.
If it's 7:00 in New York, it's 21:00 in Japan. (7:00 + 14:00 = 21:00)
Change the sign of your time difference from Tokyo and add it to your time. However, when the sum becomes more than 24:00, the day should be the next day in Japan.

[DEF] [A] : Pressing these keys displays the Japan time.

[2] : With this key pressed, the cities are sequentially changed as No. 1, No. 2, No. 3, ...

[8] : Upon depression of the key, the cities are changed in reverse order as No. 30, No. 29, No. 28, ...

Note: (1) Refer to the "Contents" for the cities.
(2) No consideration is given to leap years and summer times in some local areas.

[Example]

TOKYO	11 . 2 . 1:46
SINGAPORE	11 . 2 . 0:16
NEW YORK	11 . 1 . 11:46
LOSANGELES	11 . 1 . 8:46

PROGRAM TITLE	WORLD CLOCK	PROGRAM NO. P5—F-5 (Expanded, Standard)	2

[Contents] (Formula)

NO.	City name	Time difference	NO.	City name	Time difference	NO.	City name	Time difference
0	TOKYO	——	10	MONTREAL	−14	20	ZURICH	−8
1	SINGAPORE	−1.30	11	RIO	−12	21	HONG KONG	−1
2	NEW YORK	−14	12	MADRID	−8	22	SEOUL	0
3	LOS ANGELES	−17	13	AMSTERDAM	−8	23	PEKING	−1
4	SIDNEY	−16	14	DELHI	−3.30	24	HONOLULU	−19
5	CHICAGO	−19	15	NAIROBI	−6	25	ATHENS	−7
6	LONDON	−9	16	AUCKLAND	+4	26	CAPETOWN	−7
7	PARIS	−8	17	MOSCOW	−6	27	BERLIN	−8
8	ROME	−8	18	CAIRO	−7	28	MELBOURNE	+2
9	VANCOUVER	−17	19	TEHRAN	−5	29	ABUDHABI	−5

Note: Standard program includes 8 cities from No. 0 to No. 7.
The expanded program includes these 30 cities.

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	TIME = 1102 01.46 [ENTER]	TIME = 11 02 01.46	TIME set to Japan time
2	[DEF] [A]	TOKYO 11. 2. 1:46	
3	[8] [8]	PARIS 11. 1. 17:46 LONDON 11. 1. 16:46	In standard program
⋮	⋮	⋮	
4	[8] [2] [2]	TOKYO 11. 2. 1:46 SINGAPORE 11. 1. 23:16 NEW YORK 11. 1. 11:46	

