TRS 323200
115 FERRY RD SOUTHPORT

Canola SX-300 SERIES

CANON CANOLA SX-300 SERIES

PROGRAMMING MANUAL

Courier Office Supplies.

Canon

326533

140 mm x 15 m

Canon

CONTENTS

Chapter 1 Quick Guide to Programming

Chapter 2 Program Examples

*marks given in the Contents and the Text indicate the following grade: * stands for elementary grade; ** middle grade; and *** high grade.

*marks given in the Contents and the Text indicate the following grade: * stands for elementary grade; ** middle grade; and *** high grade.

Table for Comparison of Two Numbers 60

Answers to Exercises 54

Exercise (5) 53

*** 6. Making Tables 50

Method of Making Tables

INDIRECT Instruction

*** 5. Array 45

Subroutine

** 4. Program with Subroutines 43

CHAPTER 1

QUICK GUIDE TO PROGRAMMING

1. Specifications

1. Digits

Display	Mantissa part 14 digits and sign	Exponent part 2 digits and sign
Memory (Full)	Mantissa part 14 digits and sign	Exponent part 2 digits and sign
Memory (Short)	Mantissa part 6 digits and sign	Exponent part 2 digits and sign

2. Capacity

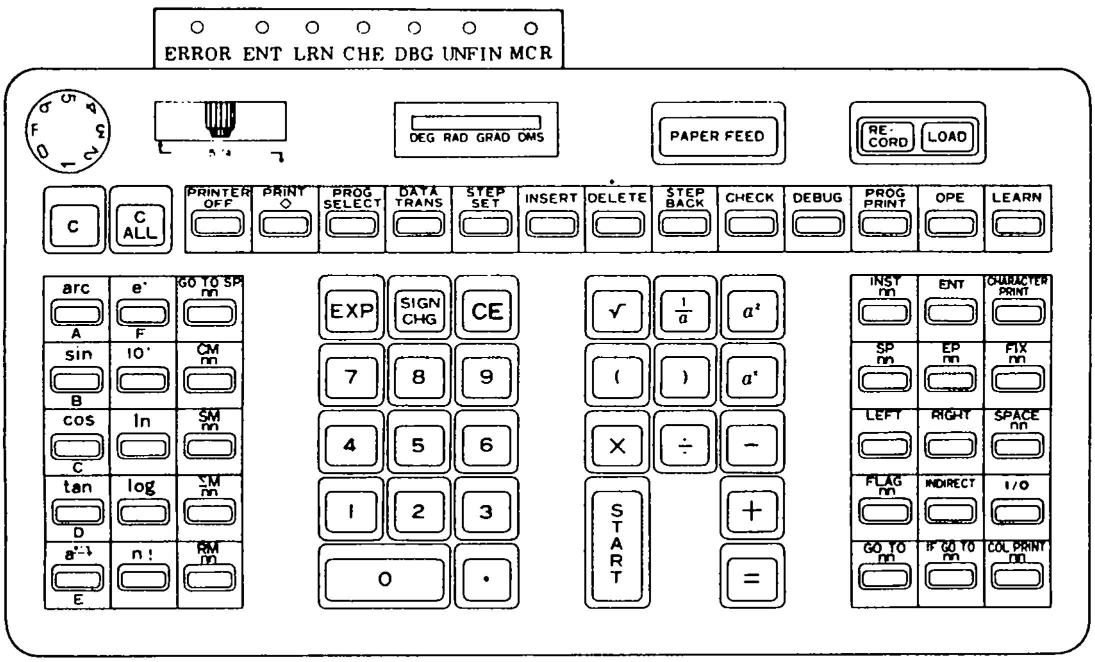
Internal expansion: Up to 100 data memories and 1000 program steps

External expansion (Maximum capacity): 500 data memories and 4000 program steps

3. Cartridge tape capacity

D-100 .... 100 data memories or 1000 program steps

D-500 .... 500 data memories or 5000 program steps



2. List of Program Instructions

Program instruction	Print symbol	Brief explanation	Key operation (omitted *nn*)
+, −, x, ÷	+, −, x, ÷	Addition, subtraction, multiplication, and division	[+] [−] [x] [÷]
(,)	(,)	Parentheses (Up to double parentheses)	[(] , [)]
=	=	Completion of calculation	[=]
0 − 9	0 − 9	Figure	[0] − [9]
.	.	Decimal point	[.]
EXP	EXP	Designation of exponents	[EXP]
SC	SC	Conversion of signs	[SIGN CHG]
e	*e*	Denotes constant *e*	[arc] [e^x]
π	π	Denotes constant π	[arc] [arc]
CE	CE	Clear buffer register	[CE]
E	E	Input of data	[ENT]
SIN	SIN	Find trigonometric functions ※	[sin]
COS	COS		[cos]
TAN	TAN		[tan]
ASIN	ASIN	Find inverse trigonometric functions ※	[arc] [sin]
ACOS	ACOS		[arc] [cos]
ATAN	ATAN		[arc] [tan]
DEG	DEG	Conversion of degree, minute, and second into decimal degree	[a°'" →]
DMS	DMS	Conversion of decimal degree into degree, minute, and second	[arc] [a°'" →]
CMA	*f*1	Clear all memories	[INST] [F] [1]
SM	SM	Store in a memory	[SM]
RM	RM	Recall memory contents	[RM]
CM	CM	Clear a memory	[CM]

※ Input and output are designated by the Angle Form Slide Switch.
Set to the Degree mode normally.

Program instruction	Print symbol	Brief explanation	Key operation (omitted *nn*)
ΣM	ΣM	Accumulation in a memory	[ΣM]
R	R	Designates right half of a memory	[RIGHT]
L	L	Designates left half of a memory	[LEFT]
IND	IND	Indirect memory addressing	[INDIRECT]
SP	SP	Denotes starting program	[SP]
EP	EP	Denotes ending program	[EP]
GOTO	GT	Unconditional jump	[GOTO]
GTSP	GS	Unconditional jump to subroutine	[GOTOSP]
IF ≠ 0	IFNZ	Jump if not zero	[IFGOTO] [=]
IF ≧ 0	IF+	Jump if zero or positive	[IFGOTO] [+]
IF < 0	IF–	Jump if negative	[IFGOTO] [–]
IFE	IFE	Jump if data input has been done	[IFGOTO] [ENT]
IFER	IFER	Jump if an error has occurred	[IFGOTO] [CE]
FLG	FLG	Destination of jump instruction	[FLAG]
BRANCH	IOf	Unconditional jump to destination step	[I/O] [F]
↑	FIX9	Round-up	[FIX] [9]
5/4	FIX5	Round-off	[FIX] [5]
↓	FIX0	Round-down	[FIX] [0]
e^x	e^x	Find exponential function to the base *e*	[e^x]
10^x	10^x	Find exponential function to the base 10	[10^x]
a^x	a^x	Find exponential function to the base *a*, *a* [a^x] *x* [=]※	[a^x]
LN	LN	Natural logarithm	[ln]
LOG	LOG	Common logarithm	[log]
√‾	√‾	Square root	[√‾]
$1/a$	$1/a$	Reciprocal	[$\frac{1}{a}$]
a^2	a2	Square	[a^2]
$\|a\|$	$\|a\|$	Absolute value	[INST] [8] [6]
N !	N !	Factorial	[n !]

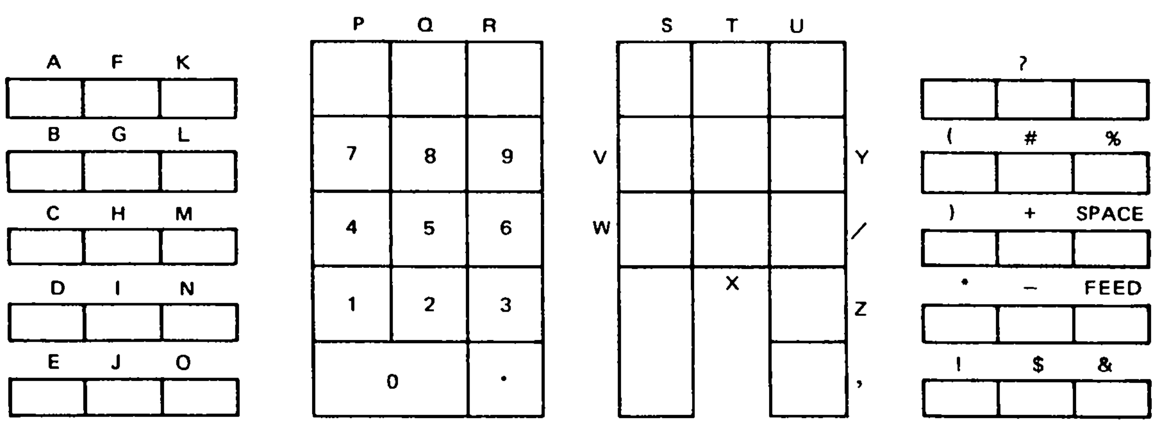
※ [+] [–] [×] [÷] will do.

Program instruction	Print symbol	Brief explanation	Key operation (omitted *nn*)
INT	INT	Take out integer only	[INST] [8] [8]
FRC	FRC	Take out decimal fraction only	[INST] [8] [7]
◇	◇	Print	[I/O] [0]
LF	LF	Feed paper by one line	[LINE FEED]
COL	COL	Print the figure in designated digits. No paper feeding.	[COL PRINT]
CHA	CHA	Put before and after character output	[CHARACTER PRINT]
SPC	→	Space by designated digits	[SPACE]
SED	*f*9	Continue calculation in an error state	[INST] [F] [9]
RED	*f*7	Nullify SED instruction	[INST] [F] [7]
SE	*f*5	Set error state	[INST] [F] [5]
RE	*f*3	Release error state	[INST] [F] [3]
DATA-P*n*	7*n*	Designates page for data	[INST] [7] [*n*]
PROG-P*n*	9*n*	Designates page for program	[INST] [9] [*n*]
NOP		No-operation instruction	[INST] [0] [0]

3. Table of Character Codes

n \ m	0	1	2	3	4	5	a	b	c	d
0			␢	0	@	P	␢	I		
1			!	1	A	Q	°			
2			◇	2	B	R	⌜			
3			#	3	C	S	⌟			
4			$	4	D	T	π			
5			%	5	E	U	^{x}			
6			&	6	F	V				
7			£	7	G	W	$\sqrt{\ }$			
8			(	8	H	X	×			
9			)	9	I	Y	÷			
a	LF		*	:	J	Z	a			
b			+	→	K	Σ	b			
c			,	<	L	¥	c			
d			−	=	M	g	d			
e			.	>	N	h	e			″
f			/	?	O	i	f			°

Arrangement of Character Keys



4. Manual Calculation (Round-off to six decimal places)

By depressing the keys following algebraic expressions, the calculator will automatically execute the calculations. The order of calculation priority is as follows:

1. Parentheses (Up to double parentheses)
2. Functions
3. Multiplications and Divisions
4. Additions and Subtractions

o $9 - 6 + 3 = 6.000000$ 9 [−] 6 [+] 3 [=]

o $4 \times (-3) = -12.000000$ 4 [x] 3 SC [=]

The [SIGN CHG] key converts the sign of displayed figure. Hereafter, the [SIGN CHG] key is expressed as [SC].

o $\sqrt{625} = 25$ 625 [√]

o $\frac{36}{1+2+3} = 6.000000$ 36 [÷] [(] 1 [+] 2 [+] 3 [)] [=]

o $5^2 = 25$ 5 [a^2]

o $2^8 = 256$ 2 [a^2] [a^2] [a^2]

o $5^{-3} = 0.008000$ 5 [a^x] 3 [SC] [=]

o $\log 5 = 0.698970004336$

5 [log]

o $\sin 25°30'15'' = 0.430576733524$

Set switch to DMS mode

25.3015 [sin]

o $30 + 4 \times (1.56 \times 10^2 - 50 \times \cos 30°36') = 481.851595$

Set switch to DMS mode

30 [+] 4 [x] [(] 1.56 [EXP] 2 [−] 50 [x] 30.36 [cos] [)] [=]

o $81 \div 9 + 27 \div 9 + 36 \div 9 = 16.000000$

81 [÷] 9 [SM] [0] [0] + 27 ÷ [RM] [0] [0] + 36 ÷ [RM] [0] [0] [=] .
Let the figure 9 memorize into address 0 0 by key operation of [SM] [0] [0] . Then recall it by [RM] [0] [0] where 9 is needed.

o $\pi \times 3^2 = 28.274334$

[arc] [arc] [×] 3 [a^2] [=]

π (pi) can be found by depressing [arc] keys twice as [arc] [arc] .

