

# Electronic Calculator



# 1614P

Canon Canola 1614P Instructions

English Edition



## FOREWORD

We are highly gratified that you have selected Canon's CANOLA 1614P Electronic Desk-top Programmable Calculator.

The CANOLA 1614P is the product of the best craftsmanship of Canon Inc., and we feel quite confident that it will meet with your full satisfaction.

The CANOLA 1614P will work more effectively if used with the Printer Unit P-1 which is available on an optional basis.

This instruction manual has been prepared to provide a reference to the basic calculating and programming logic of the 1614P. The easy-to-understand terminology should prove a boon to those operating such a machine for the first time.

The first part explains the basic keyboard operations and functions as an ordinary calculator. The latter part discusses the programming capabilities of the 1614P.

Specific examples of program calculation including jump functions have been used whenever possible to make the text and meaning as clear as possible.

March, 1972

CANON INC.

# CONTENTS

<b>1. Specifications</b> .....	<b>5-16</b>	3-10 Subroutine Function .....	54
1. Specifications .....	6	3-11 Card System .....	59
<b>2. Explanation of Keys</b> .....	<b>8</b>	3-12 Canon Program Card .....	59
2-1 Explanation of Keys .....	8	3-13 Cross Reference of Program Instruction and Code .....	59
2-2 Explanation of Switches .....	11	3-14 Input Data by Card .....	62
2-3 Explanation of Lamps .....	11	3-15 Correction & Clearing of Program Instructions Written into Cards .....	64
2-4 Explanation of Program Keys .....	11		
2-5 Explanation of Program Switches .....	12		
2-6 Explanation of Program Lamp .....	14		
<b>2. Explanation as an Ordinary Calculator</b> .....	<b>17-34</b>	<b>4. Concept &amp; Program Preparation Methods With Actual Examples</b> .....	<b>65-80</b>
1. General .....	18	1. General .....	66
2. Calculation Examples .....	19	2. Programming .....	67
2-1 Addition and Subtraction .....	19	3. Program Using ROUND Instruction (Calculation with Pythagorean theorem) ....	70
2-2 Multiplication .....	20	4. Program Using Decision Functions (Jump Functions) .....	71
2-3 Division .....	22	4-1 Program Using SENSE JUMP (Depreciation calculation) .....	71
2-4 Mixed Calculation .....	22	4-2 Program Using ENTRY JUMP (Calculation of Average Unit Price) .....	75
2-5 Constant Multiplication and Division .....	23	4-3 Program Using MINUS JUMP (Calculation of a Factorial) .....	78
2-6 Power Calculation .....	24	5. Program Using Subroutine (Calculation of the Velocity of Flowing Fluid) .....	79
2-6-1 Involution .....	24		
2-6-2 Square Root Calculation .....	24		
2-7 Calculations Using Memory (Part 1) .....	25		
2-8 Calculations Using Memory (Part 2) .....	26		
2-9 Supplement to Memory .....	27		
2-10 Applied Calculations .....	28		
<b>3. Explanation as a Programmable Calculator</b> .....	<b>35-64</b>		
1. General .....	36		
2. Before Program Operation .....	37		
2-1 Explanation of Terms .....	37		
2-2 Explanation of Symbols Used .....	37		
3. For Program Operation .....	38		
3-1 Introduction .....	38		
3-2 Manual System .....	38		
3-3 Clearing of Stored Program .....	38		
3-4 Method of Checking and Correcting Program Contents .....	40		
3-5 Incorporation of Constant in Program ....	43		
3-6 Program Step Settings .....	44		
3-7 Interpolated Calculation .....	45		
3-8 ROUND Instructions and Printer Instructions .....	46		
3-9 Decision Functions (Jump Functions) ....	48		

# 1

## **SPECIFICATIONS**

### **1. Specifications**

#### **2. Explanation of Keys**

2-1 Explanation of Keys

2-2 Explanation of Switches

2-3 Explanation of Lamps

2-4 Explanation of Program Keys

2-5 Explanation of Program Switches

2-6 Explanation of Program Lamp

# 1. SPECIFICATIONS

1. **Type** : Electronic Desk-top Programmable Calculator
2. **Keys** : 10 Key system (magnet reed switch)
3. **No. of display digits** : 16 digits
4. **Registers** : 3 operation registers ; 14 memory registers
5. **Operation capacity** :

	Operation	Result
Addition, Subtraction,	16 digits $\pm 16$	16
Multiplication, Division	16 digits $\times \div 16$	16
Square root	16 digits	16

6. **Decimal point system** : During entry & operation : complete floating system  
Result of operation : Can be selected between Preselected and Floating systems.
7. **Decimal point positioning** : Preselection system : Can be preselected at respective digits of 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12.  
Floating system : No specific positioning when in F position.
8. **Round-off function** : The decimal digit following 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12 digits can be rounded-up, -off or -down.  
ROUND instruction in the program provides for round-up, round-off, and round-down.

9. **Negative value** : True value is indicated together with a negative sign.

## 10. Specifications related to programming :

- 1) Program step : 240 steps ; can be divided into 120 and 120 steps. If a card system, expansion may be possible by means of an endless program step system.
- 2) Memory : 14 memories for accumulation ; No. 1 and No. 2 are for accumulation. No. 3 through No. 14 memories can be selectively used for accumulation or temporary storage.

- 3) Jump function  
Conditional jump : 3 kinds ; Unconditional jump : 1 kind  
Subroutine jump : 1 kind
- 4) Program input  
Card & manual combination systems.
- 5) Program check  
Program step, program instruction code simultaneous display system.
- 6) Program correction  
General or partial correction can be accomplished.
- 7) Operation check  
Step-by-step verification of numerical data procedure of program calculation.

## 11. Special Functions :

- 1) Indication function : Zero-suppressed display, automatic punctuation at every 3 digits, lamp indication of memory usage, entry stand-by indication during programmed calculation.
- 2) Automatic calculating function **AM** key, Automatic calculation of sums of products and quotients ; **X** key, Constant multiplication and division.
- 3) Safety function : Overflow indication and interlock, electronic double entry indication and interlock, automatic clear when power-ON.

## 12. Size, Weight, others :

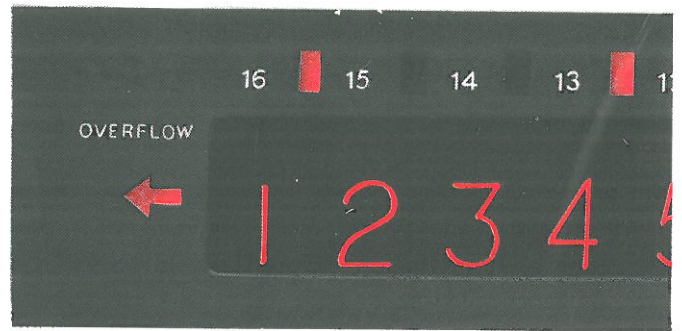
- 1) Size 310W x 395D x 153H (mm)  
(1'3/16" x 1'3-9/16" x 6")
- 2) Weight 8.2 kg (18 lbs.)
- 3) Major operational element : 10 LSIs
- 4) Power consumption  
AC100/115/220/240 V (-15% ~ +10%),  
20 W
- 5) Operational temperature 0 ~ +40°C  
(32°F ~ 104°F)
- 6) Storage temperature -40 ~ +70°C  
(-40°F ~ 158°F)

Subject to alterations.







## 2. EXPLANATION OF KEYS



### 2-1 Explanation of Keys




#### **9** Numeric keys

Entry is accomplished as random keys are depressed one after another. For example, if numeric keys    are depressed in sequence, the number 2 3 4 entered. If two or more numeric keys are depressed at the same time accidentally, the electronic double entry preventive mechanism is actuated, lighting the Overflow Lamp and preventing next entry. The  key corrects this condition.

#### **Decimal Point Key**


Complete automatic decimal point system. Depress the decimal point key during entry at the decimal point position in the value. When entering decimal digits only, depress the  key at the beginning. It is not necessary to depress the  key first.

#### **Plus Equal Key (also +key for addition)**

Depress to obtain results of addition, division, multiplication, sum of products and quotients. While the  key is locked, the result of division and multiplication is automatically added to Memory 1.

#### **Minus Equal Key (also -key for subtraction)**

Depress to obtain results of subtraction, difference of products and quotients.

Also depress it when entering negative numbers. When the  key is locked, the result of division and multiplication is automatically subtracted from Memory 1.


#### **Multiplication Key**

Depress in a multiplication after a multiplicand is entered.

#### **Division Key**

Depress in a division after a dividend is entered.

#### **Square Root Key**

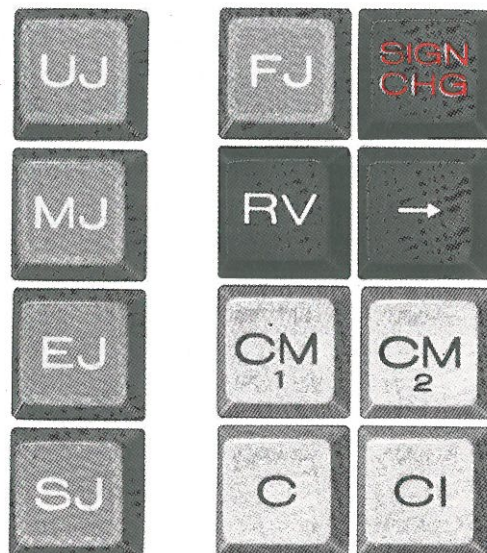
Used for finding square roots. When this key is depressed the indicated value is square rooted. The result of the square root operation is indicated by a positive value regardless of positive or negative value. It is not necessary to depress the  key after this key has been depressed.