PROGRAM TITLE	WORLD CLOCK	PROGRAM NO. P5—F-5 (Expanded, Standard)	3

[Program List] : Standard Version

```
 10:"A":CLEAR :
    WAIT 0
 20:P=7:DIM A$(P):
    DIM A(P)
 30:A$(0)="TOKYO":
    A(0)=0.0
 40:A$(1)="SINGAPO
    RE":A(1)=-1.3
 50:A$(2)="NEW YOR
    K":A(2)=-14
 60:A$(3)="LOSANGE
    LES":A(3)=-17
 70:A$(4)="SIDNEY"
    :A(4)=-16
 80:A$(5)="CHICAGO
    ":A(5)=-19
 90:A$(6)="LONDON"
    :A(6)=-9
100:A$(7)="PARIS":
    A(7)=-8
350:I=0:OI=1
360:GOSUB 500
370:GOSUB 650
380:B$=INKEY$
390:IF B$=""GOTO 3
    70
400:IF B$="8"GOTO
    430
410:IF B$="2"GOTO
    460
420:GOTO 370
430:OI=I:I=I-1
440:IF I<0LET I=I+
    P+1
450:GOTO 360
460:OI=I:I=I+1
470:IF I>PLET I=I-
    P-1
490:GOTO 360
500:CLS :USING :
    PRINT USING "&
    &&&&&&&&&";A$(
    I)
520:Y=A(I):IF Y<0
    LET X=-INT (
    ABS Y):GOTO 54
    0
530:X=INT Y
540:Y=(Y-X)*100
550:G=TIME
560:C=INT (G/10000
    )
570:D=INT (G/100)-
    C*100
580:E=INT G-C*1000
    0-D*100
590:F=INT ((G-INT
    G)*100)
600:K=C:L=D
610:M=E+X:N=F+Y:OI
    =1:RETURN
650:G=TIME
660:S=INT ((G-INT
    G)*100)
665:IF OI=1LET OI=
    0:GOTO 690
670:IF S=FRETURN
680:N=N+1
690:IF N>=60LET M=
    M+1:N=N-60
710:IF N<0LET M=M-
    1:N=N+60
730:IF M>=24LET L=
    L+1:M=M-24
750:IF M<0LET L=L-
    1:M=M+24
770:IF L<1GOTO 880
780:IF L<=28GOTO 9
    60
790:IF (K=1)+(K=3)
    +(K=5)+(K=7)+(
    K=8)+(K=10)+(K
    =12)=1LET Z=31
    :GOTO 840
800:IF K=2LET Z=28
    :GOTO 840
810:Z=30
840:IF L>ZLET L=L-
    Z:K=K+1
860:IF K>12LET K=K
    -12
870:GOTO 960
880:K=K-1
890:IF K<1LET K=K+
    12
910:IF K=2LET L=L+
    28:GOTO 960
920:IF (K=4)+(K=6)
    +(K=9)+(K=11)=
    1LET L=L+30:
    GOTO 960
930:L=L+31
960:CURSOR 10
966:USING :PRINT
    USING "###.";K
    ;USING "###.";
    L;USING "###";
    M;":";USING "#
    ##";N
970:F=S:RETURN
```

STATUS 1

1169

[Memory Contents]

A	
B	
C	Month (Japan)
D	Day (Japan)
E	Hour (Japan)
F	Minute (Japan)
G	Current Time
H	
I	City indicator
J	
K	Month for each city
L	Day for each city
M	Hour for each city
N	Minute for each city
O	
P	√
Q	
R	
S	√
T	
U	Number
V	Time Difference Month
W	Time Difference-Day
X	Time Difference-Hour
Y	Time Difference-Minute
Z	√
AS(29)	City Name Table
A(29)	Time Difference Table