☆ Print of characters ☆

o CANON

[CHARACTER PRINT] [C] [A] [N] [O] [N] [CHARACTER PRINT]

Exercise (1)

(Decimal Point Selector Dial ..... 6, Round Form Slide Switch ..... 5/4)

1. $456 + 789 = 1245.000000$

2. $56.78 - 88 = -31.220000$

3. $-2.7 \times 7.89 = -21.303000$

4. $(7890 + 192) \times (3.84 \times 10^3 - 512) = 26896896.000000$

5. $0.1 \div 2 \times \{1 + 2 \times (2.7282 + 34595 \times 10^{-4}) + 0.5\} = 0.693770$
   -9×10^{-8} is input by key operation of [9] [SC] [EXP] [8] [SC]

6. $\dfrac{63.6 \times 0.328}{5.87^2} = 0.605417$

7. $\sqrt{3^2 + 4^2} = 5$

8. $\cos 15°30' = 0.963630453209$

9. $\tan 12.34° = 0.218766692332$

10. $\sin^{-1} 0.2221157789 = 12.8333333279$

11. $e^{1.4} = 4.05519996684$

12. $e^{-2.5} = 0.0820849986239$

13. $3.6^{1.3} = 5.286805$

14. $1.4^{3/7} = 1.4^{(3 \div 7)} = 1.155118$

15. $\log 3 = 0.47712125472$

16. Obtain x when $\log x = 0.6020599914$. $x = 4.00000000066$. Use [10^x] key.

17. $\ln 5 = 1.60943791243$

18. Obtain x when $\ln x = -1.203972804$. $x = 0.300000000098$. Use [e^x] key.

19. $y = 3x^3 + 2x^2 - x$ Calculate y by letting x ($x = 2$) memorize in address 10.
    $y = 30.000000$

20. Print your name.

5. What is Programmed Calculation ?

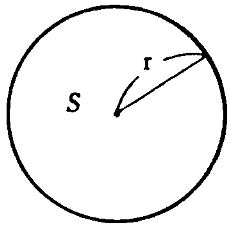
Calculation executed automatically according to the calculation procedure memorized in the calculator memory is called Programmed Calculation.

5-1 How to Program

Example 1:

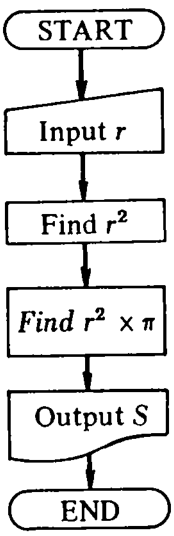
Calculation of the area of a circle $S = \pi r^2$

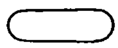
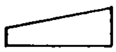
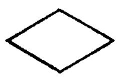
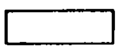
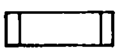
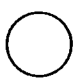
Let's explain the programming with a simple example.



1) Flow chart

The flow chart arranges the processing procedure of a problem in a readily visible diagram. Writing a flow chart will help you understand the calculation procedure more clearly than analyzing it only in your head.



Symbol	Usage and meaning
	Start or end of a program ((START) , (END)).
	Data input.
	Decision. The condition for branching is written inside the figure as <B = 0>
	Output of computed result.
	Process
	Predefined Process (Subroutine etc.)
	Connector. As Ⓐ , a symbol is written inside and connected with same symbol.
	Flowlines. Denotes direction of calculation procedure.

2) Perform manual calculation according to flow chart and check whether procedure is right.

Manual operation

2 (= r)

[a^2]

[×]

[arc] [arc] (=π)

[=]

Operation by an ordinary calculator is shown on the left side. To program this,

1. Put SP n n at the start of the program and EP n n at the end.
2. Change to the ENT instruction where variable r is entered.
3. Enter ◇ (print instruction) where the result is printed out.

Thus the program is completed.

Program

SP

00

(E)※

a^2

×

π (= [arc] [arc])

=

◇

EP

00

※ As SP n n has entry function as well as program head function, ENT instruction after SP n n can be omitted.

6. Program Input and Output

To let the calculator memorize the program somehow is called program input, and to have the memorized program printed out is called program output.

Input is done from keyboard or from the cartridge tape. Output methods are the printing out on roll paper and the transferring to the cartridge tape.

6-1 Program Input

1) Input Procedure Through Keyboard

① Depress the [LEARN] key.

② Depress keys [C] [C ALL] . (Note 2)

All the programs stored will be cleared. Depress the [C] key alone when you do not wish to clear them.

③ If necessary, depress [STEP SET] [n] [n] [n].

If this procedure is omitted, a program will be input from the 000 step. But when a program is to be input from a certain step (for example, 1 2 3 step), depress as [STEP SET] [1] [2] [3].

④ Depress keys corresponding to program instructions.

List of program instructions and their corresponding keys are given on page 2 – 4.

⑤ Depress the [OPE] key.

Correcting program instruction during input

When you notice any miss operation of keys during input of program in above operation ④, correct the instruction in the following manner.

① Go back to the step whose instruction is wrong by depressing the [STEP BACK] key as many times as required.

② Input right instruction over prior one. (Note 1)

③ Depress the [S] key to put the step to the original position.

One depressing the [S] key makes one step go forward.

(Note 1)

The following procedure shall be taken to amend only *nn* parts of the character code, symbol, address, and print instructions.

(Example)

To change RM 10 to RM 11, go back to the step where 10 is input and depress [INST] [1] [1].

(Note 2) Working of [C] [C ALL]

Mode	Memory contents	Program contents	Program step
OPE	All cleared	Remaining	Back to step 000
LRN	Remaining	All cleared	Back to step 000
CHECK	Remaining	Remaining	Back to step 000
DEBUG	Remaining	Remaining	Back to step 000

2) Input Procedure from Cartridge Tapes

The transferring can be performed in all the modes.

① Confirm that [DATA TRANS] is not locked.

② Depress [C] [C ALL] in the LEARN mode to clear all the program stored. When you do not wish them to be cleared, depress the [C] key alone.

③ Depress [STEP SET] [*n*] [*n*] [*n*], if necessary.

If this procedure is not taken, a program will be input from the 000 step, but when you especially wish them to be input from a certain step (for example, 1 2 3 step), depress [STEP SET] [1] [2] [3] .

④ Set the cartridge to the unit.

⑤ Depress the [LOAD] key.

The MCR Lamp stays on while the tape is running. If a program does not end within one cartridge, return to Step ④ .

⑥ When LOAD ends, switched to the OPE mode automatically.

6-2 Program Output

1) Procedure of Printing out Program on Roll Paper

① Depress the [C] key.

When you wish to print out from a certain step (for example, 1 2 3 step), depress [STEP SET] [1] [2] [3].

② Depress the [PROG PRINT] key in the CHECK mode.

When there is no program instruction (strictly speaking, when 2 blank steps are read sequencially), program printing is automatically stopped.

③ Depress the [C] key to stop the printing in process. (In this case, a mode must be set anew to shift to the next operation.)

④ Put the print head to the end by depressing the [C] key.

2) Procedure of Recording Program on Cartridge Tapes

① Confirm that the [DATA TRANS] key is not locked.

② Depress the [C] key.

When you especially wish to transfer a program from a certain step (for example, 1 2 3 step), depress [STEP SET] [1] [2] [3].

③ Set the cartridg to the unit.

④ Depress the [RECORD] key.

The MCR Lamp stays on while the tape is running. At the end of recording, the last step of the recorded program is displayed. If more recording is necessary, return to Step ③.

7. Data (Constants) Input and Output

Data or constants are sometimes memorized in memories before program execution. To memorize data in this manner is called data input, and to put out data memorized in memories is called data output.

7-1 Data Input

1) Procedure of Data Input by Keys

① Depress the [OPE] key. (If all the status indicating lamps are turned out, it is all right.)

② Enter data.

③ Depress [SM] [*n*] [*n*] to store data in the *n n* address.

2) Procedure of Data Transfer (LOAD) from Cartridge Tapes

This can be performed in all modes. The end of LOAD leads automatically to the OPE mode.

① Lock [DATA TRANS] key.

The leading address of the memory group to be input (for example, address 12) is set as follows:

Depress [STEP SET] [1] [2] [0].

② Set the cartridge to the unit.

③ Depress the [LOAD] key.

The MCR Lamp stays on while the tape is running. If the data does not end within one page, return to Step ②.

④ Release the locked [DATA TRANS] key.

7-2 Data Output

1) Procedure of Data Transfer (RECORD) to Cartridge Tapes

① Lock the [DATA TRANS] key.

② Set the leading address of memory group to be transferred (for example, address 12) as follows:

Depress [STEP SET] [1] [2] [0].

③ Set the cartridge to the unit.

④ Depress the [RECORD] key.

The MCR Lamp stays on while the tape is running. At the end of recording, the last address of the recorded data is displayed. If more recording is necessary, return to Step ③ .

⑤ Release the locked [DATA TRANS] key.

8. Execution Procedure of Programmed Calculation

① Depress the [OPE] key. (All status indicating lamps are turned out.)

② Depress the [C] key.

2-1 When a program starts at steps other than the 000 step (for example, 1 2 3 step), depress the key in the following manner instead of depressing the [C] key:

[STEP SET] [1] [2] [3]

2-2 Because the beginning of the program is SP *n n*, depressing [GoToSP *n n*] [*n*] [*n*] instead of depressing the [C] key is available.

2-3 When the beginning of the program is SP 8*a* depress [PROG SELECT] [A]* instead of the [C] key.

* In the case of 8b, 8c, 8d, and 8e, they correspond to the [B], [C], [D], and [E] keys, respectively.

③ When it is necessary to enter the numerals after 2-1, enter the data according to one of the following operations and start the calculation by depressing the [S]* key. When not, depress the [S] key only and execution will be started. Furthermore when it is necessary to enter the numerals in the above procedure 2-2 or 2-3, depress [GoToSP *n n*] [*n*] [*n*] or [PROG SELECT] [A] instead of depressing the [S]* key after entering numerals. If not, depress these keys as mentioned in 2-2, 2-3 and execution will be started.

123.4	[1] [2] [3] [·] [4]
1.234×10^2	[1] [·] [2] [3] [4] [EXP] [2]
1234×10^{-1}	[1] [2] [3] [4] [EXP] [1] [SC]

* [S]; abbreviation of [START] key hereinafter.

④ Start with Operation ③ when calculations from the second time are made by the same program.

9. Finding Programming Errors

When correct answers are not obtained after program execution, errors can be found in the following manner.

1) Check that a right program is input by printing out the program contents on the roll paper in accordance with the procedure given on page 13.

2) Calculation and execution step by step

   You can find the place where errors exist by executing the program step by step in the following manner.

   ① Depress the [DEBUG] key.

   ② Depress the [C] key.

   When you wish to start with any step other than the 000 step (for example, 1 2 3 step), depress [STEP SET] [1] [2] [3].

   ③ If the [S] key is depressed, instruction of that step is executed and advance to the next step.

   ④ When the entry lamp is lit half-way, input variable (data).

10. Correction of Program

When errors in the program input are found, correct them in the following manner.

1) When changing it to different instruction

① Depress the [CHECK] key.

② Set the step to a place where the correction is to be made.

③ Input the right instruction. (Note 1)

2) When addition (insertion) is made

① Depress the [CHECK] key.

② Set the step to a place where the insertion is to be made.

③ Lock the [INSERT] key.

④ Input instructions to be added or inserted through the keyboard. (Note 1) (You can insert as many instructions as you require, and all the stored programs are moved backwards successively to be protected, automatically.)

⑤ Release the locked [INSERT] key.

Key operation	Display
[CHECK]	
[STEP SET]	
[0] [2] [2]	0 2 2
[INSERT] (Lock)	
	0 2 2
[FIX] [5]	
	0 2 3
[INST nn] [0] [2]	
	0 2 4
[INSERT] (Unlock)	

Before insertion

0 0 2 0	RM
0 0 2 1	0 1
0 0 2 2	◇
0 0 2 3	LF

After insertion

0 0 2 0	RM
0 0 2 1	0 1
0 0 2 2	FIX 5
0 0 2 3	0 2
0 0 2 4	◇
0 0 2 5	LF

(Note 1)

When *n n* parts of the character code, symbol, address, and print instructions are changed or added, *n* *n* must be input as [INST] [*n*] [*n*] as shown in the example.

3) When deletion is needed

① Depress the [CHECK] key.

② Set the step to a place where the deletion is to be made.

③ Depress the [DELETE] key.

The trailing istruction will advance in turn.