**Canola 1614P**

M1 M2 ENTRY


OFF ON

LRN OPE PRO OPE  
CHE CHE



#### **Shift Key**

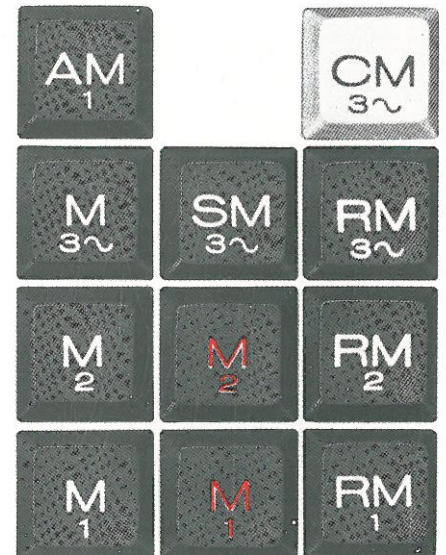
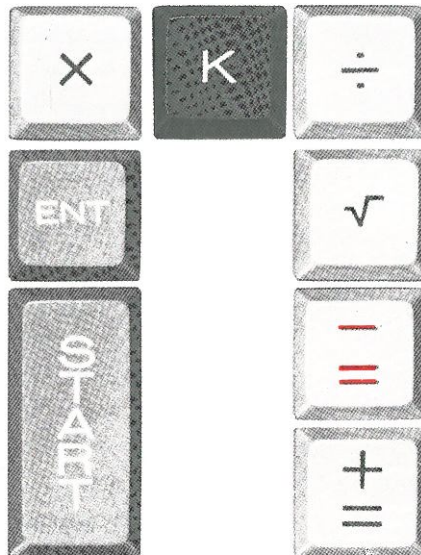
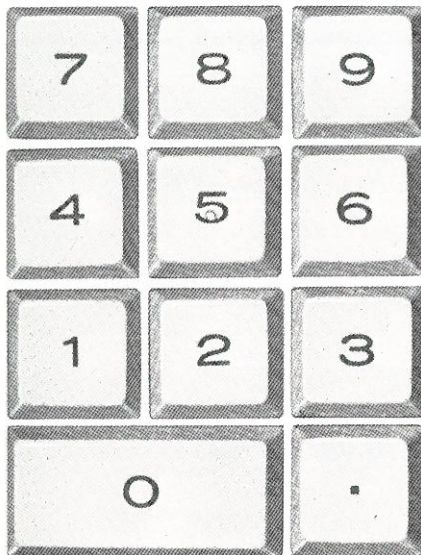
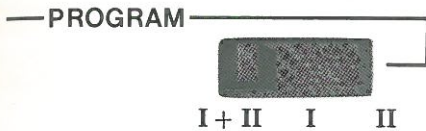
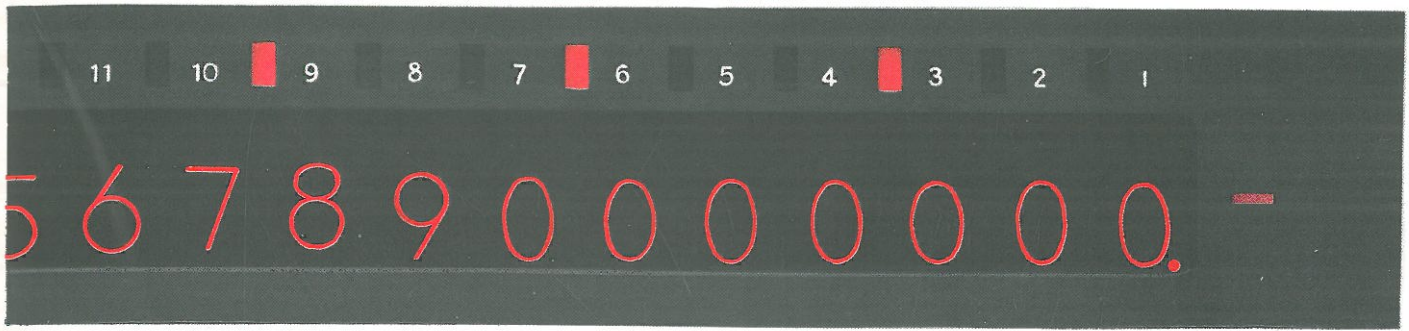
This key is provided to shift the indicated value one digit in the arrow indicated direction, and is mainly used when partially correcting accidentally entered digits.

For example, if 2 3 4 is entered accidentally as 2 3 5, depress the  key once to change the indication to 2 3 then enter 4 to indicate 2 3 4. Similar action will also reduce a digit in the result of manual calculation.

#### **Reverse Key**

This key is used to interchange the multiplicand and multiplier in multiplications and the dividend and divisor in divisions.

This key can also be used for checking and recalling a multiplicand, a divisor, an addend, or a subtrahend in multiplication, division, addition or subtraction



Note: It should be noted that the operation such as  $0 \times n \pm RV$  recalls  $n$  instead of 0. Care should be taken not to program RV instruction, just after  $\pm$  or  $=$  instruction, to recall  $a$  of  $a \times b$  where there is any possibility of  $a$  becoming a zero.

**C Clear Key**

Used to clear all the calculating registers. Provided, however, that the memory is not cleared. When the program is in LEARN MODE (Program Mode Switch in LRN) the contents of a program are cleared in accordance with the instruction of the Program Selector Switch

When the program is in OPERATION MODE (Program Mode Switch is set in OPE), the program is returned to zero step. When the program is in PROGRAM CHECK MODE

(Program Mode Switch is set in  $\text{PRO}_{\text{CHE}}$ ), the program is returned to either 1st or 121st step according to the Program Selector Switch. When the program is in OPERATION CHECK MODE (Program Mode Switch is set in  $\text{OPE}_{\text{CHE}}$ ), the program is returned to zero step.

**CI Clear Indicator Key**

Clears the indicated figures. Used for correcting all entered digits.

**AM<sub>1</sub> Accumulative Memory 1 Key**

For automatic accumulation. Used in sums and differences of products and quotients. By depressing the  $\pm$  key or  $=$  key, the displayed data is added to or subtracted from Memory 1. This is a lock key for Memory 1. (Not used in programming.)

**M<sub>1</sub> Memory 1 Key**

Used to add displayed data to Memory 1.

**M<sub>1</sub>** **Minus Memory 1 Key**  
Used to subtract displayed data from Memory 1

**RM<sub>1</sub>** **Recall Memory 1 Key**  
Used for recalling the contents of Memory 1 to the indicator.

**CM<sub>1</sub>** **Clear Memory 1 Key**  
Used to clear the contents of Memory 1.

**M<sub>2</sub>** **Memory 2 Key**  
Used to add displayed data to Memory 2.

**M<sub>2</sub>** **Minus Memory 2 Key**  
Used to subtract displayed data from Memory 2.

**RM<sub>2</sub>** **Recall Memory 2 Key**  
Used for recalling the contents of Memory 2 to the indicator.

**CM<sub>2</sub>** **Clear Memory 2 Key**  
Used to clear the contents of Memory 2.

**M<sub>3~9</sub>** **Memory 3-9 Key**  
Used to add the contents of the indicator to the contents of 3-9 memories. This key is used in combination with the 3-9 numeric keys. For example, to add 2 to Memory 5, operate keys **2**, **M** and **5** in this sequence.

Minus Memories 3-9 keys are not provided but, the **CHG** key is used to subtract the contents of the indicator from the contents of No. 3-9 memories. For example, to subtract 2 from Memory 5, operate keys **2**, **CHG**, **M** and **5** in this sequence. Thus -2 is added to the contents of Memory 5; simply, 2 has been subtracted from Memory 5.

Memories 10 through 14 are accessible only by program card. This also applies to the **SM**, **RM**, **CM** keys.

**SM<sub>3~9</sub>** **Storage Memory 3-9 key**  
Similar to the **M** key, this key is used to read into No.3 through No. 9 memories, but functions are different. That is, if the indicated value is stored in a specific memory by this key, the former contents stored in the subject memory are automatically cleared and only the new value indicated is to be stored. For example, let us assume that 6 is to be stored in

Memory 5. If keys **2**, **SM** and **5** are operated in sequence, the contents of Memory 5 is changed to 2 from 6.

**RM<sub>3~9</sub>** **Recall Memory 3-9 Key**  
Used to recall the contents of No. 3-No.9 memories to the indicator. Like **M** and similar, this key is to be used with the 3-9 numeric keys.

For example, to recall the contents of Memory 5, operate **RM** and **5** in this sequence.

**CM<sub>3~9</sub>** **Clear Memory 3-9 Key**  
Used to clear the contents in No. 3-No9 memories. Like **M**, **SM** and **RM** keys, this key is used in combination with numeric keys 3-9. For example, keys **CM** and **5** are operated in this sequence to clear contents of Memory 5.

**K** **Constant Key**  
This key is for constant multiplication and division. When it is depressed, it locks and function state is attained.

Initial calculation is performed in the normal procedure. Thereafter constant multiplication and division can be made by operating numeric and **±** keys.

In case of a multiplication, the multiplicand is a constant, and in case of a division, the divisor is a constant.