PROGRAM TITLE	WORLD CLOCK	PROGRAM NO. P5-F-5 (Expanded, Standard)	4

[Program List] : Expanded Version

```
10:"A":CLEAR :
   WAIT 0
20:P=29:DIM A$(P)
   :DIM A(P)
30:A$(0)="TOKYO":
   A(0)=0.0
40:A$(1)="SINGAPO
   RE":A(1)=-1.3
50:A$(2)="NEW YOR
   K":A(2)=-14
60:A$(3)="LOSANGE
   LES":A(3)=-17
70:A$(4)="SIDNEY"
   :A(4)=-16
80:A$(5)="CHICAGO
   ":A(5)=-19
90:A$(6)="LONDON"
   :A(6)=-9
100:A$(7)="PARIS":
   A(7)=-8
110:A$(8)="ROME":A
   (8)=-8
120:A$(9)="VANCOUV
   ER":A(9)=-17
130:A$(10)="MONTRE
   AL":A(10)=-14
140:A$(11)="RIO":A
   (11)=-12
150:A$(12)="MADRID
   ":A(12)=-8
160:A$(13)="AMSTER
   DAM":A(13)=-8
170:A$(14)="DELHI"
   :A(14)=-3.3
180:A$(15)="NAIROB
   I":A(15)=-6
190:A$(16)="AUCKLA
   ND":A(16)=+4
200:A$(17)="MOSCOW
   ":A(17)=-6
210:A$(18)="CAIRO"
   :A(18)=-7
220:A$(19)="TEHRAN
   ":A(19)=-5
230:A$(20)="ZURICH
   ":A(20)=-8
240:A$(21)="HONG K
   ONG":A(21)=-1
250:A$(2 )="SEOUL"
   :A(22)=0
260:A$(23)="PEKING
   ":A(23)=-1
270:A$(24)="HONOLU
   LU":A(24)=-19
280:A$(25)="ATHENS
   ":A(25)=-7
290:A$(26)="CAPETO
   WN":A(26)=-7
300:A$(27)="BERLIN
   ":A(27)=-8
310:A$(28)="MELBOU
   RNE":A(28)=+2
320:A$(29)="ABUDHA
   BI":A(29)=-5
350:I=0:OI=1
360:GOSUB 500
370:GOSUB 650
380:B$=INKEY$
390:IF B$=""GOTO 3
   70
400:IF B$="8"GOTO
   430
410:IF B$="2"GOTO
   460
420:GOTO 370
430:OI=I:I=I-1
440:IF I<0LET I=I+
   P+1
450:GOTO 360
460:OI=I:I=I+1
470:IF I>PLET I=I-
   P-1
490:GOTO 360
500:CLS :USING :
   PRINT USING "&
   &&&&&&&&&";A$(
   I)
520:Y=A(I):IF Y<0
   LET X=-INT (
   ABS Y):GOTO 54
   0
530:X=INT Y
540:Y=(Y-X)*100
550:G=TIME
560:C=INT (G/10000
   )
570:D=INT (G/100)-
   C*100
580:E=INT G-C*1000
   0-D*100
590:F=INT ((G-INT
   G)*100)
600:K=C:L=D
610:M=E+X:N=F+Y:OI
   =I:RETURN
650:G=TIME
660:S=INT ((G-INT
   G)*100)
665:IF OI=ILET OI=
   0:GOTO 690
670:IF S=FRETURN
680:N=N+1
690:IF N>=60LET M=
   M+1:N=N-60
710:IF N<0LET M=M-
   1:N=N+60
730:IF M>=24LET L=
   L+1:M=M-24
750:IF M<0LET L=L-
   1:M=M+24
770:IF L<1GOTO 880
780:IF L<=28GOTO 9
   60
790:IF (K=1)+(K=3)
   +(K=5)+(K=7)+(
   K=8)+(K=10)+(K
   =12)=1LET Z=31
   :GOTO 840
800:IF K=2LET Z=28
   :GOTO 840
810:Z=30
840:IF L>ZLET L=L-
   Z:K=K+1
860:IF K>12LET K=K
   -12
870:GOTO 960
880:K=K-1
890:IF K<1LET K=K+
   12
910:IF K=2LET L=L+
   28:GOTO 960
920:IF (K=4)+(K=6)
   +(K=9)+(K=11)=
   1LET L=L+30:
   GOTO 960
930:L=L+31
960:CURSOR 10
966:USING :PRINT
   USING "###.";K
   :USING "###.";
   L;USING "###";
   M;":";USING "#
   ##";N
970:F=S:RETURN
```

```
STATUS 1
                    1802
```

SHARP

PROGRAM TITLE	DOT PATTERN DEVELOPMENT	PROGRAM NO. P5–F–6	1
		CE–150 required	

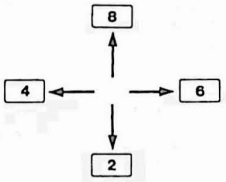
[Outline]

With this unique program, you can play a decisive role in pattern generation! Using the [2], [4], [6] and [8] keys, as well as alphanumeric keys [M] and [T], you can develop dot patterns at your discretion on the computer display.

Any pattern generated can be recorded for printout by using the [P] key.

[Operating Guide]

1. Key Operation



The moves are in response to each key.

2. Mode Setting

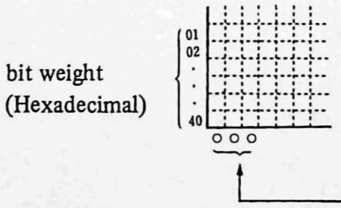
[M] : cancels the dot on display. (MOVE)
[T] : holds the dot on display. (TRACE)
[P] : sends the display pattern to the printer.
[E] : ends the program.

3. O–100 columns are available for patterns.

4. Remark:
Normal key operation causes a beep tone to confirm the entry.
With a key pressed other than specified, two beep tones warn the key operator.
When the dot range is going to exceed the specified range, three beep tones occur to give warning.