④ If you want to erase more than one step, depress the [DELETE] key as many times as required.

Key operation
[CHECK]
[STEP SET] [0] [0] [7]
[DELETE]
[DELETE]

Before deletion

0 0 0 5	RM
0 0 0 6	0 1
0 0 0 7	FIX 0
0 0 0 8	0 5
0 0 0 9	◇
0 0 1 0	LF

After deletion

0 0 0 5	RM
0 0 0 6	0 1
0 0 0 7	◇
0 0 0 8	LF

Because the program step moves with the correction, and without program printing it is difficult to know the step numbers of the instructions following the corrected parts, it is more convenient to correct instructions from the end of the program, as follows:

SP
00
INSERT ①
DELETE ②
INSERT ③
EP
00

It is more convenient to make correction from the end, i.e., in order of ③, ②, ① .

11. Rules When the Capacity is Extended over One Page

The capacity of one page is as given below.

1 data page .... 100 memories

1 program page .... 1000 steps

Follow the rules given below when the capacity is extended over one page.

1. Ordinary calculation

Use a 3-digit number as manual input or output for a memory.

[Example] Storing a certain number in the address 1 2 3 (Address 23 of Page 1).

[SM] * [1] [2] [3]

* [CM] [ΣM] [RM] should be handled similarly.

2. Programmed calculation

2-1 Use a 4-digit number for the step set.

[Example] Setting to the step 1 2 3 4 (Step 2 3 4 of Page 1).

[STEP SET] [1] [2] [3] [4]

2-2 Use the DATA-P*n* instruction for storing or recalling the data.

[Example] Storing a datum in the address 1 2 3 (Address 23 of Page 1)

```
   |
   |       It means Page 1.
DATA-P1  ( [INST] [7] [1] )
   SM
   23
   |
```

This DATA-P1 remains effective until the next DATA-P*n* instruction comes.

2-3 When memories after Page 1 are specified by the INDIRECT instruction, DATA-P*n* instruction is unnecessary. Even pages are specified by indirect addressing.

3. Near page boundary

3-1 If you come to a new page during program loading by keys, begin the new page with a step set and continue loading.

(A program is automatically loaded on the next page during cartridge loading.)

3-2 During program printing, interrupt printing at step 999 of each page (by depressing the [C] key), begin a new page with the step set and continue printing.

3-3 Instructions are moved by inserting or deleting within one page alone.

3-4 Use an unconditional jump instruction when a program runs over to the next page.

[Example]

Page 0

PROG-P1([INST] [9] [1]) — It means Page 1.

GoTo

23

CE

CE

CE

FLG

23

Page 1

We recommend to enter some meaningless instructions. They may be lost, when some additional instructions are inserted and the program instructions at the end of the page would overflow. Do not enter 0's here as a meaningless instruction. If you do, recording will be stopped here during recording to a cartridge.

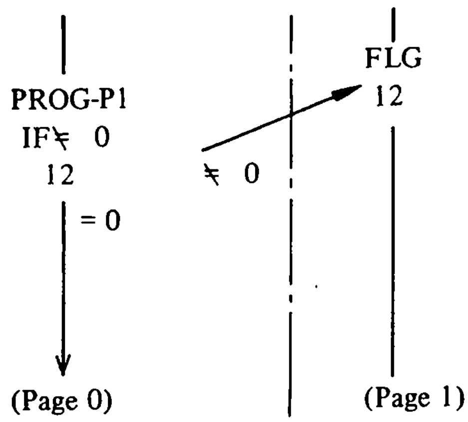
4. Jumps

The symbols related to jump instructions are effective within one page alone.

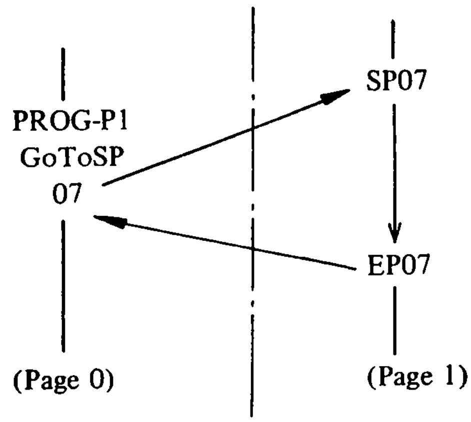
Specify a page with the PROG-P*n* instruction for a jump to another page.

Since the step to jump is specified by a 4-digit number in the case of the absolute address system (BRANCH), there is no need for specifying a page with the PROG-Pn instruction.

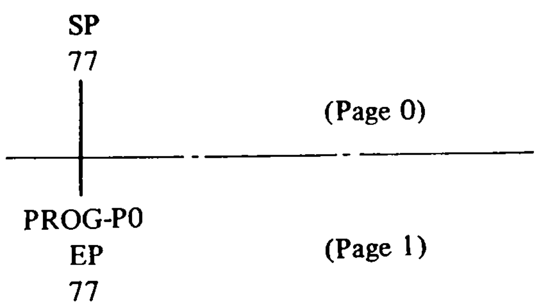
4-1 Conditional jump



4-2 Subroutine jump



4-3 Returning to the SP *n n* instruction from the EP *n n* instruction.



CHAPTER 2

PROGRAM EXAMPLES

* marks given in examples and exercises in the text indicate the following grade.
*, **, and *** means elementary, middle, and high grade, respectively.

1. Straight Line Program

This is a program for which a flow chart is drawn on a straight line without branching or repeating a certain process.

•Example 1   Calculation of area of a triangle

Find S in $S = \frac{1}{2} a h$ where h is the length of the base of the triangle and a is height.

Study Points:
1. Head of a program and end of a program.
2. Usage of memories.
3. Rounding
4. Input of constants

1. Head of a program and end of a program

Put SP n n at the head and EP n n at the end. SP n n (n represents numbers from 0 to 9) has ENT function at the same time for inputting variable.

SP n n and EP n n shall be handled in combination in one program as SP12 and EP12.

2. Usage of memories

There are the following instructions about memories.

CM*nn* : Clear the contents of the memory of the *n n* address.

SM*nn* : Store the contents of the buffer register (strictly speaking, changing them into scientific floating values) in a memory of the *n n* address.

ΣM*nn* : Accumulate the contents of the buffer register into a memory of the *n n* address.

RM*nn* : Recall the contents memorized in a memory of the *n n* address to the buffer register.

3. Rounding

Decimal Point Selector Dial and Round-Form Slide Switch do not work during programmed calculations. Therefore the values are printed out with exponents unless the following instructions for rounding are given.

			Key operation
Designation of round-form	Round-up	(↑)	[FIX] [9]
	Round-off	(5/4)	[FIX] [5]
	Round-down	(↓)	[FIX] [0]
Designation of decimal digits	Number of digits below the decimal point is designated as *nn*.		

Example Round-off, to 4 decimal places

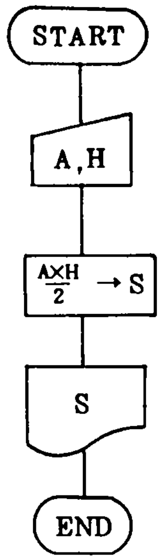
[FIX] [5] [0] [4]

Round down below the decimal point

[FIX] [0] [0] [0]

4. Input of constants

To enter constants in a program, simply depress the numeral keys (including decimal point and EXP) according to the expression.



Operation procedure

1. [LEARN]
2. [C] [C ALL]
3. Input the program.
4. [OPE]
5. [C]
6. a [S]
   h [S]
   S ◇

Repeat operation 6.

($a = 2$ $h = 3$ $S = 3.000$)

STEP	Instruction	Remark		Key operation	
000	SP			SP	
1	10		Input A		1 0
2	SM			SM	
3	10	A			1 0
4	E		Input H.	ENT	
5	SM			SM	
6	11	H			1 1
7	RM			RM	
8	10	A			1 0
9	×			×	
010	RM		$S = \frac{1}{2} A H$	RM	
1	11	H			1 1
2	÷			÷	
3	2			2	
4	=			=	
5	5/4			FIX	5
6	04				0 4
7	◇			◇	
8	EP			EP	
9	10				1 0
0					
1		Furthermore, this calculation can be made by			
2		the following simple program.			
3					
4					
5					
6					
7					
8					
9					
000	SP			SP	
1	11		Input a		1 1
2	×			×	
3	E		Input h	ENT	
4	÷			÷	
5	2			2	
6	=			=	
7	5/4			FIX	5
8	04				0 4
9	◇			◇	
010	EP			EP	
1	11				1 1
2					
3					
4					
5					
6					
7					
8					
9					

Exercise (2)

(Straight line program)

* (21) Find the area S by the following Helon Formula by giving the length of triangle sides *a, b, c*.

$$S = \sqrt{s(s-a)\ (s-b)\ (s-c)}$$

$$s = \frac{a+b+c}{2}$$

($a = 3$ $b = 4$ $c = 5$ S = 6.000)

* (22) Find x and y of the following system of two linear equations after giving coefficients.

$$\begin{cases} ax + by = c \\ lx + my = n \end{cases}$$

$$x = \frac{cm - bn}{am - bl}$$

$$y = \frac{an - cl}{am - bl}$$

($a = 1$ $b = 2$ $c = 5$ $l = 2$ $m = 3$ $n = 8$; $x = 1.000000$ $y = 2.000000$)

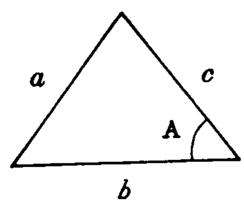
* (23) What will be the amount ($S) after n years on annual interest i if a certain amount of $R is saved at the end of a year?

$$S = R\left[\frac{(1+i)^{n+1} - 1}{i}\right]$$

(R = 10,000 $i = 0.07$ $n = 5$ S = 71,533)

* (24) Find the area S by giving two sides length of a triangle, b and c, and angle A (in degree, minute, and second mode).

$$S = \frac{1}{2} b\ c\ \sin A$$



($b = 2$ $c = 3$ A = 20° 15′ S = 1.038)

2. Program with Branching

This type is a program of which the calculation procedure is branching in two depending on the condition.

Example 2-1 Decision of comparing two figures

Here, we are dealing with the decision of comparing two figures. At that step, one way of the two is selected after judging if the data is greater (or less) than the given value.

Arrange the program such that the result of addition of two angles (both are represented in Degree, Minute, Second Mode.) becomes the answer just as it is when the calculated value does not exceed 360°. When it exceeds 360°, the answer is the value obtained when 360° is subtracted from the calculated value.

Study Points;

1. Conditional jump instructions (IF $<$ 0, IF $\fallingdotseq$ 0)
2. Conversion; Degree, Minute, Second Mode $\rightleftarrows$ Decimal degree Mode

1. Jump instructions

The function to process the thinking pattern of "If ....., do .....", when the program is branched to right flow or left flow calculations, is called a decision function. As the calculation flow jumps in a flow chart, this is also called a jump function. And the jumps of which destination change depending on the conditions as explained above are the conditional jumps, while those that jumps unconditionally without regarding the conditions are called the unconditional jumps.

This calculator has the following versatile jump functions.

(1) Full of various kinds, IF $<$ 0, IF $\fallingdotseq$ 0, IF $\geqq$ 0, IF ENT, and IF ER, for conditional jumps.

(2) There are almost limitless usable combinations of destination symbol $n\ n$ (about 100 pairs)

(3) However much complicated jumps may be arranged, there is no problem. What is to be noted is that FLG $n\ n$ as a destination must be only one in one program page (including main and subroutine programs).

(4) An absolute address system jump instruction, (BRANCH), that allows unconditional jump to a specified step is, added to the symbol system jump instruction, GO TO $n\ n$.

IF < 0: (Key operation: [IFGOTO] [−])

Program jumps to FLG *n n* if the sign of the figure in the buffer register is negative or continues if it is zero or positive. As to the comparison of two figures, see the table attached at the end of this manual.

Other jump instructions will be explained in each section of study.

2. Conversion of degree minute second to decimal degree

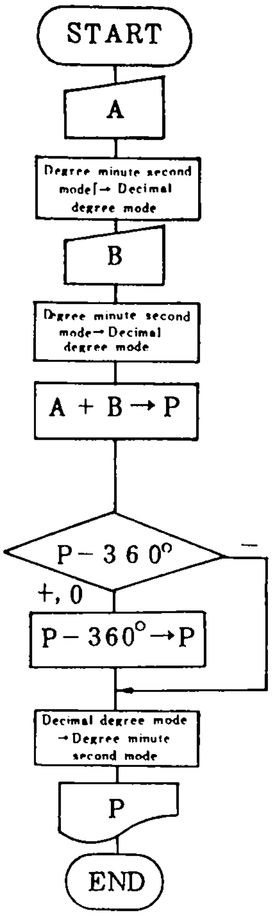
As angles are often represented in degree, minute, and second mode, they must be calculated after conversion into the decimal degree mode when the ordinary +, −, x, and ÷ operations are performed in the program.