This key is also used in involutions. The  $n$ th power of a figure is obtained by depressing the **±** key  $(n-1)$  times. Thus,  $a \times \pm \pm \pm \pm$  gives  $a^5$ .

**SIGN CHG** **Sign Change Key**  
Used to convert positive and negative signs of the indicated value. Provided that this key is not applicable when the contents of indication are zero. The relationship of sign change is as follows:

(Contents of indication)	(Key operation)	(Contents of indication)
--------------------------	-----------------	--------------------------

a	→	<b>SIGN CHG</b>	→	-a
-a	→	<b>SIGN CHG</b>	→	a
0	→	<b>SIGN CHG</b>	→	0

(Where  $a \neq 0$ )

## 2-2 Explanation of Switches



### Power Switch

Power is on when this switch is depressed to ON ; and power is off when it is depressed to OFF. When power is turned on, all registers, operation registers and memories are automatically cleared.



### Round-off Switch

With the Decimal Point Selector Dial set at any of the positions except F, when an answer is obtained by depressing either  $\pm$  or  $\frac{\square}{\square}$  key, the digit next to the preselected position is rounded-up, -off or -down. Set this switch to the left position ( $\leftarrow$ ) for round-up to the center (5/4) for round-off, and to the right ( $\rightarrow$ ) for round-down.



### Decimal Point Selector Dial

This dial is used to specify the position of the decimal point in the calculation result. 12 positions may be preselected, i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12 and F. If F position is preselected, the calculation result becomes completely floating, where the position of the decimal point is not specified in the operation result.

The ROUND instruction takes the place of the round-off switch and decimal point selector dial during programmed operation. Refer to information on the ROUND instruction for details.

## 2-3 Explanation of Lamps



### Memory 1 Lamp

Lights when values other than zero are stored in Memory 1, indicating that Memory 1 is in use. Goes off automatically when Memory 1 is cleared by the  $\square$  key, indicating that no values are stored in Memory 1.



### Memory 2 Lamp

Lights when values other than zero are stored in Memory 2, indicating that Memory 2 is in use. Goes off automatically when Memory 2 is cleared by the  $\square$  key, indicating that no values are stored in Memory 2.



### Punctuation Indicators

They light up automatically at every 3rd digit for punctuation, in accordance with the setting of Decimal Point Selector Dial.



### Overflow Lamp

Lights in any one of the following cases, causing entry and operation to be stopped.

- 1) When entry exceeds 16 digits.
- 2) When two or more keys are touched simultaneously by mistake.
- 3) When the integer part of the calculation result exceeds 16 digits.
- 4) When the integer part of the value stored in a memory exceeds 16 digits.



### Minus Lamp

Lights when the value is negative, indicating that the indicated value is negative.

## 2-4 Explanation of Program Keys

Program keys are the keys which function primarily in storing programs in manual LRN, for the execution of programs already stored or checking the contents of a program. Most of the keys explained in section 2-1 for ordinary operation are used as program keys and function identically.

There are some keys, however, which are used exclusively for program operation. These are  $\square$ ,  $\square$ ,  $\square$ ,  $\square$ ,  $\square$ ,  $\square$  and  $\square$  keys, and are distinguished by their blue color.

The usage and functions of these program keys are explained later in Chapter II, but a brief explanation is made below.



### Entry Key

This program key is used to store a temporary stop program instruction when programming in manual LRN. During execution of program at this instruction the program stops and the ENTRY lamp lights.

It is used for the following ;

- 1) Entry instruction to put the calculator in stand-by state for entry.
- 2) Temporary stop instruction for reading calculation results.

### 3) Check of program contents

Checking of a particular stored program is also accomplished by selecting I + II, I or II, as explained according to the above diagram.



- (i) When checking program A, set the selector to either I + II or I.
- (ii) When checking program B, set the selector to II.
- (iii) When checking two programs, set the selector at I for A and II for B.

### 4) Clearing contents of program

When clearing a program, the range to be cleared is determined by the changeover among I + II, I and II. Notice the schematic diagram.

- (i) To clear *a* range, steps 1 through 240, set the switch in I + II position.
- (ii) To clear *b* range, steps 1 through 120, set the switch in I position.
- (iii) To clear *c* range, steps 121 through 240, set the switch in II position.  
(See 3 3-3 Clearing Contents of Program.)





the program proceeds without jumping. (SJ)

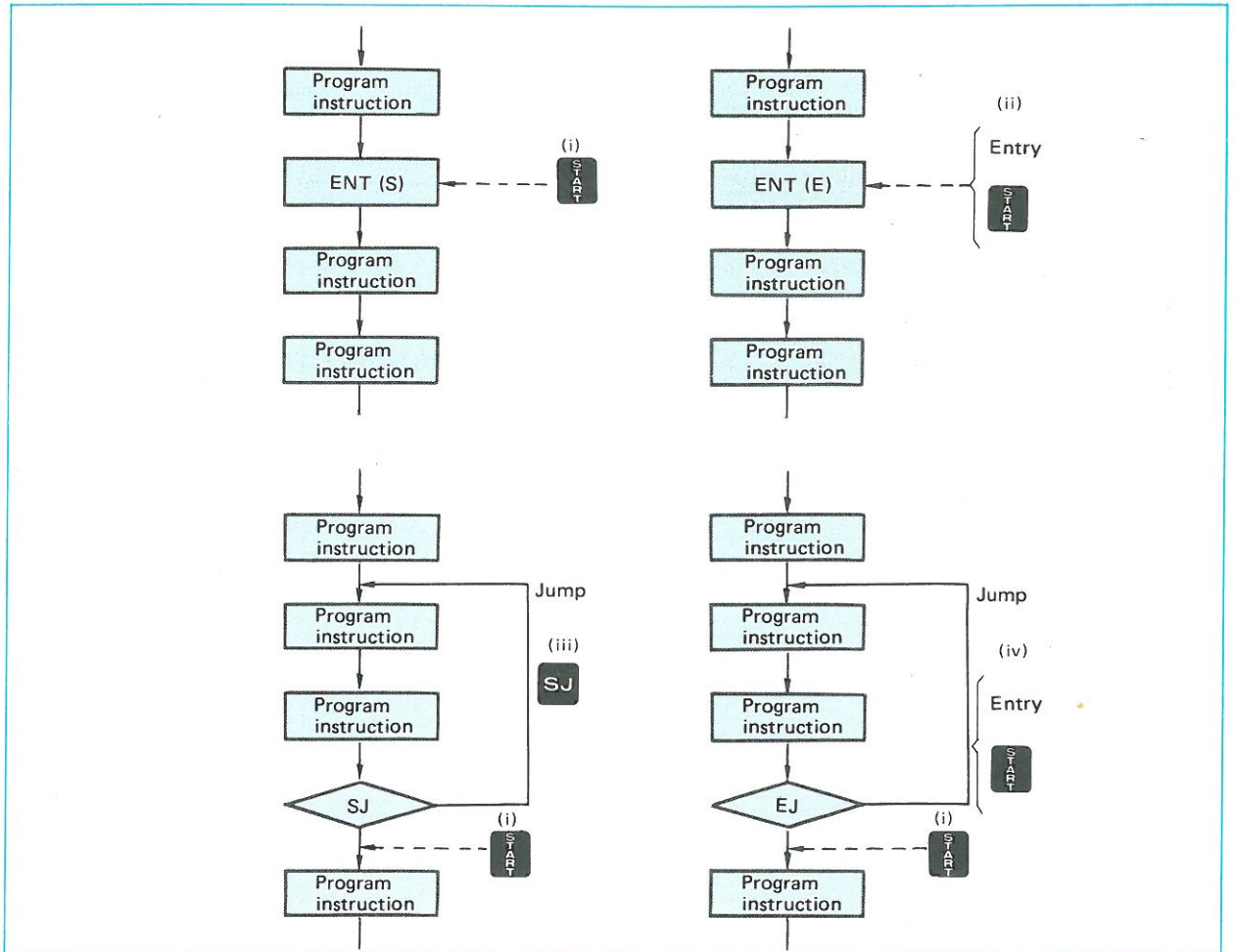
- (iv) While data is entered by the operation of numeral keys and the  key is depressed, the program is caused to jump to the specified step. If the  key is depressed without entry, the program proceeds without jumping. Respective instructions are illustrated in the following flowchart.