[Contents] (Formula)

1. Processing is performed in respones to the numeral keys and mode setting keys.
2. When set, the mode is indicated on the right side of the display.
3. When selected, printout mode [P] sends the pattern on the display to the printer, after which the mode is reset to pattern generation mode [M] This allows you to modify or upgrade the pattern.



The pattern is coded in the hexadecimal system. The red on the printer paper represents a completed dot.

4. The [E] key is pressed to end this program.

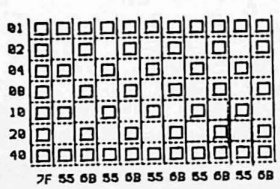
PROGRAM TITLE	DOT PATTERN DEVELOPMENT	PROGRAM NO. P5–F–6	2

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF A	DOT RANGE(0–>100)?_	This designates the dot range for pattern generation.
	11 ENTER	(One dot blinks at the upper left end.)	
	2	The dot moves downwards.	
	4	The dot moves upwards.	
	6	The dot moves to the right.	
	8	The dot moves to the left.	
	M	MOVE	This key is designated when moving the displayed dot while it is being erased.
	T	TRACE	This key is designated when moving the dot while leaving it at the displayed position.
	P	PRINT	Pattern printout
	E		Ends program

[Printout]

The actual printout is colored. Refer to page 4 .



PROGRAM TITLE	DOT PATTERN DEVELOPMENT	PROGRAM NO. P5–F–6	3

[Program List]

```
 10:"A":CLEAR
 20:CLS :WAIT 0:
    INPUT "DOT PAN
    GE(0->100)? ";
    N
 30:IF (N>=0)+(N<=
    100)<>2GOTO 20
 40:CLS :D=1:P=0
 41:CURSOR 21:
    PRINT " MOVE"
 50:WAIT 0:A$="":X
    =POINT P
 55:Y=DOR X:A$=""
 60:GCURSOR P:
    GPRINT Y
 65:D1=D
 70:A$=INKEY$
 80:IF A$<>""GOTO
    150
 90:A=&7F-D:A=AAND
    X
100:GCURSOR P:
    GPRINT A
105:D=0
110:A$=INKEY$
120:IF A$<>""GOTO
    150
130:D=D1:GOTO 55
150:BEEP 1,10,10:
    IF A$="8"LET D
    W=INT ((D1+1)/
    2):GOTO 250
160:IF A$="2"LET D
    W=D1*2:GOTO 25
    0
170:IF A$="6"LET P
    W=P+1:GOTO 260
180:IF A$="4"LET P
    W=P-1:GOTO 260
190:IF A$="M"LET M
    ODE=0:CURSOR 2
    1:PRINT " MOVE
    ":GOTO 130
200:IF A$="T"LET M
    ODE=1:CURSOR 2
    1:PRINT "TRACE
    ":GOTO 130
210:IF A$="P"LET M
    ODE=2:CURSOR 2
    1:PRINT "PRINT
    ":GOTO 300
220:IF A$="E"GRAPH
    :GOTO 600
230:BEEP 2,10,40:D
    =D1:GOTO 55
250:IF DW>64LET DW
    =64
251:IF MODE<>0GOTO
    255
252:A=&7F-D1:X=A
    AND X
253:GCURSOR P:
    GPRINT X:D=DW:
    GOTO 50
255:A=YOR DW
256:GCURSOR P:
    GPRINT A:D=DW:
    GOTO 50
260:IF PW<0LET PW=
    0:BEEP 3,10,20
    :GOTO 280
270:IF PW>NLET PW=
    N:BEEP 3,10,30
280:IF MODE=0GOTO
    286
284:GCURSOR P:
    GPRINT Y:GOTO
    290
286:A=&7F-D1:A=A
    AND X
287:GCURSOR P:
    GPRINT A
290:P=PW:D=D1:GOTO
    50
300:GCURSOR P:
    GPRINT X:D=D1:
    E=0
301:T$="123456789A
    BCDEF"
305:GRAPH :
    GLCURSOR (0,0)
310:COLOR 1:ROTATE
    1:CSIZE 1
320:C=110
330:FOR J=1TO 7
340:A=2^(J-1)
341:GOSUB 570
350:GLCURSOR (C,0)
    :LPRINT D$
360:C=C-15
370:NEXT J
380:GLCURSOR (0,-1
    5):SORGN
390:LINE (15,0)-(1
    20,0),0,2
400:FOR I=0TO N
410:A=POINT I
412:GLCURSOR (0,E)
    :SORGN
413:E=-16
415:LINE (15,-16)-
    (120,-16),0,2
420:FOR J=1TO 7
430:B=2^(J-1)
440:B=AAND B
450:C=120-(J*15)
470:IF B=0GOTO 490
480:LINE (C+3,-4)-
    (C+12,-13),0,3
    ,B
490:GLCURSOR (C,0)
    :LINE (C,0)-(C
    ,-15),1,2
500:NEXT J
505:GOSUB 570
515:GLCURSOR (2,-4
    ):COLOR 1
520:LPRINT D$
530:NEXT I
540:TEXT :LF 2
550:CURSOR 21:
    PRINT " MOVE":
    MODE=0
560:GOTO 50
570:F=INT (A/16):G
    =A-(F*16)
571:IF F=0LET F$="
    0":GOTO 574
572:F$=MID$ (T$,F,
    1)
574:IF G=0LET G$="
    0":GOTO 576
575:G$=MID$ (T$,G,
    1)
576:D$=F$+G$
579:RETURN
600:CSIZE 2:COLOR
    0:CLS :ROTATE
    0:TEXT
610:END
```

```
STATUS 1
                    1428
```