For this purpose there are conversion instructions as follows:

Degree, minute, second ⟶ Decimal degree (key operation; [0 ' " *a* →])

123° 45' 67.8" is dealt with as 123.45678.

Decimal degree ⟶ Degree, minute, second (key operation; [arc] [0 ' " *a* →])



Operation procedure

1. [LEARN]
2. [C] [C ALL]
3. Input the program.
4. [OPE]
5. Set the Angle Form Slide Switch to the DEG position.
6. [C]
7. A [S]
   B [S]
   P ◇

Repeat operation 7.

(A = 123° 12′ 34″
B = 300° 23′ 45″
P = 63° 36′ 19″)

STEP	Instruction	Remark		Key operation	
000	SP			SP	
1	21		Input angle A*		2 1
2	DEG		Convert degree, minute, and second mode into	$a^{\circ\prime\prime}\rightarrow$	
3	SM		decimal degree mode.	SM	
4	05	A			0 5
5	E		Input angle B*	ENT	
6	DEG		Convert degree, minute, and second mode into decimal	$a^{\circ\prime\prime}\rightarrow$	
7	SM		degree mode.	SM	
8	06	B			0 6
9	RM			RM	
010	05	A			0 5
1	+			+	
2	RM			RM	
3	06	B			0 6
4	=			=	
5	SM			SM	
6	07	P	P = A + B		0 7
7	(RM			RM	
8	07)	P			0 7
9	−			−	
020	3			3	
1	6			6	
2	0			0	
3	=			=	
4	(SM			SM	
5	08)	P−360			0 8
6	IF<0		When P-360 is negative,	IFGOTO	−
7	01		namely, A + B is less		0 1
8	(RM		than 360°, it jumps to	RM	
9	08)	P−360	FLG 01 (Step 32).		0 8
030	SM			SM	
1	07				0 7
2	FLG			FLAG	
3	01				0 1
4	RM			RM	
5	07	P			0 7
6	DMS		Convert decimal degree mode into degree, minute, and second mode.	arc	$a^{\circ\prime\prime}\rightarrow$
7	5/4		Round-off to 6 decimal	FIX	5
8	06		places.		0 6
9	◇			◇	
040	EP			EP	
1	21				2 1
2					
3					
4			* 150° 12′ 34.5″ is		
5			dealt with as		
6			150.12345		
7					
8					
9					

This program can be shortened by omitting step 17, 18 (RM, 07), step 24, 25 (SM, 08), and 28, 29 (RM, 08).

**** Example 2-2 The roots of quadratic equation**

This is the case where a calculation flow changes according with positive or negative of a certain value (value of discriminant in this case.)

> Write a program to find the roots by giving coefficients of the following quadratic equation, a, b, and c. (Program classification No. X1-1090)
>
> $ax^2 + bx + c = 0$

Study Point: Unconditional jump instructions (GOTO, BRANCH)

Way of thinking: Discriminant $D = b^2 - 4ac$

IF denoted as $A = \dfrac{-b}{2a}$, $B = \dfrac{\sqrt{|D|}}{2a}$

Real roots; $A \pm B$

Imaginary roots; $A \pm Bi$

In this example, the discrimination between the real and imaginary roots is made by printing 111 in the case of the real root and 999 in the case of the imaginary root before printing out the answers.

Unconditional jump instructions

GOTO: (Key operation; [GOTO])

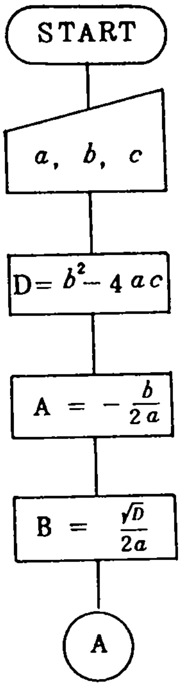
Program jumps to FLG $n\ n$ unconditionally, at this instruction.

(Symbol system)

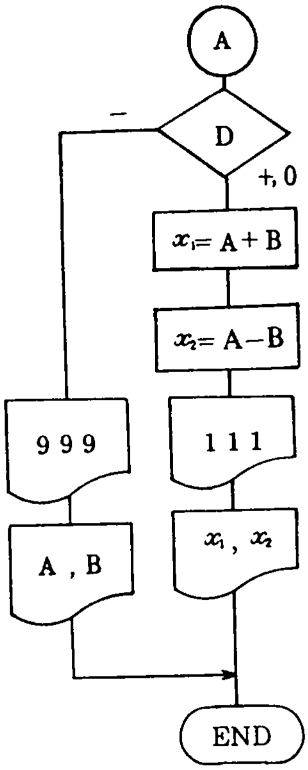
BRANCH: (Key operation: [INST] [I/O] [F])

The number immediately preceding this instruction is interpreted as a step number and unconditional jump to this step takes place. Since the step number and the page number is specified by a 4-digit number, a jump to another page is possible.

(Absolute address system.)



STEP	Instruction	Remark		Key operation	
000	SP			SP	
1	22		Input coefficient a.		2 2
2	SM			SM	
3	11	a			1 1
4	E		Input coefficient b.	ENT	
5	SM			SM	
6	12	b			1 2
7	E		Input coefficient c.	ENT	
8	SM			SM	
9	13	c			1 3
010	RM			RM	
1	12	b			1 2
2	a^2	b^2		a^2	
3	−			−	
4	4			4	
5	×			×	
6	RM			RM	
7	11	a			1 1
8	×			×	
9	RM			RM	
020	13	c			1 3
1	=			=	
2	SM			SM	
3	15	D	$D = b^2 - 4ac$		1 5
4	RM			RM	
5	12	b			1 2
6	÷			÷	
7	2			2	
8	÷			÷	
9	RM			RM	
030	11	a			1 1
1	=			=	
2	SC			SIGNCHG	
3	SM			SM	
4	16	A	$A = -\frac{b}{2a}$		1 6
5	RM			RM	
6	15	D			1 5
7	$\lvert a \rvert$			INST	8 6
8	√			√	
9	÷			÷	
040	2			2	
1	÷			÷	
2	RM			RM	
3	11	a			1 1
4	=			=	
5	SM			SM	
6	17	B	$B = \frac{\sqrt{\lvert D \rvert}}{2a}$		1 7
7	LF			I/O	0
8	LF			I/O	0
9	RM			RM	
050	15	D			1 5
1	IF <0		If discriminant D is	IFGOTO	−
2	03		negative, it jumps.		0 3
3	RM			RM	



Operation procedure

1. LEARN
2. C C ALL
3. Input the program.
4. OPE
5. C
6. *a* S

   b S

   c S

In the case of the real root

111 ◇

x_1 ◇

x_2 ◇

In the case of the imaginary root

999 ◇

Real part A ◇

Imaginary part B ◇

Repeat operation 6.

$$\left(\begin{array}{l} a = 2 \quad b = 7 \quad c = 3 \\ x_1 = 3.000 \quad x_2 = 0.5000 \end{array} \right)$$

STEP	Instruction	Remark		Key operation	
4	16	A			1 6
5	+			+	
6	RM			RM	
7	17	B			1 7
8	=			=	
9	SM			SM	
060	18	x_1			1 8
1	RM			RM	
2	16	A			1 6
3	−			−	
4	RM			RM	
5	17	B			1 7
6	=			=	
7	SM			SM	
8	19	x_2			1 9
9	1	}	Indication of real root.	1	
070	1			1	
1	1			1	
2	◇			◇	
3	RM			RM	
4	18	x_1			1 8
5	5/4			FIX	5
6	04				0 4
7	◇			◇	
8	RM			RM	
9	19	x_2			1 9
080	5/4			FIX	5
1	04				0 4
2	◇			◇	
3	GoTo			GOTO	
4	02				0 2
5	FLG			FLAG	
6	03				0 3
7	9	}	Indication of imaginary root.	9	
8	9			9	
9	9			9	
090	◇			◇	
1	RM			RM	
2	16	A			1 6
3	5/4			FIX	5
4	04				0 4
5	◇			◇	
6	RM			RM	
7	17	B			1 7
8	|a|			INST	8 6
9	5/4			FIX	5
100	04				0 4
1	◇			◇	
2	FLG			FLAG	
3	02				0 2
4	LF			I/O	0
5	EP			EP	
6	22				2 2

Exercise (3)

(Program with branching)

** (31) Write a program to calculate a deduction for life insurance from income tax. The deduction is calculated according to the following standard.

i) Total amount, when the insurance fee to be paid is up to $80.

ii) (The insurance fee to be paid) x 1/2 + $40,
when the insurance fee to be paid exceeds $80 up to $160.

iii) $125, when the insurance fee to be paid exceeds $160.
($80 .... $80, $100 .... $90, $150 .... $115)

** (32) The trunk line telegram fee is 20 cents for up to 10 letters with additional 3 cents per 5 letters over 11 letters. Write a program to find the fees by inputting the number of letters.
(9 letters .... ¢20, 11 leters .... ¢23, 15 letters .... ¢23, 17 letters .... ¢26)

** (33) There is a game of throwing a ball to a place just five meters away.
A prize is calculated on the base of the distance thrown as follows:

1. $200 per meter in the case of less than 5 meters.
2. $2,000 in the case of just 5 meters.
3. $200 is fined per meter over the 5 meters if a ball is thrown to a place more than 5 meters.
   Write a program to calculate a prize or a fine (shown with – sign) by inputting the distance thrown.

   (4.3 m .... $800, 5.0 m .... $2,000, 5.8 m .... –$200)

3. Program with Iterative Routine

This type is a program having a routine where the same procedure is iterated sequentially. This is distinguished from the subroutine, because this uses a conditional jump to repeat the calculation or to be free from the loop, while the subroutine calculation is executed by unconditional jump instruction. The two kinds of this type program are as follows.

3-1 Input Repetition

In statistic calculation, storing of many data is usually done before starting the calculation. In this case, the procedure to input one datum is repeated as many times as the number of data, and for that reason the number of the data should be counted.

3-2 Calculation Repetition

This is the case such that the same calculation procedure is repeated until the calculated value is regarded as same as given value. When the calculated value has reached to that value, the calculation proceeds to the next step.

**** Example 3-1 Mean, standard deviation (Input repetition)**

> Arrange the program to obtain the mean and standard deviation after the input of the data $x_1, x_2, \ldots\ldots x_n$.
>
> Mean : $\bar{x} = \dfrac{x_1 + x_2 + \cdots\cdots + x_n}{n}$
>
> Standard deviation : $\sigma = \sqrt{\dfrac{1}{n}\{\Sigma x^2 - \dfrac{(\Sigma x)^2}{n}\}}$

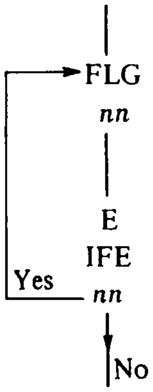
Study Points:

1. Jump instruction (IF ENT)
2. Splitting a memory in two
3. Method of counting the number of data

1. Jump instruction

IFE: (Key operation, [IFGOTO] [ENT])

This is a judging instruction that the calculation procedure changes according as the input has been done or not at the ENT (E) instruction.



The program is written as shown on the left. E instruction is needed just before IFE. When this program proceeds to the E, calculation stops. Then,

1. If the [START] key is depressed after putting numerals, the program jumps to FLG *n n*.
2. If the [START] key is depressed without putting numerals, it proceeds to the next instruction.

2. Splitting memories

Memories can be used by dividing them in two.

In this case, the significant digits of accuracy is decreased to 6 digits, while the number of memories is doubled.

Key operations are shown below.

[RIGHT] [SM] * [1] [3] : Store in right half of memory of address 13.

[LEFT] [RM] * [0] [4] : Call out the left half of memory of address 4.