## 2-6 Explanation of Program Lamp

### Entry Lamp

This lamp is used when executing a program. The lamp lights when program advancement is stopped by ENT, SJ and EJ instructions indicating that the calculator is waiting for instruction from the operator. The four instructions to be put in by the operator are as follows :

- (i) The program is advanced as it is by the  key operation. (ENT, SJ, EJ)
- (ii) When data is entered by the operation of numeral keys and the  key is depressed, and the program calculation is advanced to the following step. (ENT)
- (iii) By the operation of the  key, the program is caused to jump to the specified step. (SJ)  
If the  key is depressed instead,





# 2

## EXPLANATION AS AN ORDINARY CALCULATOR

### 1. General

### 2. Calculation Examples

2-1 Addition and Subtraction

2-2 Multiplication

2-3 Division

2-4 Mixed Calculation

2-5 Constant Multiplication and Division

2-6 Power Calculation

2-6-1 Involution

2-6-2 Square Root Calculation

2-7 Calculations Using Memory (Part 1)

2-8 Calculations Using Memory (Part 2)

2-9 Supplement to Memory

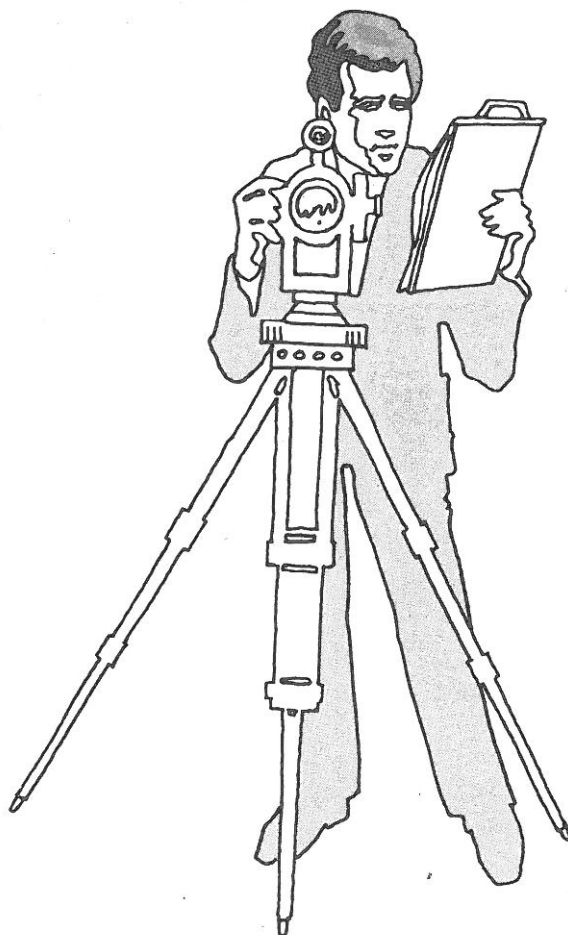
2-10 Applied Calculations

## 1. General

Ordinary manual calculation is made with the Program Mode Switch set in OPE position.

As will be discussed later in Chapter III (Explanation as a Programmable Calculator) if the power is switched on while the Program Mode Switch is set in OPE position, or the Program Mode Switch is changed from LRN or  $\begin{matrix} \text{PRO} \\ \text{CHE} \end{matrix}$  to OPE position with the power being switched on, the calculator is automatically reset to the step for manual calculation.

Consequently, manual calculation is made immediately with the Program Mode Switch set in OPE position without depressing the  $\square$  key. Both ordinary manual calculation and programs can be done in this OPE mode, because the functions of the keys for executing a program and those of the keys for ordinary manual calculation are distinctly classified. Explanation of the methods of operation is made as follows with actual calculation examples.



## 2. Calculation Examples

### 2-1 Addition and Subtraction

Calculation example	Setting	Key operation	Display
1) $123 + 456 = 579$		1 123 2 $\pm$ 3 456 4 $\pm$	123. 123.0000 456. 579.0000
2) $456 - 123 = 333$		1 456 2 $\pm$ 3 123 4 $\pm$	456. 456.0000 123. 333.0000
3) $12.34 + 1.5 + 8.765 - 4.91 = 17.695$		1 12.34 2 $\pm$ 3 1.5 4 $\pm$ 5 8.765 6 $\pm$ 7 4.91 8 $\pm$	12.34 12.340 1.5 13.840 8.765 22.605 4.91 17.695
4) $12.34 - 25 = -12.66$		1 12.34 2 $\pm$ 3 25 4 $\pm$	12.34 12.34 25. 12.66 -

- (1) Simply enter the numerals and the decimal point in their natural order. Positioning is automatic.
- (2) The minus sign lights up to indicate that the displayed figure is a negative number.
- (3) Set the Decimal Point Selector Dial according to the contents of the calculation.
- (4) Decimal digits in excess of the setting may be entered but will be dropped off (regardless of the setting of the round-off switch) when the  $\pm$  or  $\pm$  keys are depressed.

## 2-2 Multiplication

Calculation example	Setting	Key operation	Display
1) (Simple Multiplication) $123 \times 456 = 56,088$		1 123 2 $\times$ 3 456 4 $\pm$	123. 123. 456. 56,088.

- The multiplicand remains displayed when the  $\times$  key is depressed. When the multiplier is entered, the multiplicand disappears and only the multiplier is displayed.  
If the  $\text{RV}$  key is now depressed, the multiplicand is recalled. Depress the  $\text{C}$  key once more recalls the multiplier. This permits figures to be checked after entering.
- Clearing after multiplication and division is automatic. It is not necessary to depress the  $\text{C}$  key after each calculation.
- The overflow lamp does not light even though the number of integral digits in the result exceeds (16 digits—preselected decimal digits). If this exceeds 16 however, the overflow lamp lights and excess digits at the right are dropped according to the leftmost digit priority system. Both multiplications and divisions can be continued if the  $\text{C}$  key is depressed.
- If the  $\div$  key is depressed by mistake instead of  $\times$  key, simply depress the  $\times$  key after it. Calculation can be continued as normal.

Calculation example	Setting	Key operation	Display
2) (Negative Multiplication with Decimal) $-1.2 \times (-3.45) = 4.14$		1 1.2 2 $\text{CH}$ 3 $\times$ 4 3.45 5 $\text{CH}$ 6 $\pm$	1.2 1.2- 1.2- 3.45 3.45- 4.1400
3) (Multiplication with Round-Off) $123.456 \times 789.789 = 97,504.1908$		1 123.456 2 $\times$ 3 789.789 4 $\pm$	123.456 123.456 789.789 97,504.1908

- The negative signs are entered with the  $\text{CH}$  key.
- For example (2), the key operation  $1.2 \text{CH} \times 3.45 \text{CH}$  is also available.
- With settings at  $\text{5/4}$  and  $\text{0}$ , the result is obtained to four decimal places, with rounding off of the fifth decimal digit.
- With settings at  $\text{5/4}$  and  $\text{0}$ , the fifth decimal digit is dropped off and the result becomes 97,504.1907. With the Round-off Switch set at  $\text{5/4}$  rounding up occurs and the result is the same as in (3).

Calculation example	Setting	Key operation	Display	
4) (Chain Multiplication) $123.456 \times 789.456 \times 456.123$ $= 44,455,152.4096$		1	123.456	123.456
		2	$\times$	123.456
		3	789.456	789.456
		4	$\times$	97,463.079936
		5	456.123	456.123
		6	$\pm$	44,455,152.4096

- (1) Key operation in chain multiplication follows the calculation order. There is no need to depress the  $\pm$  key to obtain intermediate results.
- (2) Depress the  $\times$  key for intermediate results. The position of the decimal point in such results is floating, and is not limited by the setting of the decimal point selector. This results in greater accuracy in the final result.
- (3) The setting of the Decimal Point Selector Dial affects only the final result. At entry and in intermediate results, the position of the decimal point is floating, and figures with more than the designated number of decimal digits can be registered as they stand.
- (4) If the number of digits in the result exceeds 16, the leftmost digit priority system functions and excess digits at the right are dropped. If the number of integral digits exceeds 16, the position of the decimal point automatically indicates the number of integral digits that have been dropped. (For instance, if 3 integral digits are dropped, the decimal point appears after the third digit from the left.) In this case, the overflow lamp lights, but the actual position of the decimal point is preserved within the registers.

Calculation example	Setting	Key operation	Display	
5) (Floating Calculation) $12.345 \times 789.456$ $= 9,745.8342$		1	12.345	12.345
		2	$\times$	12.345
		3	789.456	789.456
		4	$\pm$	9,745.83432

- (1) If the Decimal Point Selector Dial is set at  $F$ , all effective digits within the capacity of the indicator are displayed.
- (2) In floating calculations, the results are always obtained without round-off or-up.

## 2-3 Division

Calculation example	Setting	Key operation	Display
1) (Simple Division) $456 \div 123 = 3.707317$		1 456 2 $\div$ 3 123 4 $\pm$	456. 456. 123. 3.707317

- (1) The dividend remains displayed when the  $\div$  key is depressed. When the divisor is entered, the dividend disappears and only the divisor is displayed.  
If the  $\text{RV}$  key is now depressed, the dividend is recalled. Depressing the  $\text{RV}$  key once more recalls the divisor. This permits checking of figures after entry.
- (2) The dividend and divisor can both be entered up to 16 digits.
- (3) The overflow lamp does not light even though the number of integral digits in the result exceeds

(16 digits—preselected decimal digits). If this number exceeds 16, however, the overflow lamp lights and excess digits at the right are dropped according to the leftmost digit priority system. Both multiplications and divisions can be continued if the  $\text{MC}$  key is depressed.

- (4) If the  $\times$  key is depressed by mistake instead of the  $\div$  key, simply depress the  $\div$  key after it. Calculations can be continued as normal.
- (5) Round-up, round-off and round-down of the decimals at the preselected decimal digit are carried out in the same way as for multiplication.

Calculation example	Setting	Key operation	Display
2) (Chain Division) $123,456,789 \div 456.123 \div 789.456 = 342.850727$		1 123456789 2 $\div$ 3 456.123 4 $\div$ 5 789.456 6 $\pm$	123,456,789. 123,456,789. 456.123 270,665.5638939496 789.456 342.850727

- (1) Key operation in chain division follows the natural order.
- (2) The position of the decimal point in intermediate results is floating, and is not limited by the setting of the decimal point selector.