PROGRAM TITLE	DOT PATTERN DEVELOPMENT	PROGRAM NO. P5–F–6	4

[Memory Contents]

A	√	A$	Area for INKEY$	D1	Moving Dot Save
B	√	B$		DW	Dot Position Save during movement
C	Cursor Position of the printer	C$		PW	Cursor Position Save during movement
D	Moving Dot Position	D$	Print Data	MODE	Mode Save
E	Cursor Start Point of the printer	E$			
F		F$	Hexadecimal Code (Upper digits)		
G		G$	Hexadecimal Code (Lower digits)		
H		H$			
I	√	I$			
J	√	J$			
K		K$			
L		L$			
M		M$			
N	Dot range used	N$			
O		O$			
P	Cursor Position	P$			
Q		Q$			
R		R$			
S		S$			
T		T$	Hexadecimal conversion table		
U		U$			
V		V$			
W		W$			
X	Present Pattern	X$			
Y	Present Pattern + Moving Dot	Y$			
Z		Z$			

SHARP

PROGRAM TITLE	WORD MEMORY	PROGRAM NO. P5–F–7	1
		CE–150, CE–151 and CTR required	

[Outline]

By storing into the machine foreign word spellings and the equivalents in your native language, this program can help your memory work in foreign languages.

[Operating Guide]

[DEF] [A] : Translates foreign words into native words.
[DEF] [B] : displays native words, then input the spellings of foreign words.
[DEF] [C] : stores foreign and native words. (Addition and Modification)
[DEF] [D] : prints out the stored data.
[DEF] [F] : Order of word appearances in A and B can be selected either in random or in order of registration.
[DEF] [G] : inputs native and foreign words from the cassette tape, and also outputs them to the cassette tape.

1. Data registration/correction: Input approximately ten data.
2. Translate native words into foreign words. (Input the spelling).
3. Translate foreign words into the native.
4. Switch the order of word appearances.
5. Store data into the cassette tape, and load the data from the cassette tape.
6. Data list and output.

[Example]

Suppose the native language here is Japanese and the foreign language is English.

1. Data registration/modification

a) Registration

1.	FESTIVAL	MATSURI
2.	MOONLIGHT	GETSUKOU
3.	JOINT	SETSUGOU
4.	SPECIALITY	TOKUSYOKU
5.	WEATHER	TENKI
6.	QUEEN	JYOUOU
7.	INDUSTRIAL	SANGYOU
8.	GRASS	KUSA
9.	INNOVATION	KAKUSHIN
10.	DISTRIBUTE	BUNPAI SURU

b) Modification

For example, modify the entry, assuming "GRASS KUSA" in item 8 is input inadvertently as "KUSA" at the time of registration.