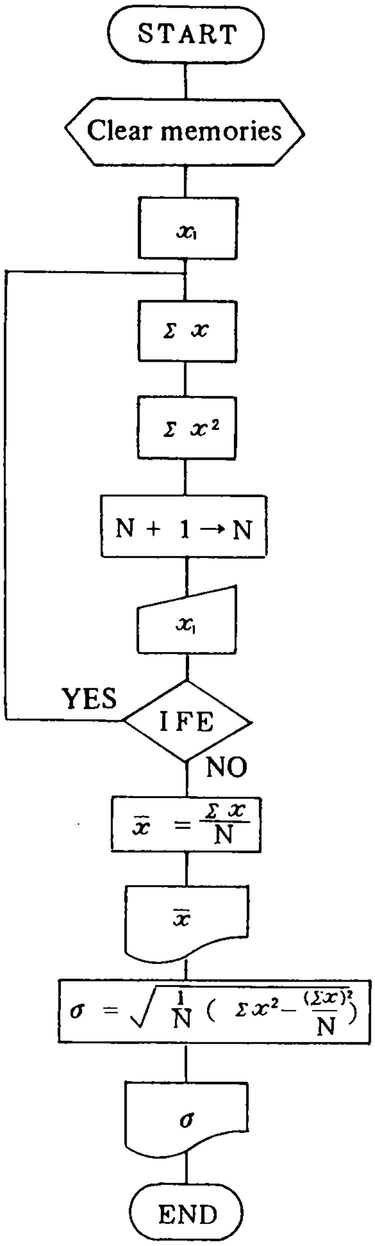
*This system is applied to any of CM, SM, ΣM, RM instructions. In this case, each of them operates on the half of the designated memory.

3. Method of counting the number of data

The number must be often counted in the program besides counting the number of data. In these cases, the following method is used.

```
         |
         R.
         CM    }  A memory to memorize the counted numbers. At first clear this.
         13
   ┌───→ FLG
   │     02
   │     |
   │     1
   │     R
   │     ΣM    }  1 is accumulated and added in the right half of the memory of address 13.
   │     13
   │     E
   │     IFE
   └Yes─ 02
         |No
         R
         RM
         13
         |
```

1 is accumulated and added in the right half of the memory of address 13. Namely, when the calculation passes here once, 1 is added to the right half of the memory of address 13. After several passes, when the right half of address 13 is called out, it shows the number of times that the program has passed this part, namely the number of data in this example.



STEP	Instruction	Remark		Key operation	
000	SP			SP	
1	31				3 1
2	CMA		Clear all memories	INST	F 1
3	E		Input x_i	ENT	
4	FLG			FLAG	
5	02				0 2
6	ΣM		Find $x_1 + x_2 + x_3$ ·····	ΣM	
7	11	Σx			1 1
8	a^2			a^2	
9	ΣM		Find $x_1^2 + x_2^2 + x_3^2$·····	ΣM	
010	12	Σx^2			1 2
1	1		Accumulate 1 in the right half of memory address 13 to find N, the number of data.	1	
2	R			R	
3	ΣM			ΣM	
4	13	N	N = N + 1		1 3
5	E		Input x_i	ENT	
6	IFE		If there is input in previous step, program jumps to FLG02 (04 step).	IFGOGO	ENT
7	02				0 2
8	RM			RM	
9	11	Σx			1 1
020	÷			÷	
1	R			R	
2	RM			RM	
3	13	N			1 3
4	=		$\bar{x} = \Sigma x / N$	=	
5	5/4		Round-off to 3 decimal places.	FIX	5
6	03				0 3
7	◇	$\bar{x}$		◇	
8	RM			RM	
9	12	Σx^2			1 2
030	−			−	
1	RM			RM	
2	11	Σx			1 1
3	a^2			a^2	
4	÷			÷	
5	R			R	
6	RM			RM	
7	13	N			1 3
8	=			=	
9	÷			÷	
040	R			R	
1	RM			RM	
2	13	N			1 3
3	=			=	
4	√			√	
5	5/4			FIX	5
6	06				0 6
7	◇	σ		◇	
8	EP			EP	
9	31				3 1

Operation procedure

1. [LEARN]
2. [C] [C ALL]
3. Input the program.
4. [OPE] [C]
5. [S] x_1 [S]
   x_2 [S]
   ⋮
   x_n [S]
   [S]
   $\bar{x}$ ◇
   σ ◇

Repeat operation 6.

(25, 26, 27 $\bar{x}$ = 26.000 σ = 0.816497)

*** Example 3-2 Find one real root of a equation of third degree.

> Find one real root by giving the coefficients a, b, c of the equation of third degree, by Newton method.
>
> $ax^3 + bx^2 + cx + d = 0$ (a ≒ 0)

Study Point: Jump instruction (IF ≒ 0)

Way of thinking: Let $f(x) = x^3 + \frac{b}{a}x^2 + \frac{c}{a}x + \frac{b}{a} = x^3 + px^2 + qx + r$

$f'(x) = 3x^2 + 2px + q$

According to Newton method,

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{x_i^3 + px_i^2 + qx_i + r}{3x_i^2 + 2px_i + q}$$

When $|x_{i+1} - x_i| \leqq 10^{-6}$, x_{i+1} is regarded as same as x_i to complete the repetition calculation and x_i becomes the answer.

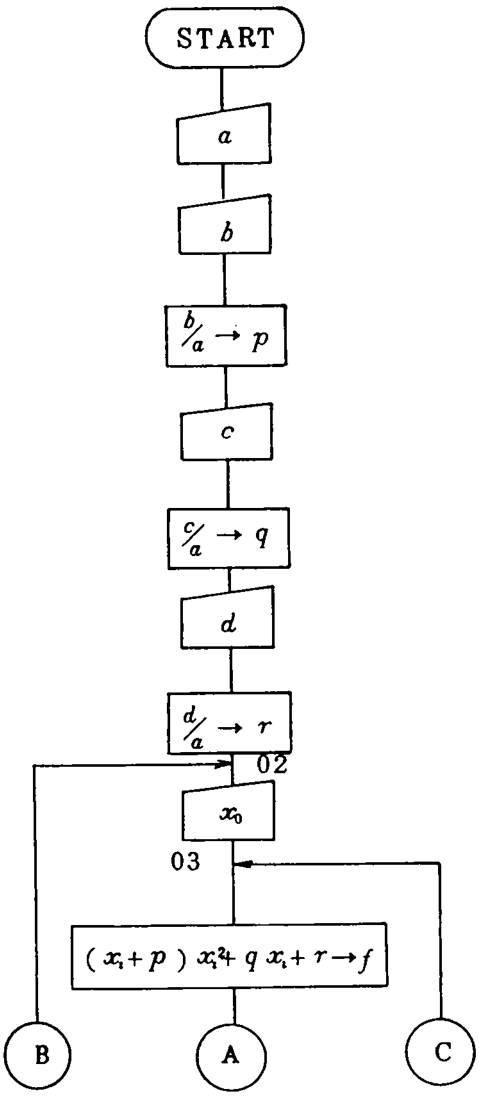
Here, if $f'(x_i) = 0$ and $f(x_i) = 0$, x_i becomes the answer.

If $f'(x_i) = 0$ and $f(x_i) \fallingdotseq 0$, different initial value should be input.

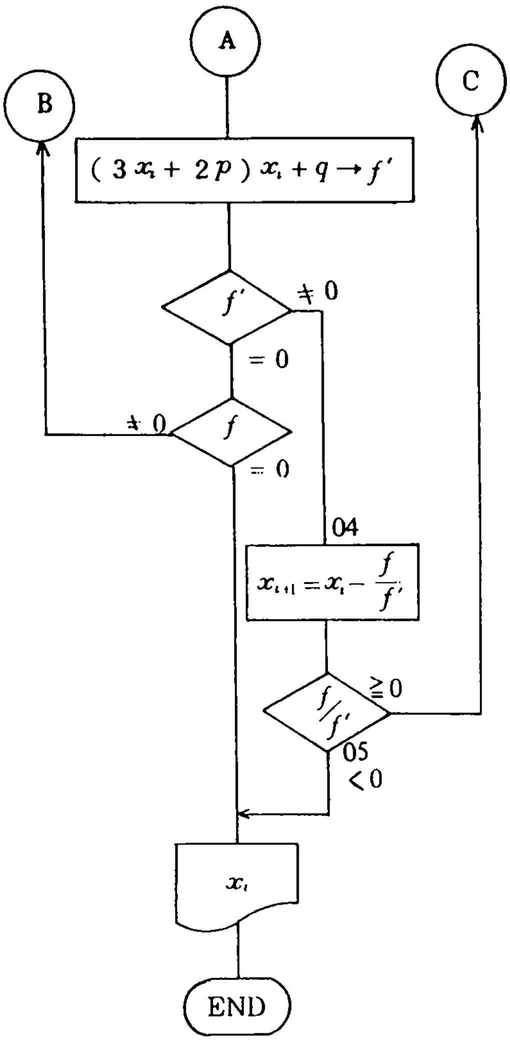
Jump instruction

IF ≒ 0: (Key operation; [IFGOTO] [=])

Program jumps to FLG nn if the contents of the buffer register is not zero. Continues if it is zero. Refer to the Table for comparison of two numbers.



STEP	Instruction	Remark		Key operation	
000	SP			SP	
1	32		Input coefficient a.		3 2
2	SM			SM	
3	17	a			1 7
4	E		Input coefficient b.	ENT	
5	÷			÷	
6	RM			RM	
7	17	a			1 7
8	=			=	
9	SM			SM	
010	19	p	$= b/a$		1 9
1	E		Input coefficient c.	ENT	
2	÷			÷	
3	RM			RM	
4	17	a			1 7
5	=			=	
6	SM			SM	
7	18	q	$= c/a$		1 8
8	E		Input coefficient d.	ENT	
9	÷			÷	
020	RM			RM	
1	17	a			1 7
2	=			=	
3	SM			SM	
4	17	r	$= d/a$		1 7
5	LF			I/O	0
6	FLG			FLAG	
7	02				0 2
8	E		Input initial value x_0.	ENT	
9	SM			SM	
030	16	x_0			1 6
1	FLG			FLAG	
2	03				0 3
3	RM			RM	
4	16	x_i			1 6
5	+			+	
6	RM			RM	
7	19	p			1 9
8	=			=	
9	×			×	
040	RM			RM	
1	16	x_i			1 6
2	a^2			a^2	
3	+			+	
4	RM			RM	
5	18	q			1 8
6	×			×	
7	RM			RM	
8	16	x_i			1 6
9	+			+	



($a = 1,\ b = 0,\ c = 0,\ d = -1,$
$1,\ 0.5 \pm 0.866i$)

STEP	Instruction	Remark		Key operation	
050	RM			RM	
1	17	r			1 7
2	=			=	
3	SM	f		SM	
4	15				1 5
5	3			3	
6	×			×	
7	RM			RM	
8	16	x_i			1 6
9	+			+	
060	2			2	
1	×			×	
2	RM			RM	
3	19	p			1 9
4	=			=	
5	×			×	
6	RM			RM	
7	16	x_i			1 6
8	+			+	
9	RM			RM	
070	18	q			1 8
1	=			=	
2	SM			SM	
3	14	f'			1 4
4	IF≠0			IFGOTO	=
5	04				0 4
6	RM			RM	
7	15	f			1 5
8	IF≠0			IFGOTO	=
9	02				0 2
080	GoTo			GOTO	
1	05				0 5
2	FLG			FLAG	
3	04				0 4
4	RM			RM	
5	15	f			1 5
6	÷			÷	
7	RM			RM	
8	14	f'			1 4
9	=			=	
090	SM		$f/f' = x_i - x_{i+1}$	SM	
1	11				1 1
2	SC			SIGNCHG	
3	ΣM		x_{i+1}	ΣM	
4	16				1 6
5	|a|			INST	8 6
6	−			−	
7	1			1	
8	EXP		1×10^{-6}	EXP	
9	6			6	

Operation procedure

1. [LEARN]
2. [C] [C ALL]
3. Input the program
4. [OPE]
5. [C]
6. a [S]

   b [S]

   c [S]

   d [S]

Initial value x_0 [S]

Real Root ◇

Repeat operation 6.

Input another initial value, when the real root is not printed even if x_0 is input and calculation stops in ENTRY status.

STEP	Instruction	Remark		Key operation	
100	SC			SIGN CHG	
1	=			=	
2	IF≧0			IFGOTO	+
3	03				0 3
4	FLG			FLAG	
5	05				0 5
6	RM			RM	
7	16	x_1	Print the answer		1 6
8	5/4			FIX	5
9	03				0 3
110	◇			◇	
1	LF			I/O	0
2	EP			EP	
3	32				3 2
4					
5					
6					
7					
8					
9					
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					

Exercise (4)

(Repetition program)

* (41) When results x_1 x_2 ..... x_n of one subject is given and its average $\bar{x}$ and standard deviation σ are known, write a program to find deviation T_i of x_1 x_2 ..... x_n.