## 2-4 Mixed Calculation

Calculation example	Setting	Key operation	Display
$(268 + 283) \times 1.28 \div 12 = 58.77$		1 268 2 $\pm$ 3 283 4 $\pm$ 5 $\times$ 6 1.28 7 $\div$ 8 12 9 $\pm$	268. 268.00 283. 551.00 551.00 1.28 705.28 12. 58.77

## 2-5 Constant Multiplication and Division

Calculation example	Setting	Key operation	Display
1) (Constant Multiplication)		1 359.42	359.42
$359.42 \times 2,145 = 770,955.90$		2 $\times$	359.42
$359.42 \times 4,950.6 = 1,779,344.65$		3 2145	2,145.
$359.42 \times 3,784.38 = 1,360,181.86$		4 $\pm$	770,955.90
		5 4950.6	4,950.6
		6 $\pm$	1,779,344.65
		7 3784.38	3,784.38
		8 $\pm$	1,360,181.86
	For longer calculations, repeat as in steps 7 and 8.		

- (1) In multiplication, the first figure entered after the  $\text{K}$  key is locked (i.e. the multiplicand) becomes the constant. Subsequent calculation is automatic.



Calculation example	Setting	Key operation	Display
2) (Constant Division)		1 31750	31,750.
$31,750 \div 360 = 88.1944$		2 $\div$	31,750.
$54,236 \div 360 = 150.6556$		3 360	360.
$876.12 \div 360 = 2.4337$		4 $\pm$	88.1944
		5 54236	54,236.
		6 $\pm$	150.6556
		7 876.12	876.12
		8 $\pm$	2.4337
	For longer calculations, repeat as in steps 7 and 8.		

- (1) In division, the second figure entered after  $\text{K}$  key is locked (i.e. the divisor) is memorized as the constant. Subsequent calculation is automatic.

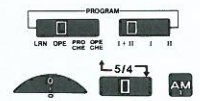
- (2) For chain multiplications and divisions with constants, the second memory must be used. In case of an intermediate change in the decimal point setting, the contents of the memory are calculated with the initial decimal point setting, independent of the new setting for the results. This function ensures high accuracy of results.





Calculation example	Setting	Key operation	Display
3) (Sum of Quotients of Chain Division) $3,761.87 \div 0.86 \div 17 = 257$ $14,377.03 \div 0.78 \div 33 = 558$ <hr/> Total 815		1 3761.87	3,761.87
		2 $\div$	3,761.87
		3 .86	0.86
		4 $\div$	4,374.267441860465
		5 17	17.
		6 $\pm$	257.
		7 14377.03	14,377.03
		8 $\div$	14,377.03
		9 .78	0.78
		10 $\div$	18,432.08974358974
		11 33	33.
		12 $\pm$	558.
		13 FM	815.
4) (Sum of Quotients of Constant Division) $31,750 \div 308 = 103.0844$ $54,236 \div 308 = 176.0909$ $78,912 \div 308 = 256.2077$ <hr/> Total 535.3830		1 31750	31,750.
		2 $\div$	31,750.
		3 308	308.
		4 $\pm$	103.0844
		5 54236	54,236.
		6 $\pm$	176.0909
		7 78912	78,912.
		8 $\pm$	256.2077
		9 FM	535.3830

### 2-8 Calculations Using Memory (Part 2)

Calculation example	Setting	Key operation	Display
1) (Sum of Products of Constant Multiplication) $1\$ = \text{¥}308$ Quantity \$ ¥ $271 \times \$35.27 \times 308 = \text{¥}2,943,916$ $123 \times \$25.32 \times 308 = \text{¥} 959,222$ <hr/> Total ¥3,903,138	 <p>308 M (Enter the constant to Memory 2)</p>	1 271	271.
		2 $\times$	271.
		3 35.27	35.27
		4 $\times$	9,558.17
		5 FM	308.
		6 $\pm$	2,943,916.
		7 123	123.
		8 $\times$	123.
		9 25.32	25.32
		10 $\times$	3,114.36
		11 FM	308.
		12 $\pm$	959,222.
		13 FM	3,903,138.

- When the **M** key is depressed, the displayed figure is memorized as the constant in the second memory.
- When the constant has a decimal point, use any of Memories 3-9 for temporary storage. (For

details, see Difference of Decimal Point Positions in Memories Due to M & SM.)

- To recall the figure from the second memory, depress the **FM** key. The recalled figure can be freely used for all calculations.








## 2-10 Applied Calculations

### Explanation of $\text{RV}$ key

The  $\text{RV}$  key interchanges the multiplicand and the multiplier in multiplications and the dividend and the divisor in divisions, and is used in percentage and ratio calculations.

For example, when finding the total and then obtaining the percentage for each item, it is normally necessary to clear the contents of the calculator and

then calculate dividend  $\div$  divisor  $\times 100$ , since the total becomes the divisor. Since the  $\text{RV}$  key interchanges the dividend and the divisor, however, the total can be left as it is, without clearing and the dividend registered as the divisor. Then all that is needed is to depress the  $\text{RV}$  key to reverse the two figures. If the  $\text{K}$  key is now depressed, the divisor is memorized as a constant and the percentages can be calculated

Calculation example	Setting	Key operation	Display
1) (Composite Ratio Calculation)		1 3524500	3,524,500.
Obtaining the sales percentage of each store against total sales :		2 $\pm$	3,524,500.00
		3 5163200	5,163,200.
		4 $\pm$	8,687,700.00
		5 4048600	4,048,600.
		6 $\pm$	12,736,300.00
		7 2975800	2,975,800.
		8 $\pm$	15,712,100.00 ... Total sales
		9 $\div$	15,712,100.00
		10 100	100.
		11 $\pm$	157,121.00
		12 $\text{AM}$	157,121.00
		13 $\text{K}$	157,121.00
		14 $\div$	157,121.00
		15 3524500	3,524,500.
		16 $\text{RV}$	157,121.
		17 $\pm$	22.43
		18 5163200	5,163,200.
		19 $\pm$	32.86
		20 4048600	4,048,600.
		21 $\pm$	25.77
		22 2975800	2,975,800.
		23 $\pm$	18.94
		24 $\text{RM}$	100.00

$$\text{Formula : } \frac{\text{Sales of each store}}{\text{Total sales}} \times 100$$

$$= \frac{\text{Sales of each store}}{157,121}$$

- (1) In certain calculations, the figure recalled at the end by the  $\text{RM}$  key may not be 100.00 This is due to the round-off of decimals and should be corrected as necessary.
- (2) The  $\text{K}$  key can be depressed anywhere before step 15.



Calculation example

3) (Sales and Gross Profit Calculation)

Cost price	Quantity	Total cost price	Installation rate	Gross profit	Sales
1,234	23	(28,382)	1.2	( 5,676.40)	(34,058.40)
4,567	12	(54,804)	1.25	(13,701.00)	(68,505.00)
789	45	(35,505)	1.3	(10,651.50)	(46,156.50)
		Total (118,691)		Total (30,028.90)	Total (148,719.90)

Setting	Key operation	Display
	1 1234	1,234.
	2	1,234.
	3 23	23.
	4	28,382.
	5 .2	0.2
	6	5,676.40
	7	5,676.40
	8	28,382.
	9	34,058.40
	10 4567	4,567.
	11	4,567.
	12 12	12.
	13	54,804.
	14 .25	0.25
	15	13,701.00
	16	13,701.00
	17	54,804.
	18	68,505.00
	19 789	789.
	20	789.
	21 45	45.
	22	35,505.
	23 .3	0.3
	24	10,651.50
	25	10,651.50
	26	35,505.
	27	46,156.50
	28	148,719.90 (Sales)
	29	30,028.90 (Gross profit)
	30	118,691.00 (Total cost price)



Calculation example	Setting	Key operation	Display
<p>4) Calculation of Redemption by Annual Installments Using <b>FV</b> Key)</p> <p>Obtaining the amount to be repaid at one installment when a principal of \$2,000,000 is to be redeemed in 8 equal installments (one installment annually). Annual interest is 6 percent.</p> <p>Formula: <math>2,000,000 \times \frac{0.06}{1 - \frac{1}{(1 + 0.06)^8}}</math></p>		1 .06	0.06
		2 <b>M</b>	0.06
		3 1	1
		4 $\frac{\pm}{\pm}$	1.060000
		5 $\times$	1.060000
		6 $\frac{\pm}{\pm}$	1.123600 ... square
		7 $\times$	1.123600
		8 $\frac{\pm}{\pm}$	1.262476 ... 4th power
		9 $\times$	1.262476
		10 $\frac{\pm}{\pm}$	1.593845 ... 8th power
		11 $\div$	1.593845
		12 1	1.
		13 <b>FV</b>	1.593845
		14 <b>=</b>	0.627413 -
		15 1	1.
		16 $\frac{\pm}{\pm}$	0.372587
		17 $\div$	0372487
		18 <b>RM</b>	0.060000
		19 <b>FV</b>	0.372587
		20 $\times$	0.161036214360672
		21 2000000	2,000,000.
		22 $\frac{\pm}{\pm}$	322,072.428721 (Redemption)