2. Japanese words to English Words
   a. "MATSURI" is displayed.
   b. Wrong spelling is input.
   c. Display the spelling of English word for N characters from the left. (N means 1 to the number of entries.)
   d. Input the remaining spelling other than displayed in Para.c above.
   e. If the spelling agrees, the following Japanese "GETSUKOU" is displayed. (The display in this case is in order of registration.)

3. English words to Japanese words
   a. "FESTIVAL" is displayed.
   b. Input either Y (in case you know the corresponding Japanese) or N (in case the corresponding Japanese is unknown to you).
   c. To input Y: The following English word "MOON-LIGHT" for "GETSUKO" is diaplayed.
      To input N: The Japanese "MATSURI" for "FESTIVAL" is displayed.

[Contents] (Formulas)

Up to 16 characters of a native or a foreign word can be registered in the standard capacity of PC–1500.
The pairs of foreign and native words which can be registered is up to 143.
The cassette tape file is called "F–N MEMORY".
The maximum number of N in registration is 16 pairs in the standard capacity of PC–1500.

[Printout]

```
 1 FESTIVAL
   MATSURI
 2 MOONLIGHT
   GETSUKOU
 3 JOINT
   SETSUGOU
 4 SPECIALITY
   TOKUSYOKU
 5 WEATHER
   TENKI
 6 QUEEN
   JYOUOU
 7 INDUSTRIAL
   SANGYOU
 8 GRASS
   KUSA
 9 INNOVATION
   KAKUSHIN
10 DISTRIBUTE
   BUNPAI SURU
```

PROGRAM TITLE	WORD MEMORY	PROGRAM NO. P5–F–7	3

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF C	ENTRY/UP-DATE? (E/U)	
2	E ENTER	N =	⟶ to step 3.
	U ENTER	UP-DATE NO. =	⟶ to step 9.
3	10 ENTER	F. LANG.(1) =	Pressing this key at English word input ends registration processing.
4	FESTIVAL ENTER	N. LANG.(1) =	
5	MATSURI ENTER	F. LANG. (2) =	
6	MOONLIGHT ENTER	N. LANG. (2) =	
	⋮	⋮	Input all the pairs.
7	DISTRIBUTE ENTER	N. LANG. (10) =	
8	BUNPAI SURU ENTER	ENTRY END	Processing is over.
		>	
9	8 ENTER	GRASS CHANGE? (Y/N)	
10	N ENTER	UP-DATE NO. =	⟶ to step 9.
	Y ENTER	F. LANG. =	⟶ to step 11.
	ENTER		Modification is over.
11	GRASS ENTER	N. LANG. =	
	KUSA ENTER	UP-DATE NO. =	⟶ to step 9.

[Key Operation Procedure]

Step No.	Input	Display	Remarks
1	DEF B	MATSURI	
2	HESTIVAL ENTER	MATSURI F	Wrong input
3	ESTIVAL ENTER	GETSUKOU	
4	MOONLIGHT ENTER	SETSUGOU	
	ENTER		Pressing this key ends the processing.
1	DEF A	FESTIVAL ····· Y/N?	
2	Y ENTER	MOONLIGHT ··· Y/N?	⟶ to step 2.
OR	N ENTER	MATSURI	⟶ to step 3.
	ENTER		Pressing this key ends the processing.

PROGRAM TITLE	WORD MEMORY	PROGRAM NO. P5–F–7	4

Step No.	Input	Display	Remarks
3	ENTER	MOONLIGHT ···Y/N?	⟶ 2 Pressing this key displays the next English word
1	DEF F	SEQ./RND.? (S/R)	
2	S ENTER		Designates sequential extraction.
OR	R ENTER		Designate random extraction
1	DEF G	CLOAD/CSAVE? (L/S)	
2	L ENTER		Load the data from the cassette tape
OR	S ENTER		Save the data to the cassette tape
1	DEF D		Print out English and Japanese words registered in this program.