In this case, deviation is represented by the following equation.

$$T_i = \frac{x_i - \bar{x}}{\sigma} \times 10 + 50$$

($\bar{x} = 60, \sigma = 20, x_1 = 80, Ti = 60{,}000$)

* (42) Input $x_1, f_1, x_2, f_2, \ldots\ldots x_n, f_n$ in this order and write a program to find x by the formula shown below. (Average of data with frequency distribution)

$$\bar{x} = \frac{x_1 \, f_1 + x_2 \, f_2 + \cdots\cdots + x_n \, f_n}{f_1 + f_2 + \cdots\cdots\cdots\cdots + f_n}$$

($x_1 = 5, f_1 = 2, x_2 = 7, f_2 = 3, \bar{x} = 6{,}200$)

** (43) The following equation indicates the relation between average temperature $t^\circ C$ from 2 to 3 o'clock p.m. and the number Y of ice creams sold during that hour. Write a program to find the number of sales in succession at average temperatures, 21°, 22°, ... 30°, using the following formula. (This is a hypothetical equation.)

$$Y = \frac{1}{10}(t^3 - 8t^2 + 15t)$$

(605, 711, 828, 958, 1100, 1256, 1426, 1610, 1810, 2025)

** (44) Write a program to find the three-month moving average in the monthly data x_1, x_2 .....

$$\bar{x}_1 = \frac{x_1 + x_2 + x_3}{3}$$

$$\bar{x}_2 = \frac{x_2 + x_3 + x_4}{3}$$

(1, 2, 3, 4 ...., 2.000, 3.000)

***(45) Write a program to find the combination ${}_nC_r$ with given n, r and using the following formula.

(Program Classification No. X1-1075)

(If $n!$ function is used, overflow comes in n > 70. Therefore study the following formula.)

$${}_nC_r = \frac{n!}{(n-r)!\,r!} = \frac{n(n-1)(n-2)\cdots\cdots(n-r+1)}{r(r-1)(r-2)\cdots\cdots\cdots\cdots 1}$$

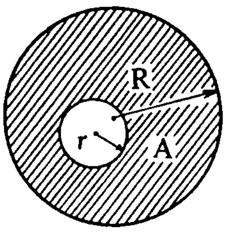
However, r represents the smaller value between $n-r$ and r.

(${}_6C_4 = 15$, ${}_7C_3 = 35$)

4. Program with Subroutines

When the same calculation is used several times here and there, you had better arrange this part as a subroutine program.

Example 4.



Arrange the program to obtain the internal area A bounded by two circles as shown in the figure. The radii R (of larger circle) and r (of smaller one) are given.

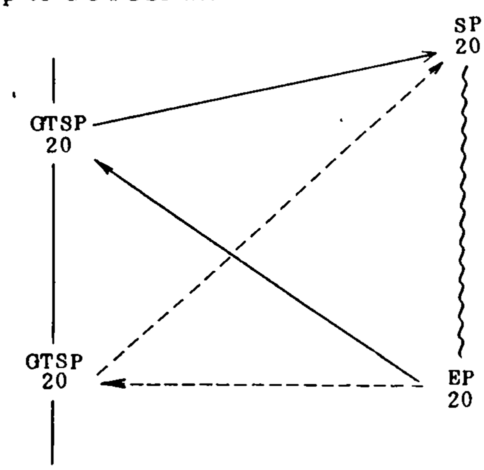
Study Point; Subroutine

Subroutine

As the area of a circle must be calculated twice in this example, a program of that calculation part is written as a subroutine.

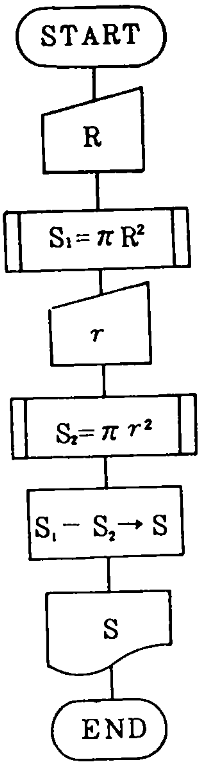
1 Subroutine program start with SP*nn* and ends with EP*nn* as a main program.

2 In a main program, GOTOSP*nn* instruction is put at the step to be jumped to the subroutine. When the program comes as far as this instruction, jumps automatically to the subroutine SP*nn* and returns to the next step to GOTOSP*nn*.



3 Subroutine nesting is up to two depths.

4 Special consideration (given in P. 21) is required for jumping to a subroutine on another page.



Operation procedure

1. [LEARN]
2. [C] [C ALL]
3. Input the program.
4. [OPE]
5. [C]
6. R [S]

   r [S]

   S ◇

Repeat operation 6.

(R = 4 r = 2 S = 37.6991)

STEP	Instruction	Remark		Key operation	
000	SP		Input R	SP	
1	41				4 1
2	GTSP		Jump to subroutine	GoToSP	
3	10		(SP10)		1 0
4	SM			SM	
5	10	S_1	$= \pi R^2$		1 0
6	E		Input r	ENT	
7	GTSP		Jump to subroutine	GoToSP	
8	10		(SP10)		1 0
9	SM			SM	
010	11	S_2	$= \pi r^2$		1 1
1	RM			RM	
2	10	S_1			1 0
3	−			−	
4	RM			RM	
5	11	S_2			1 1
6	=			=	
7	5/4			FIX	5
8	04				0 4
9	◇			◇	
020	LF			I/O	0
1	EP			EP	
2	41				4 1
3	SP			SP	
4	10				1 0
5	a^2			a^2	
6	×			×	
7	π			arc	arc
8	=			=	
9	EP			EP	
030	10			10	
1					
2					
3					
4					
5					
6					
7					
8					
9					
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					

5. Array

When many data are dealt with as one group by arranging them, this is called an array. There are programs in which the data recalled by designating the index, and exchanged or processed. In the following example, data are input one after another, and after completion of input, they are printed out one after another after the processing of each datum.

*** Example 5

There are some data x_1, x_2 ..... x_n. Write a program to find mean $\bar{x}$ and then the difference between the mean and each datum. Make the print out so that the individual data and difference from the mean can be compared at a glance. (The number of data is within the memory capacity−3.)

Study Points: 1. INDIRECT instruction

2. Method of making tables

1. INDIRECT instruction

As memory addresses are indirectly designated, this name is given.

Suppose that the number 3 is memorized in the 00 address.

In this case, when instruction is given as [IND] [SM*] [0] [0], this works in the same way as [SM] [0] [3].

This is a convenient method by which some data are memorized one by one, for instance from address 3.

Now, x_1 .... Address 3

x_2 .... Address 4

x_3 .... Address 5, thus data are to be memorized.

In this case, the program is written as follows:

```
 |
 3
SM          For the first time, the contents of the 0 0 address is 3.
00          Therefore they work in the same way as SM03.
E           The second time, the contents of the 0 0 address is 4.
FLG ←┐      Therefore they work in the same way as SM04.
10   │      The third time, the contents of the 0 0 address is 5.
IND  │      Therefore they work in the same way as SM05.
SM   │
00   │
 |   │
1    │
ΣM   │      1 is accumulated whenever one pass is made. The first time, namely, the
00   │      contents of the 00 address change from 3 to 4 (3 + 1 → 4) and then the program
E    │      jumps to FLG 10.
IFE ─┘
10
```

* CM, ΣM, or RM can be used instead of SM. And program is performed according to the instruction at that time.

2. Method of making tables

A simple table can be made since thermal paper is very wide (for 48 characters). Alphabet, numerals and symbols can be printed in the desired format.

There are several instructions for that purpose.

① SPACE *nn*

Print-head spaces by number designated by *nn*. Namely, this is an instruction by which blanks are made by necessary digits.

② CHARACTER PRINT

Place this instruction before and after character (and figure symbol) printing.

See the key layout on p. 5 for the correspondence between keys and characters, figures and symbols.

The characters, figures, and symbols on the keyboard can be printed by manual operation as well as in a program, while the symbols and some characters that are specified only in the Character Code Table but not on the keyboard are printed only in a program. In this case, look for *m* and *n*, the code corresponded to the character, in the Character Code Table (p. 5) and enter it as [INST] [*m*] [*n*] .

③ COL - PRINT *nn*

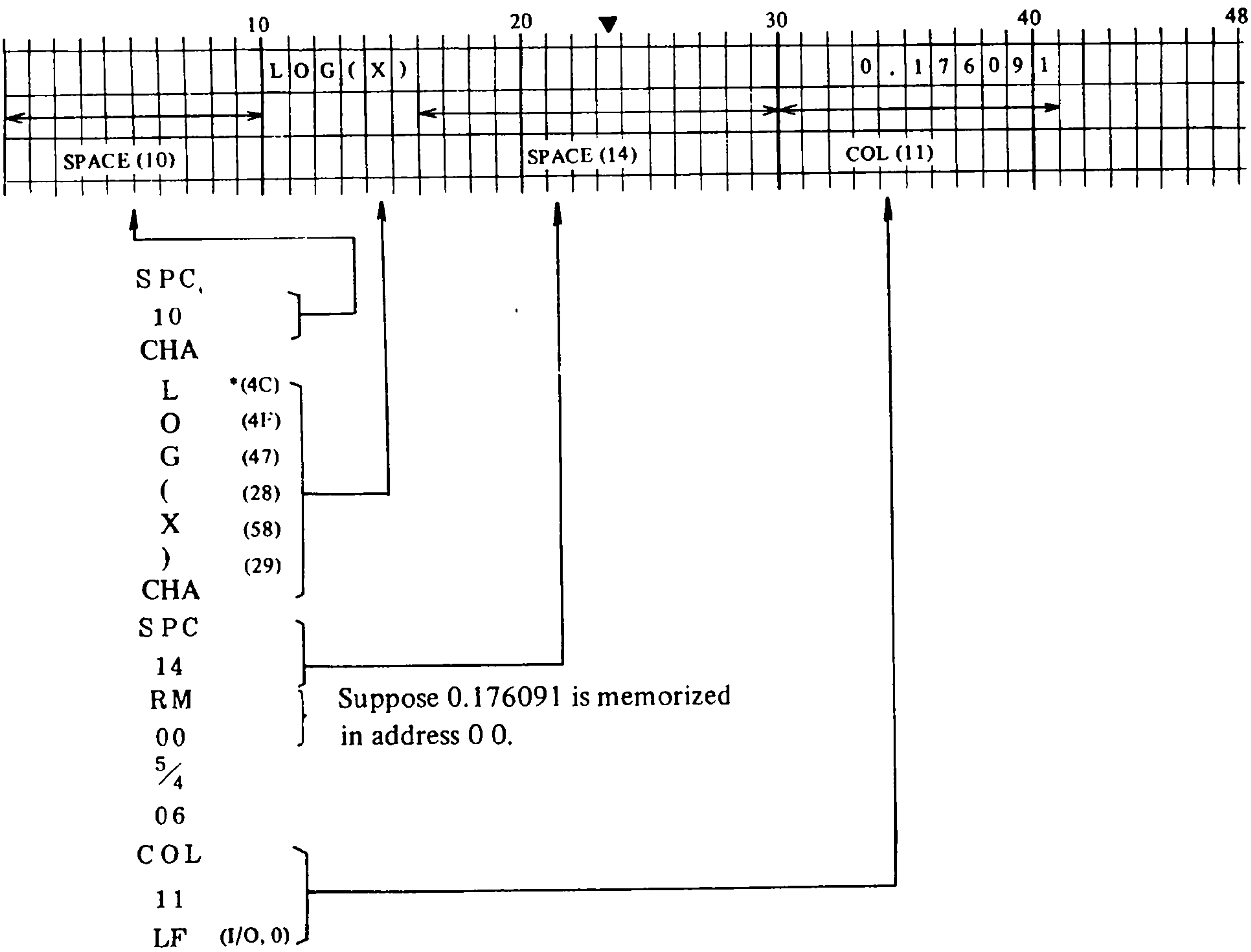
This is an instruction to print the contents of buffer register in digits designated by number *nn*. When the specified column is longer than that of data, spacing is carried out in higher digits to fill the specified column.

| FIX*n* *nn* COL *nn* |	Round instruction must be put just before this instruction, and they are always used in such combination as shown on the left side.