Calculation example				Setting	Key operation	Display	
5) (Depreciation Calculation) Fifth year depreciation					1	350000	350,000. . . . Acquisition cost
					2	$\frac{\square}{\square}$	350,000.
					3	$\times$	350,000.
					4	.369	0.369
					5	RM	350,000.
					6	$\frac{\square}{\square}$	129,150. - . . . First year depreciation
					7	RM	220,850. . . . First year book value
					8	$\frac{\square}{\square}$	81,494. - . . . Second year depreciation
					9	RM	139,356. . . . Second year book value
					10	$\frac{\square}{\square}$	51,422. - . . . Third year depreciation
					11	RM	87,934. . . . Third year book value
					12	$\frac{\square}{\square}$	32,448. - . . . Fourth year depreciation
					13	RM	55,486. . . . Fourth year book value
					14	$\frac{\square}{\square}$	20,474. - . . . Fifth year depreciation
					15	RM	35,012. . . . Fifth year book value
Rate of depreciation							
Durable years	One business year	One business half-year					
2	.684	.438					
3	.536	.319					
4	.438	.250					
5	.369	.206					
6	.319	.175					
7	.280	.152					
8	.250	.134					
9	.226	.120					
10	.206	.109					





# 3

## EXPLANATION AS A PROGRAMMABLE CALCULATOR

### 1. General

### 2. Before Program Operation

2-1 Explanation of Terms

2-2 Explanation of Symbols Used

### 3. For Program Operation

3-1 Introduction

3-2 Manual Learn System

3-3 Clearing of Stored Programs

3-4 Method of Checking and Correcting Program Contents

3-5 Incorporation of Constant in Program

3-6 Program Step Settings

3-7 Interpolated Calculation

3-8 ROUND Instructions and Printer Instructions

3-9 Decision Functions (Jump Functions)

3-10 Subroutine Function

3-11 Card System

3-12 Canon Program Card

3-13 Cross Reference of Program Instruction and Code

3-14 Input Data by Card

3-15 Correction & Clearing of Program Instruction Written into Cards

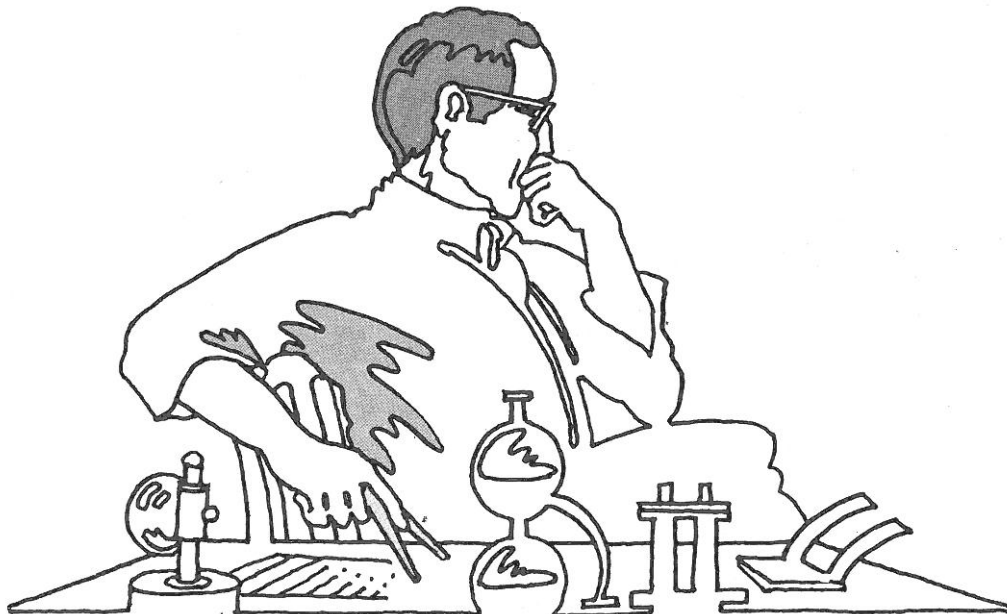
## 1. General

The fundamental operations of the 1614P as an ordinary calculator should have been understood by reading the preceding chapter. However the principal object of the 1614P is its use as a programmable calculator.

Program storage is accomplished by the stored program system (internal storage system), through two different input systems. This is one, the "manual system" whereby calculation procedures are stored as a program by keyboard operation ;and the other "card system" whereby a program, precoded and punched into a card (Canon program card), is read into the 1614P through the built-in program card-reader. Relatively simple calculations can be accomplished by

using the manual system on the keyboard, but more complicated repeated calculations can be read into the program storage using the card system for immediate use.

The most outstanding feature of the 1614P as a programmable calculator is its card system, which utilizes Canon program cards. By programming calculations which are often involved in routine work and by providing a library of program cards, it is possible to greatly shorten the time required for calculation and improve overall calculation accuracy. Features and their method of use are explained one after another in the following pages.



## 2. Before Program Operation

Before commencing actual detailed explanation, basic fundamentals such as special terms and symbols necessary for program preparation and related procedures are described.

### 2-1 Explanation of Terms





The following terms will be used in the sense of the succeeding explanation below.

- 1) Program instruction
- 2) Program step *parto de programa*
- 3) Programming
- 4) Program storage *biblioteca de programas?*
- 5) Execution of program calculation

#### 1) Program Instruction

An instruction is a clear command to actuate the calculator in a particular way, resulting in a mathematical operation usually. There are 95 instructions in this machine. Examples are X, M<sub>1</sub>, M<sub>2</sub>, etc.

#### 2) Program Step

A program is made of a series of sequential instructions. Each instruction is a step in a total program, so that by defining the step number we determine at what step the program has reached. The maximum number of steps in programs for this machine is 240. Further, , ,  and  keys are used in two operations in combination with 3-9 numeric keys in the manual system, are entered as one program instruction. Consequently, they are also counted as one program step. (See Program Instruction ↔ Code Cross Reference Table)

#### 3) Programming

The term programming means work involved in preparation of a program (calculation processing procedure) to be stored in the calculator for execution of program calculation by simple operation.

This program is prepared by using the program instructions of the Canola 1614P.

The program thus prepared is stored in the calculator by manual keyboard operation for the manual system ; and, the program is stored in the calculator by punching the program on cards according to "Program Instruction ↔ Code Cross Reference Table" and then reading the card into the calculator

through the built-in program card-reader.

#### 4) Program Storage

Program storage means work involved in storing the contents of a program into the calculator.

Since storage methods for manual and card systems vary, they are explained separately.

##### (i) Manual System

The programmed program is stored in the machine by manual step-by-step keyboard operation with the Program Mode Switch in LRN position.

##### (ii) Card System







The program is stored in the machine by reading the card punched according to the contents of the program with the card-reader.

#### 5) Execution of Program Calculation

This term means work involved in operation required in obtaining an answer from a stored program through entry of data with the numeric keys and related key operation with the Program Mode Switch set in OPE position.

### 2-2 Explanation of Symbols Used

Each symbol is used according to the following :

- 1) Manual key operation is represented by , i.e., , , .
- 2) Program instruction is represented without , i.e., M<sub>1</sub>, SJ, ±.
- 3) Program instructions E and S : The program key  and program instruction ENT which correspond to it are used to temporarily stop program advancement. In order to facilitate better understanding, observe the following two examples.
  - 1) E (Entry) . . . Entry instruction which places the machine in an entry waiting state.
  - ii) S (Stop) . . . Temporary stop instruction to read calculation results.

### 3. For Program Operation

#### 3-1 Introduction

The basic functions and method of operation which should be mastered in making actual programs using the 1614P are explained in this chapter.

Earlier we discussed usage of the machine with its manual system and with its card system. Except for a few variations in usage of the Canon program card system there is almost no difference between the two systems. Therefore, explanation in the first part will be based on the manual system followed by discussion on points inherent only to the card system.

#### 3-2 Manual System

Manual system means storing of a calculation procedure as a program by means of ordinary keyboard operation as per the calculation formula, so that subsequent calculation answers are obtained merely by entering data (variables) and operating the  $\square$  and  $\square$  keys.

(Ex.)  $a \times b \times c = (d)$

If keys  $\square$ ,  $\square$ ,  $\square$ ,  $\square$ ,  $\square$  and  $\square$  are operated as indicated in a calculation formula, a "variable x variable x variable =" program is stored.

For execution of this calculation, only by entering the variables with numeric keys and also advancing the program with the  $\square$  key, like  $\square$  3  $\square$  4  $\square$  5  $\square$ , the answer (60) of  $3 \times 4 \times 5$  can be obtained.

#### 3-2-1 Programming Under Manual System

##### 1) Program Instructions Used for Programming

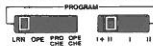
95 kinds of program instructions can be used for programming by using 38 program key combinations excluding  $\square$ ,  $\square$ ,  $\square$  and  $\square$ .

##### 2) Programming

A program (calculation process) to be stored in the calculator is composed to execute program calculations by simple operations. This program is composed by using the program instructions for this calculator.

#### 3-2-2 How to Store the Program

##### 1) Setting of Program Switches

Set the Program Switches in LRN and I + II positions like 

(The Program Selector Switch is represented in the position of I + II hereinafter.)

##### 2) Clearing of Previous Program Contents

When the first program key is depressed under manual system, previously stored programs are cleared automatically and the new program instruction is stored. Consequently, it is not necessary to manually clear the program by depressing the  $\square$  key or the like. (See the section for clearing programs for further details.)

##### 3) Program Storage

A program is stored by sequentially depressing program keys corresponding to the respective program instructions of the program to be stored.

#### 3-2-3 Execution of Programs by Manual System

1) When program storage is completed, change the Program Mode Switch to OPE position from LRN position.

With this changeover, the program will return to the step of manual calculation (zero step). (See Section III-3-6 Program Step Settings for further details.)

2) Depress the  $\square$  key once to bring the program located in the manual calculation step to the area of program calculation.