[Program List]

```
 5:"C":WAIT 0:CLS
10:INPUT "ENTRY /
    UP-DATE?(E/U)
   ";A$
20:IF (A$="E")+(A
   $="U")<>1GOTO
   10
30:IF A$="U"GOTO
   150
40:CLEAR :INPUT "
   N=";N:DIM E$(N
   -1),J$(N-1)
50:FOR I=0TO N-1
60:A$="F.LANG.("+
   STR$ (I+1)+")=
   "
70:CLS :PRINT A$;
80:INPUT E$(I):
   GOTO 90
85:CLS :END
90:A$="N.LANG.("+
   STR$ (I+1)+")=
   "
100:CLS :PRINT A$;
110:INPUT J$(I)
120:NEXT I
130:PAUSE "ENTRY E
    ND"
140:END
150:INPUT "UP-DATE
    NO.=";A:GOTO
    170
160:END
170:IF A>NPAUSE "T
    ABLE OVER-FLOW
    ":END
180:CLS :PRINT E$(
    A-1);
190:INPUT " CHANGE
    ?(Y,N)";A$
200:CLS :IF (A$="Y
    ")+(A$="N")<>1
    GOTO 180
210:IF A$="N"GOTO
    150
220:INPUT "F.LANG.
    =";E$(A-1)
230:INPUT "N.LANG.
    =";J$(A-1)
240:GOTO 150
250:"D":FOR I=0TO
    N-1
```

(To be continued)

PROGRAM TITLE	WORD MEMORY	PROGRAM NO. P5–F–7	5

[Program List]

```
255:IF E$(I)=""
    GOTO 280
260:LPRINT (STR$ (
    I+1)+" ");E$(I
    )
270:LPRINT "    ";J
    $(I)
280:NEXT I
290:END
300:"F":CLS :INPUT
    "SEQ./RND.?(S/
    R)";A$
310:IF (A$="S")+(A
    $="R")<>1GOTO
    300
320:S=0:IF A$="R"
    LET S=1
330:END
340:"A":WAIT 0
345:IF S=1LET I=
    RND N:I=I-1:
    GOTO 360
350:FOR I=0TO N-1
360:CLS :PRINT E$(
    I);
370:INPUT "---Y/N?
    ";A$:GOTO 390
380:END
390:IF (A$="Y")+(A
    $="N")<>1GOTO
    370
400:IF A$="Y"GOTO
    420
410:CLS :WAIT :
    PRINT J$(I)
420:WAIT 0:IF S=1
    GOTO 345
430:NEXT I
440:CLS :WAIT 60:
    PRINT "TABLE E
    ND":END
450:"B":WAIT 0
460:IF S=1LET I=
    RND N:I=I-1:
    GOTO 475
470:FOR I=0TO N-1
475:K=0
480:CLS :PRINT J$(
    I);" ";MID$ (E
    $(I),1,K);
490:INPUT A$:GOTO
    510
500:END
510:B$=MID$ (E$(I)
    ,1,K)+A$
520:IF E$(I)=B$
    GOTO 540
530:K=K+1:GOTO 480
540:K=0:IF S=1GOTO
    460
550:NEXT I
560:CLS :WAIT 60:
    PRINT "TABLE E
    ND":END
570:"G":INPUT "CLO
    AD/CSAVE?(L/S)
    ";A$
580:IF (A$="L")+(A
    $="S")<>1GOTO
    570
590:IF A$="S"GOTO
    640
600:CLEAR
610:INPUT #"F-N ME
    MORY";N:DIM E$
    (N-1),J$(N-1)
620:INPUT #"F-N ME
    MORY";E$(*),J$
    (*)
630:END
640:PRINT #"F-N ME
    MORY";N
650:PRINT #"F-N ME
    MORY";E$(*),J$
    (*)
660:END
```

```
STATUS 1
                    1347
```

[Memory Contents]

A	Modification No.
B	
C	
D	
E	
F	
G	
H	
I	√
J	
K	√
L	
M	
N	Number of pairs to be regiotered
O	
P	
Q	
R	
S	Sequential/Random Extraction Flag.
T	
U	
V	
W	
X	
Y	
Z	
A$	√
J$ (N–1)	Native Word Registration Table
E$ (N–1)	Native Word Registration Table

SHARP CORPORATION
OSAKA, JAPAN

Printed in Japan
2G08Ō〈TMANE1022CCZZ〉⑤