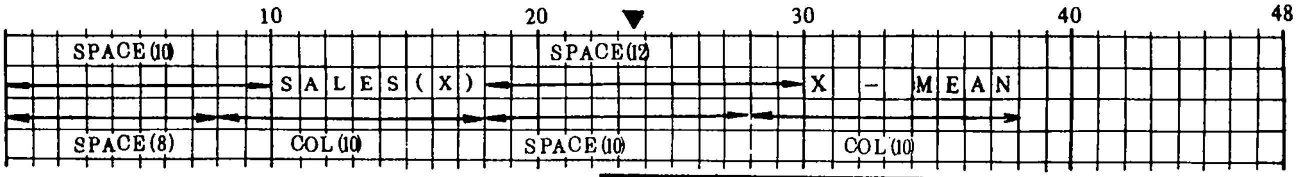
④ LINE FEED

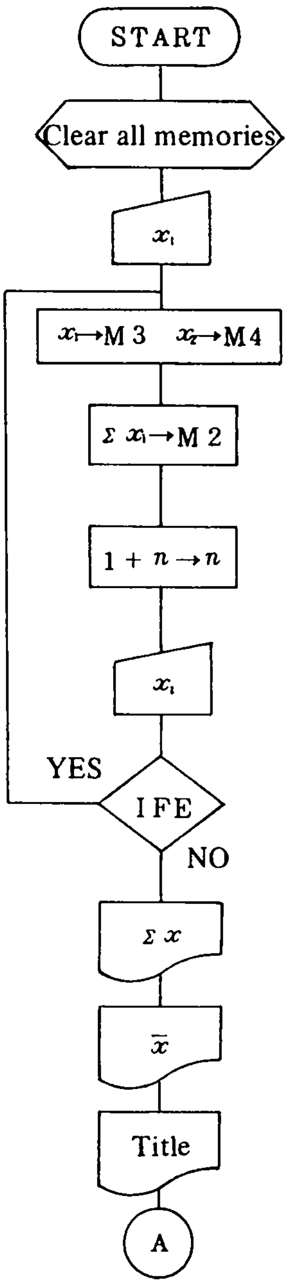
Shift the print-head to the end of the right hand by giving LF instruction (key operation [I/O] [0]) after the last CHARACTER PRINT or COL-PRINT instruction of one line. (No paper feeding in this time)

The following example shows the usage of this three kinds of Print instructions.

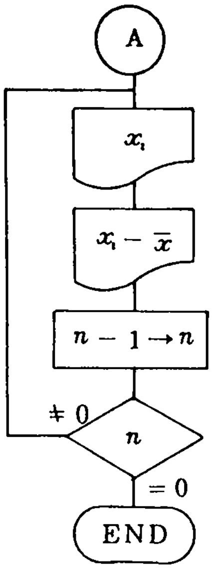


* Depress the corresponding character keys on the keyboard or use the [INST] key as [INST] [n] [n].





STEP	Instruction	Remark		Key operation	
000	SP			SP	
1	51				5 1
2	CMA		Clear all memories	INST	F 1
3	3		Memorize the head	3	
4	SM		address 3 in address 00	SM	
5	00				0 0
6	E	x_1	Input x_1	ENT	
7	FLG			FLAG	
8	10				1 0
9	IND			INDIRECT	
010	SM			SM	
1	00				0 0
2	ΣM			ΣM	
3	02	Σx	$x_1 + x_2 + \cdots$		0 2
4	1			1	
5	ΣM		Count the number of data.	ΣM	
6	01	n	$n + 1 \to n$		0 1
7	ΣM			ΣM	
8	00				0 0
9	E		Input x_i	ENT	
020	IFE		If data is input, pro-	IFGOTO	ENT
1	10		gram jumps to FLG 10.		1 0
2	LF			I/O	0
3	RM			RM	
4	02	Σx			0 2
5	5/4			FIX	5
6	03				0 3
7	◇			◇	
8	÷			÷	
9	RM			RM	
030	01	n			0 1
1	=			=	
2	SM			SM	
3	02	$\bar{x}$			0 2
4	5/4			FIX	5
5	03				0 3
6	◇			◇	
7	LF			I/O	0
8	SPC			SPACE	
9	10				1 0
040	CHA		Print characters and	CHA	
1	S	53	symbols.		S
2	A	41			A
3	L	4C			L
4	E	45			E
5	S	53			S
6	(	28			(
7	X	58			X
8	)	29			)
9	CHA			CHA	



Operation procedure

1. [LEARN]
2. [C]
3. Input the program.
4. [OPE]
5. [C]
6. [S] x_1 [S]
   x_2 [S]
   ⋮
   x_n [S]
   [S]
   Σx ◇
   $\bar{x}$ ◇
   Title ◇
   x_i ◇
   $x_i - \bar{x}$ ◇

Repeat operation 6.

STEP	Instruction	Remark		Key operation	
050	SPC			SPACE	
1	12				1 2
2	CHA			CHA	
3	x	58			X
4	ƀ	20			SPACE
5	–	2D			–
6	ƀ	20			SPACE
7	M	4D			M
8	E	45			E
9	A	41			A
060	N	4E			N
1	CHA			CHA	
2	LF			I/O	0
3	3		Store the head ad-	3	
4	SM		dress 3 again for	SM	
5	00		recalling		0 0
6	FLG			FLG	
7	11				1 1
8	SPC			SPACE	
9	08				0 8
070	IND			IND	
1	RM			RM	
2	00				0 0
3	5/4			FIX	5
4	00				0 0
5	COL	x_i	Print x_i	COL	
6	10				1 0
7	SPC			SPACE	
8	10				1 0
9	IND			IND	
080	RM			RM	
1	00				0 0
2	–			–	
3	RM			RM	
4	02				0 2
5	=			=	
6	5/4			FIX	5
7	00				0 0
8	COL	$x_i - \bar{x}$	Print $x_i - \bar{x}$	COL	
9	10				1 0
090	LF			I/O	0
1	1			1	
2	ΣM			ΣM	
3	00				0 0
4	SC			SC	
5	ΣM			ΣM	
6	01				0 1
7	RM			RM	
8	01				0 1
9	IF≠0		Continue repetition	IFGOTO	–
100	11		until 0.		1 1
1	EP			EP	
2	51				5 1

6. Making Tables

Tables can be made by taking advantage of wide roll paper.

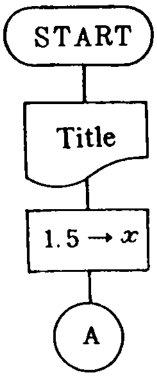
*** Example 6 Common logarithm table

> Write a program to print out x at the 0.5 interval from 1.5 to 9.5 and common logarithm, $log\ x$, correspondingly.
> Print out titles "COMMON LOGARITHM TABLE", "X", and "LOG(X)" as shown below.

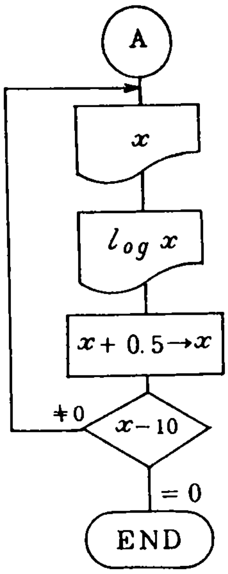
Study Points: 1. Usage of functions

2. Method of making tables

```
        10        20   ▼    30        40      48
          COMMON LOGARITHM TABLE

              X       LOG(X)
             1.5     0.176091
             2.0     0.301030
             2.5     0.397940
             3.0     0.477121
              :         :
              :         :
             9.5     0.977724
```



STEP	Instruction	Remark	Key operation	
000	SP		SP	
1	60			6 0
2	SPC		SPACE	
3	10			1 0
4	CHA		CHA	
5	C	43		C
6	O	4F		O
7	M	4D		M
8	M	4D		M
9	O	4F		O
010	N	4E		N
1	␢	20		SPACE
2	L	4C		L
3	O	4F		O
4	G	47		G
5	A	41		A
6	R	52		R
7	I	49		I
8	T	54		T
9	H	48		H
020	M	4D		M
1	␢	20		SPACE
2	T	54		T
3	A	41		A
4	B	42		B
5	L	4C		L
6	E	45		E
7	CHA		CHA	
8	LF		I/O	0
9	LF		I/O	0
030	SPC		SPACE	
1	14			1 4
2	CHA		CHA	
3	X	58		X
4	CHA		CHA	
5	SPC		SPACE	
6	07			0 7
7	CHA		CHA	
8	L	4C		L
9	O	4F		O
040	G	47		G
1	(	28		(
2	X	58		X
3	)	29		)
4	CHA		CHA	
5	LF		I/O	0
6	1		1	
7	•		•	
8	5		5	
9	SM		SM	



STEP	Instruction	Remark		Key operation	
050	00	x			0 0
1	FLG			FLAG	
2	10				1 0
3	SPC			SPACE	
4	12				1 2
5	RM			RM	
6	00	x			0 0
7	5/4			FIX	5
8	01				0 1
9	COL		Print X	COLPRINT	
060	04				0 4
1	SPC			SPACE	
2	05				0 5
3	RM			RM	
4	00	x			0 0
5	LOG			LOG	
6	5/4			FIX	5
7	06				0 6
8	COL			COLPRINT	
9	08		Print log X		0 8
070	LF			I/O	0
1	•			•	
2	5			5	
3	ΣM			ΣM	
4	00	x	$x+0.5 \rightarrow x$		0 0
5	RM			RM	
6	00				0 0
7	−			−	
8	1			1	
9	0			0	
080	=			=	
1	IF≠0			IFGOTO	=
2	10				1 0
3	EP			EP	
4	60				6 0
5					
6					
7					
8					
9					
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					

Exercise (5)

*** (51) Find the possibility P of Fisher's exact method by the following equation from the contingency table shown below.

			Total
	a	b	A
	d	c	B
Total	D	C	T

$$P = \frac{A!B!C!D!}{T!a!b!c!d!} \quad \cdots\cdots (1)$$

$$= \frac{{}_{b+c}C_b \cdot {}_{a+d}C_a}{{}_{a+b+c+d}C_{a+b}} \quad \cdots\cdots (2)$$

This problem can be calculated by equation (1) using N! instruction when $T \leqq 69$ and easy to write a program. But equation (2) must be utilized when T exceeds 69, when ${}_nC_r$ may be calculated by using the program of Exercise (45) on Page 41 as a Subroutine.

*** (52) There are several sales. Find a percentage of total sales for each sale.
As to a print form, try to make it so that each sale can be clearly compared with its percentage.
In this problem, a program can be easily written without dividing memories but a program requires a higher skill when the memories must be divided for storing more data.

*** (53) Print out 30 values of the trigonometric functions (SIN, COS, TAN) per a certain angle from any designated angle (Degree, Minute, Second Mode).
Print out "INFINITY" when TAN 90°. Take note of using SED, RED, and RE functions to deal with TAN 90°.

Answers to Exercises

(21)

SP
21
SM
11
E
SM
12
E
SM
13
LF
RM
11
+
RM
12
+
RM
13
=
÷
2
=
SM
14
×
(
RM
14
−
RM
11
)
×
(
RM
14
−
RM
12
)
×
(
RM
14
−
RM
13
)
=
√
5/4 (FIX, 5)
3
◇
EP
21

(22)

SP
22
SM
11
E
SM
12
E
SM
13
LF (I/O, 0)
E
SM
14
E
SM
15
E
SM
16
LF (I/O, 0)
RM
11
×
RM
15
−
RM
12
×
RM
14
=
SM
17
RM
13
×
RM
15
−
RM
12
×
RM
16
=
÷
RM
17
=
5/4
06
◇
RM
11
×
RM
16
−
RM
13
×
RM
14
=
÷
RM
17
5/4 (FIX, 5)
06
◇
LF (I/O, 0)
LF (I/O, 0)
EP
22

(23)

SP
23
SM
11
E
SM
12
E
+
1
=
SM
13
RM
12
+
1
=
a^x
RM
13
−
1
=
÷
RM
12
×
RM
11
=
5/4 (FIX, 5)
00
◇
EP
23

(24)

SP
24
×
E
×
E
DEG (a °′″ →)
SIN
÷
2
=
5/4 (FIX, 5)
03
◇
EP
24

(31)

SP
31
SM
10
–
1
6
0
=
IF ≥ 0 (IFGoTo, +)
10
RM
10
–
8
0
=
IF < 0 (IFGoTo, –)
11
RM
10
÷
2
+
4
0
=
GoTo
12
FLG
10
1
2
5
GoTo
12
FLG
11
RM
10
FLG
12
↓ (FIX, 0)
00
◇
LF (I/O, 0)
EP
31

(32)

SP
32
–
1
0
=
SM
10
IF ≧ 0 (IFGoTo, +)
10
2
0
GoTo
12
FLG
10
RM
10
÷
5
=
↑ (FIX, 9)
00
×
3
+
2
0
=
FLG
12
↓ (FIX, 0)
00
◇
LF (I/O, 0)
EP
32