Thus preparation for execution of program calculation is completed. When using a memory in the program, the memory shall be cleared in advance as required by means of the  $\square$  key or equivalent.

3) Then subsequent answers can be easily obtained by data input and simple operation of the  $\square$  and  $\square$  keys according to calculation procedure.

#### 3-3 Clearing of Stored Programs

Program correction, to be explained later, involves program clearing but it shall be understood that clearing means clearing of program segments.

To clear a program, first set the Program Mode Switch in LRN position. If the  $\square$  key is then manually depressed, a program within the range is

cleared in accordance with the setting of the Program Selector Switch.

The relationship between Program Selector Switch and the range to be cleared is as noted below.

Program Selector Switch (Range to be cleared)

(i)	I + II	1 - 240 steps	All cleared
(ii)	I	1 - 120 steps	Partial
(iii)	II	121 - 240 steps	cleared

(See the section of I-2-5 Program Selector Switch discussed earlier)

The programs are cleared in the above mentioned procedure, but this method is not applicable for making ordinary program calculations. A separate method is available that automatically clears the programs within the required range without depressing the  $\square$  key.

In the 1614P, when the Program Switch is set in LRN position to store programs, the program will auto-

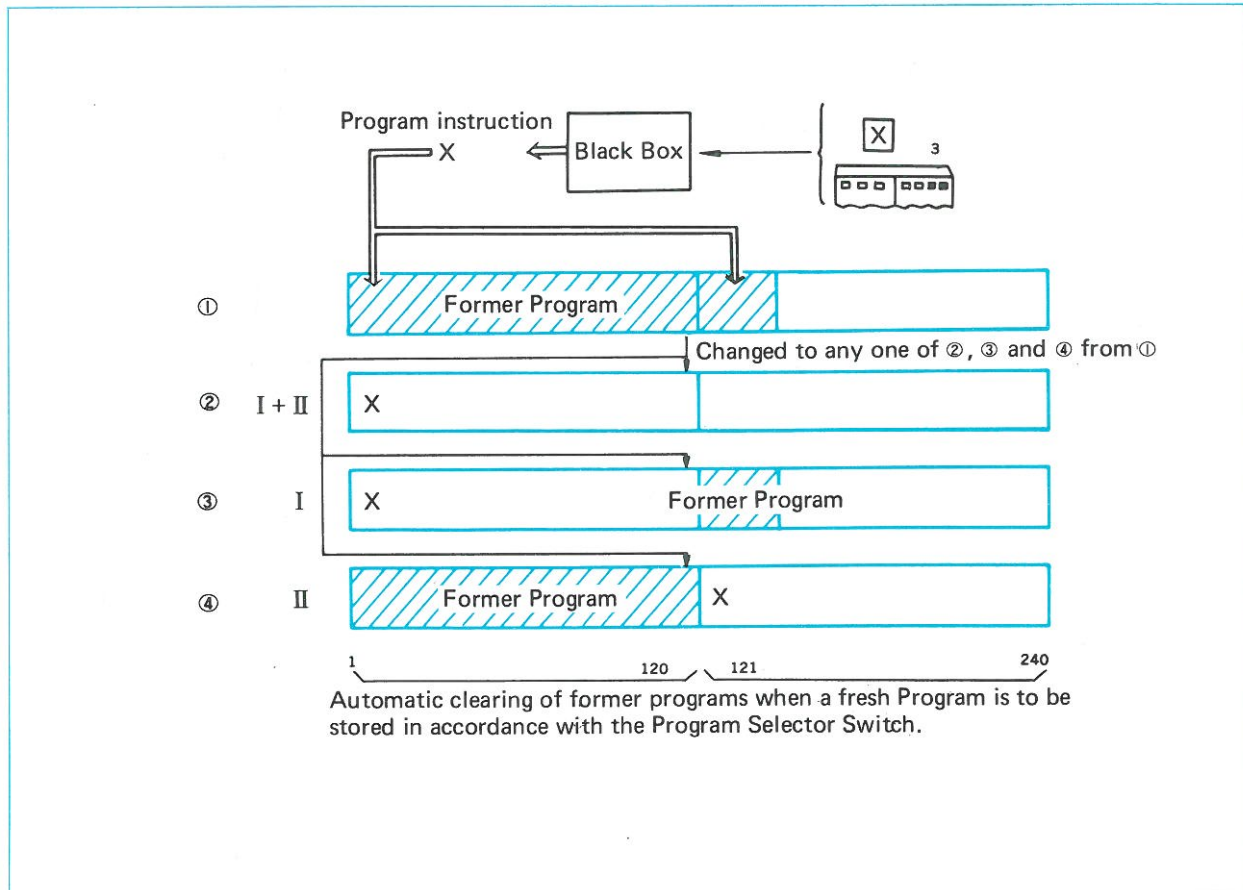
matically return to the 1st or 121st step in accordance with the Program Selector Switch.

In the earlier explanation operation by the  $\square$  key in this stage was needed, but in this case the program is directly stored through manual key operation or program card reading, without depressing the  $\square$  key. Accordingly, the following actions will occur in the calculator. When manual entry of program instruction is made, the first program key depressed, or the first punched hole on a program card is read, simultaneously a previous program ( a series of instructions) is automatically cleared.

More specifically, the program instruction first stored in the calculator functions in place of the  $\square$  key.

The range to be cleared is dependent on the instruction given by the Program Selector Switch just as when the  $\square$  key is used.

The above relationship is illustrated as follows:



### 3-4 Method of Checking and Correcting Program Contents


Verification of the stored program instructions can be easily checked step-by-step (Program check), and if an error is detected, a manual correction of that step can be easily done.

Actual numerical data procedure of a programmed calculation can also be verified step-by-step (Operation check).

#### 3-4-1 Program Check










When the Program Mode Switch is set in  $\text{PRO}_{\text{CHE}}$  position, it is possible to check the stored program contents.

To check, depress the  key and advance the program step-by-step.

As the program is advanced step-by-step by depressing the  key, the program step and the code of the program instruction are indicated on the indicator correspondingly as shown below. The place to the left in the three-place program instruction code is used for major breakdown and corresponds to the three bits at the left of the Canon program card. The two places to the right are used for minor breakdown and correspond to the four bits at the right of the card. The operator checks the contents while making cross-reference of the indicated program step and the program code instruction.

This may become clearer in the following example.


(Ex.) Area of a circle  $S = \pi r^2 (= r^2 \times \pi)$  ;  
 $\pi = 3.14$

Step	Program instruction	Key operation	Display										Remarks							
1	E											0	0	1	1	0	1			r
2	x											0	0	2	0	0	3			
3	$\frac{\pm}{=}$											0	0	3	0	0	1			r <sup>2</sup>
4	x											0	0	4	0	0	3			
5	3											0	0	5	7	0	3			
6	.											0	0	6	6	0	0			
7	1											0	0	7	7	0	1			} $\pi$
8	4											0	0	8	7	0	4			
9	$\frac{\pm}{=}$											0	0	9	0	0	1			$\pi r^2$

Program Contents Check

7 digits zero-suppressed program step  
 1 digit zero-suppressed program instruction code  
 2 digits zero-suppressed

#### 3-4-2 Operation check

With the Program Mode Switch set to  $\text{OPE}_{\text{CHE}}$ , operation of the  key results in operation and execution of stored program instructions step-by-step.

The program check described previously checks the contents of the stored program. The operation check

on the other hand ascertains and confirms step-by-step operation of the assembled program, using actual data.

Let's see how this works in relation to the calculation of circular area.

Area of a circle :  $S = \pi r^2$        $r = 2$



Step	Program instruction	Key operation	Display	Remarks
0			0	
1	E		2	r
2	X		2	
3	$\pm$		4.00	$r^2$
4	X		4.00	
5	3		3	}
6	·		3	
7	1		3.1	$\pi$
8	4		3.14	
9	$\pm$		12.56	$\pi r^2$

In repetitive loop calculation too, step-by-step operation is effected according to jump instructions.

Let's explain with an example of factorial calculation :

$$n! = n (n-1) (n-2) \dots \dots \dots 2 \cdot 1$$

(For details concerning  $n!$ , refer to information on concepts and assembly of programs through actual examples.)

Step	Program instruction	Key operation	Display	Key operation	Display	Key operation	Display	Remarks
0			0					
1	E		3					
2	SM <sub>3</sub>		3					
3	1		1					
4	SM <sub>4</sub>		1					
5	FJ							
6	1							
7	RM <sub>3</sub>		3		2		1	
8	X		3		2		1	
9	RM <sub>4</sub>		1		3		6	
10	$\pm$		3		6		6	
11	SM <sub>4</sub>		3		6		6	
12	1		1		1		1	
13	SC		- 1		- 1		- 1	
14	M <sub>3</sub>		- 1		- 1		- 1	If a figure becomes minus at step 16, a jump takes place to step 5.
15	RM <sub>3</sub>		2		1		0	
16	SC		- 2		- 1		0	
17	MJ							
18	1							Upon completion of the loop, advance to the next step takes place.
19	RM <sub>4</sub>						6	

### 3-4-3 Correction

If an error is discovered as a result of the program check, it is possible to correct only the error and leave the correct portion as it is. Explanation is made with an actual example.

(Ex.) Area of a circle  $S = \pi r^2 (= r^2 \times \pi)$  ;  
 $\pi = 3.14$

The program for this example is as described in the paragraph for checking,

$E \times \pm \times 3.14 \pm$

Assume that the program instruction of the 4th step  $x$  is mistaken as  $\div$ .