(33)

SP
33
5/4 (FIX, 5)
00
SM
10
–
5
=
IF < 0 (IFGoTo, –)
11
IF ≠ 0 (IFGoTo, =)
13
GoTo
12
FLG
11
RM
10
×
2
0
0
=
GoTo
14
FLG
13
5
–
RM
10
=
×
2
0
0
=
GoTo
14
FLG
12
2
0
0
0
FLG
14
↓ (FIX, 0)
00
◇
LF (I/O, 0)
EP
33

(41)

SP
41
SM
10
E
SM
11
FLG
10
E
–
RM
10
=
÷
RM
11
×
1
0
+
5
0
=
5/4 (FIX, 5)
03
◇
GoTo
10
EP
41

(42)

SP
42
CMA (INST, F, 1)
E
FLG
02
×
E
ΣM
13
=
ΣM
11
E
IFE (IF GoTo, ENT)
02
RM
11
÷
RM
13
=
5/4 (FIX, 5)
03
◇
EP
42

(43)

SP
43
2
1
SM
15
FLG
10
RM
15
a^2
×
RM
15
–
8
×
RM
15
a^2
+
1
5
×
RM
15
=
÷
1
0
=
5/4 (FIX, 5)
00
◇
1
ΣM
15
RM
15
–
3
1
=
IF ≠ 0 (IFGoTo, =)
10
EP
43

(44)

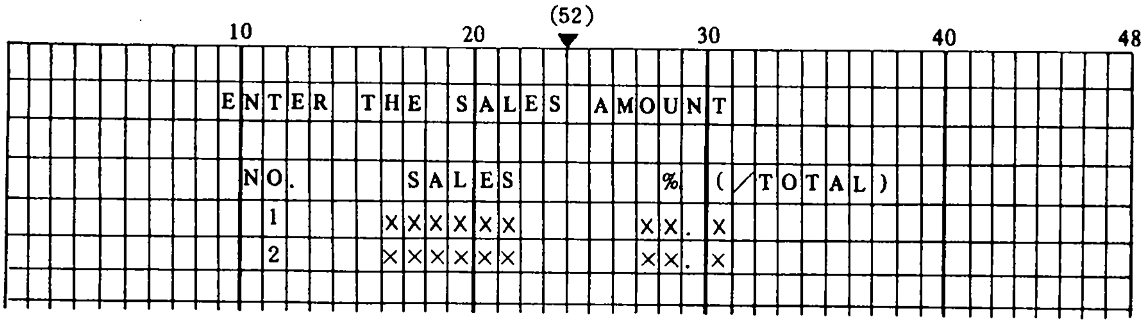
SP
44
SM
11
E
SM
12
FLG
10
E
SM
13
RM
11
+
RM
12
+
RM
13
=
+
3
=
5/4 (FIX, 5)
03
◇
LF (I/O, 0)
RM
12
SM
11
RM
13
SM
12
GoTo
10
EP
44

(45)

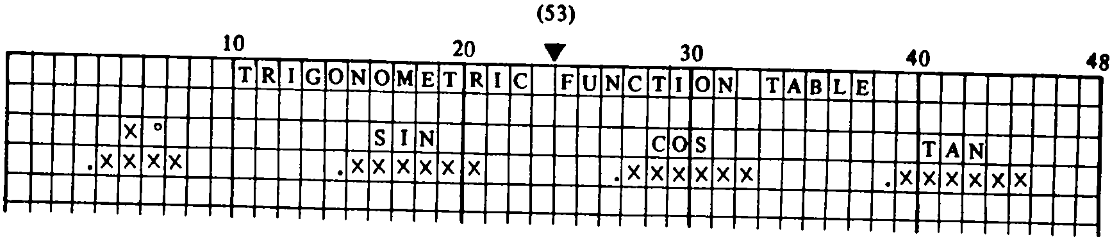
SP
45
SM
17
E
SM
19
LF (I/O, 0)
RM
19
×
2
–
RM
17
=
IF < 0 (IFGoTo, –)
10
SC
ΣM
19
FLG
10
1
SM
18
FLG
12
RM
19
IF ≠ 0 (IFG0To, =)
11
RM
18
5/4 (FIX, 5)
04
◇
GoTo
14
FLG
11
RM
18
×
RM
17
÷
RM
19
=
SM
18
1
SC
ΣM
19
ΣM
17
GoTo
12
FLG
14
EP
45

(51)

```
SP
51
SM
11    a
E
SM
12    b
E
SM
13    c
E
SM
14    d
LF (I/O, 0)
RM
12
ΣM
13
RM
13
SM
17
RM
12
SM
19
GTSP
46
SM
10
RM
11
ΣM
14
RM
14
SM
17
RM
11
SM
19
GTSP
46
×
RM
10

=
SM
10
RM
13
+
RM
14
=
SM
17
RM
11
+
RM
12
=
SM
19
GTSP
46
÷
RM
10
=
1/a
5/4   (FIX, 5)
04
◇
EP
51

SP
46
RM
19
×
2
–
RM
17
=
IF < 0   (IF GoTo, –)
10
SC
ΣM
19
FLG
10
1
SM
18
FLG
12
RM
19
IF ≯ 0   (IF GoTo, =)
11
RM
18
GoTo
14
FLG
11
RM
18
×
RM
17
÷
RM
19
=
SM
18
1
SC
ΣM
19

ΣM
17
GoTo
12
FLG
14
EP
46
```




```
SP
52
SPC
09
CHA
E      45
N      4E
T      54
E      45
R      52
ƀ      20
T      54
H      48
E      45
ƀ      20
S      53
A      41
L      4C
E      45
S      53
ƀ      20
A      41
M      4D
O      4F
U      55
N      4E
T      54
CHA
LF  (I/O, 0)
LF  (I/O, 0)
CM
01
CM
02
4
SM
00
E
FLG
00
IND
L
SM
00
ΣM
02
1
ΣM
01
E
IFE
01
GoTo
02
FLG
01
IND
R
SM
00
ΣM
02
1
ΣM
01
ΣM
00
E
IFE
00
FLG
02
LF  (I/O, 0)
SPC
10
CHA
N      4F
O      4F
.      2E
ƀ      20
ƀ      20
ƀ      20
ƀ      20
S      53
A      41
L      4C
E      45
S      53
CHA
SPACE
06
CHA
%      25
ƀ      20
(      28
/      2F
T      54
O      4F
T      54
A      41
L      4C
)      29
CHA
LF (I/O, 0)
4
SM
00
1
SM
03
FLG
03
↓ (FIX, 0)
00
COL
12
IND
L
RM
00
↓ (FIX, 0)
00
COL
10
÷
RM
02
×
1
0
0
=
5/4 (FIX, 5)
01
COL
09
LF  (I/O, 0)
1
ΣM
03
SC
ΣM
01
RM
01
IF ≠ 0  (IF GoTo, =)
04
GoTo
05
FLG
04
RM
03
↓  (FIX, 0)
00
COL
12
IND
R
RM
00
↓  (FIX, 0)
00
COL
10
÷
RM
02
×
1
0
0
=
5/4  (FIX, 5)
01
COL
09
LF (I/O, 0)
1
ΣM
03
ΣM
00
SC
ΣM
01
RM
01
IF ≠ 0   (IF GoTo, =)
03
FLG
05
LF  (I/O, 0)
EP
52
```



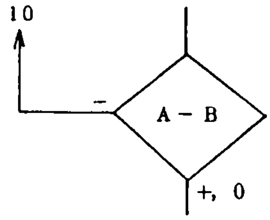
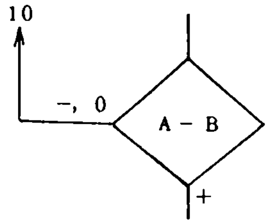
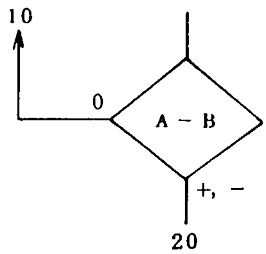
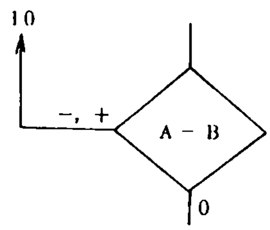
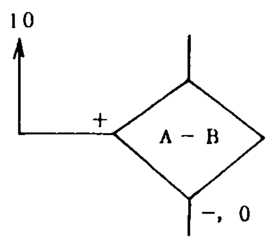
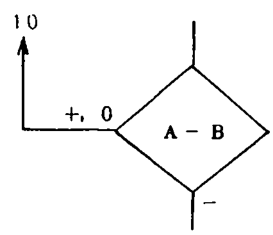
SP
53
SPC
10
CHA
T 54
R 52
I 49
G 47
O 4F
N 4E
O 4F
M 4D
E 45
T 54
R 52
I 49
C 43
b̸ 20
F 46
U 55
N 4E
C 43
T 54
I 49
O 4F
N 4E
b̸ 20
T 54
A 41
B 42
L 4C
E 45
CHA
LF (I/O, 0)
LF (I/O, 0)
SPC
05
CHA
X 58
° DF

CHA
SPC
09
CHA
S 53
I 49
N 4E
CHA
SPC
09
CHA
C 43
O 4F
S 53
CHA
SPC
09
CHA
T 54
A 41
N 4E
CHE
LF (I/O, 0)
E
DEG
SM
00
E
DEG
SM
01
1
0
SM
05
FLG
00
RM
00
DMS (*arc*, $a^{\circ\prime\prime\prime}$ →)
5/4 (FIX, 5)

04
COL
08
RM
00
SIN
5/4 (FIX, 5)
06
COL
13
RM
00
COS
5/4 (FIX, 5)
06
COL
12
SED (INST, F, 9)
RM
00
TAN
IFER (IFGOTO, CE)
01
5/4
06
COL
12
GoTo
02
FLG
01
RE (INST, F, 3)
SPC
03
CHA
I 49
N 4E
F 46
I 49
N 4E
I 49

T 54
Y 59
CHA
FLG
02
LF (I/O, 0)
RED (INST, F, 7)
RM
01
ΣM
00
RM
05
–
1
=
SM
05
IF ≭ 0 (IF GoTo, ≒)
00
LF (I/O, 0)
LF (I/O, 0)
EP
53

Table for Comparison of two numbers

Item	Flow chart	Description by FORTRAN	Program by SX300
Decision <		IF(A LT B)GoTo 10 A<B	RM A − RM B = IF<0 10
Decision ≦		IF(A LE B)GoTo 10 A≦B	RM A − RM B = SC IF≧0 10
Decision =		IF(A EQ B)GoTo 10 A = B	RM A − RM B = IF≠0 20 GoTo 10
Decision ≠		IF(A NE B)GoTo 10 A≠B	RM A − RM B = IF≠0 10
Decision >		IF(A GT B)GoTo 10 A>B	RM A − RM B = SC IF<0 10
Decision ≥		IF(A GE B)GoTo 10 A≧B	RM A − RM B = IF≧0 10

CANON CANOLA SX-300 SERIES
PROGRAMMING MANUAL

Canon

CANON INC. 11-28, Mita 3-chome, Minato-ku, Tokyo 108, Japan

U.S.A. — **CANON U.S.A., INC. HEAD OFFICE**
10 Nevada Drive, Lake Success, Long Island, N.Y. 11040, U.S.A.

CANON U.S.A., INC. CHICAGO OFFICE
457 Fullerton Avenue, Elmhurst, Illinois 60126, U.S.A.

CANON U.S.A., INC. LOS ANGELES OFFICE
123 Paularino Avenue East, Costa Mesa, California 92626, U.S.A.

CANADA — **CANON OPTICS & BUSINESS MACHINES CANADA, LTD.**
HEAD OFFICE
3245 American Drive, Mississauga, Ontario, L4V 1B8, Canada

EUROPE, AFRICA & MIDDLE EAST — **CANON AMSTERDAM N.V.**
Gebouw 70, Schiphol Oost, Holland

CENTRAL & SOUTH AMERICA — **CANON LATIN AMERICA, INC. SALES DEPARTMENT**
P. O. Box 7022, Panama 5, Rep. of Panama

CANON LATIN AMERICA, INC. REPAIR SERVICE CENTER
P. O. Box 2019, Colon Free Zone, Rep. of Panama

SOUTHEAST ASIA — **CANON INC. HONG KONG BRANCH**
5th Floor 2-6, Fui Yiu Kok Street, Tsuen Wan, New Territories, Hong Kong

PUB SME 3004

0475 F0.5

PRINTED IN JAPAN