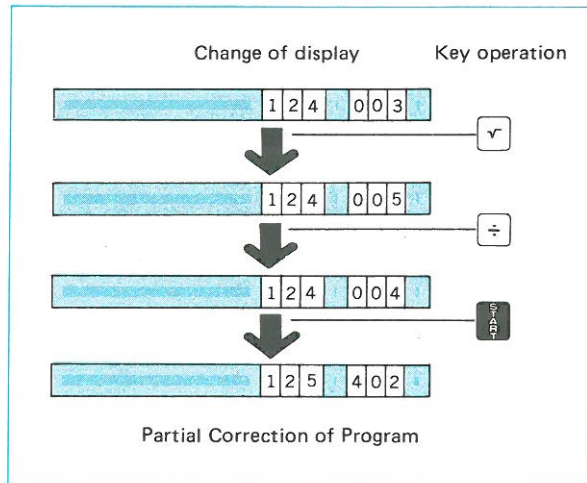
In other words, the wrong program is

$E \times \pm \div 3.14 \pm$

If this wrong program is checked as explained previously, the indication is made as 004 in the 4th step, even though it should be indicated as 003, and thus it is found the wrong program instruction  $\div$  is stored instead of the program instruction  $x$ .

To change this wrong program instruction code 004, that is program instruction  $\div$ , into the correct program instruction code 003, that is program instruction  $x$ , depress program key  $\boxed{x}$  which corresponds to program instruction code 004 is changed to 003 with the program step unchanged as illustrated below. Thus, the wrong program instruction  $\div$  is corrected. As explained above, partial program correction can be accomplished by simply depressing the program key corresponding to the program instruction to replace an error.

Unless the  $\boxed{\text{P}} \rightarrow \text{P}$  key is depressed, the program is not advanced. Therefore, program keys may be repeatedly depressed, and the program instruction which corresponds to the program key depressed last is stored as program instruction for that particular step. In the example shown below, program instruction  $\div$  is finally stored.




### 3-4-4 Program Check and Correction Supplement


Methods of checking and correcting a program are as explained above. In addition, the following method described below can also be used to check and correct a part of a long step program.

For example, it is possible to check only the 65th step and subsequent steps such as in the program shown below.

Step	Program instruction	Key operation
		OPE
1	E	
2	X	
3	$\pm$	
4	X	
5	$\cdot$	
6	3	
7	5	
8	7	
9	3	
10	$\div$	
11	2	
12	X	
~~~~~		
60	$\pm$	
61	M <sub>1</sub>	
62	RM <sub>2</sub>	
63	E	OPE $\rightarrow$ PRO CHE
64	$\pm$	
65	RM <sub>3</sub>	
66	X	
67	E	
68	$\pm$	
69	$\sqrt{\quad}$	
70	$\div$	
71	RM <sub>1</sub>	
72	$\pm$	
73	$\sqrt{\quad}$	

In such a case, first set the Program Mode Switch in OPE position and then advance the program to E of 63th step by depressing the  key twice.




Then change the Program Mode Switch to PRO CHE position from OPE position.

By this change-over, the program is set to the step next to the step of the change-over, it is possible to check step-by-step by depressing the  key thereafter. In this case, however, there must be no program instructions such as E, S, SJ and EJ to stop the program between the 2nd and 62nd steps.





### 3-5 Incorporation of Constant in Program

If it is necessary to incorporate a constant such as  $\pi$  (3.14), make the following operation.

1) Manual system

Set the Program Mode Switch in LRN position and depress numeric keys  ~ , and  and one after another in correspondence with the value to be incorporated in the program as a constant.

(Ex.) If 3.14 is the constant

Step	Key operation	Constant	Remarks
1		3	} 4 steps
2		.	
3		1	
4		4	

Thus 3.14 is entered in the program as a constant.

2) Card system

Under the card system, write a constant into the card by punching the card according to the codes converted from the value to be incorporated into the program as a constant.

When this punched card is read by the card-reader, the intended value is stored in the calculator as a constant. (See Section III 3-13 Cross Reference of Program Instruction and Code for the relationship between constants and codes).

(Ex.) To write 3.14 into the card

- (i) Convert 3.14 into codes
- (ii) Punch a card in accordance with the converted codes.

CANON PROGRAM CARD							
	40	20	10	8	4	2	1
1	■	■	■	□	□	■	■
2	■	■	□	□	□	□	□
3	■	■	■	□	□	□	■
4	■	■	■	□	■	□	□

Code      Constant

703  $\longrightarrow$  3

600  $\longrightarrow$  .

701  $\longrightarrow$  1

704  $\longrightarrow$  4

Thus a constant 3.14 is written in a card.

### 3-6 Program Step Settings

Programs are reset to the following program steps respectively by the changeover of the Program Mode Switch and by means of the program reset key  $\square$ .

#### 1) LRN

(i) Set the Program Mode Switch in LRN position, turn the power ON, or depress the  $\square$  key. Thus the program is automatically reset to the 1st or 121st step as specified by the Program Selector Switch. If the  $\square$  key is depressed, the former program is cleared at the same time, as described earlier.

(ii) When the Program Mode Switch is set in OPE position and if the switch is changed over to LRN position from the original position, the program is automatically reset to the 1st or 121st step as specified by the Program Selector Switch.

#### 2) OPE

(i) If the power is turned ON or the  $\square$  key is depressed with the Program Mode Switch set in OPE position, the program is reset to a step of manual calculation (assume that there is a step or area called

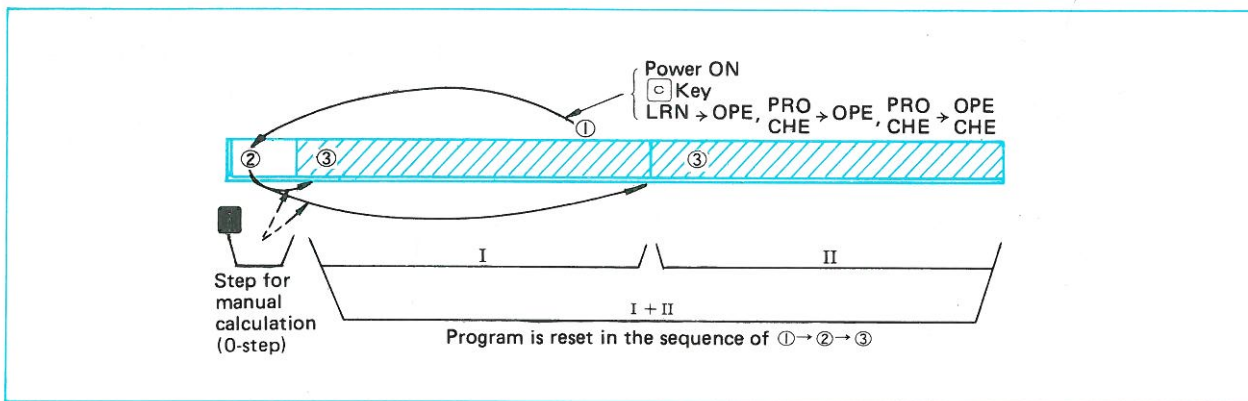
0-step before the 1st step, exclusive for manual calculation) instead of being reset to the 1st or 121st step as when the switch is in LRN position.

(ii) When the Program Mode Switch is set in either LRN or  $\frac{\text{PRO}}{\text{CHE}}$  position, if the switch is changed over to OPE position from the original position, the program is also reset to 0-step (exclusive for manual calculation). Consequently it is necessary to advance the program from the step of manual calculation to the area of program calculation, and for this purpose depress the  $\square$  key once.

Operation of the  $\square$  key for this purpose shall be considered as preparation for execution of program calculation.

The program which moves off the area of manual calculation by means of the  $\square$  key is reset to the 1st or 121st step as specified by the Program Selector Switch.

The above mentioned relationship is illustrated as follows :



#### 3) $\frac{\text{PRO}}{\text{CHE}}$

(i) If the power is turned ON or the  $\square$  key is depressed with the Program Mode Switch set in  $\frac{\text{PRO}}{\text{CHE}}$  position, the program is reset to the 1st or 121st step as specified by the Program Selector Switch.

(ii) If the Program Mode Switch is changed over to  $\frac{\text{PRO}}{\text{CHE}}$  position when it was set in either OPE or  $\frac{\text{OPE}}{\text{CHE}}$  position, the program goes to the next step to the step where the switch was in OPE or  $\frac{\text{OPE}}{\text{CHE}}$  position.

#### 4) $\frac{\text{OPE}}{\text{CHE}}$

Step-settings of  $\frac{\text{OPE}}{\text{CHE}}$  conforms to that of OPE. The program is always reset to a step of manual calculation.

Table of Program Reset

Operation Switch position	Power applied, $\square$ key operation	Program control switching
LRN	Step 1 or 121	Step 1 or 121
OPE	Area of manual calculation	Area of manual calculation
PRO CHE	Step 1 or 121	Next step at time of switching
OPE CHE	Area of manual calculation	Area of manual calculation

### 3-7 Interpolated Calculation

#### 1) Interpolated Calculation during Execution of Program Calculation.

During execution of program calculation, the advancement of the program is temporarily stopped at the locations of E, S, SJ and EJ instructions in the program. At these "stopped" locations manual cut-in calculation can be made freely.

For example, data can be quite freely read into E instruction after advance processing.

This will come clear by following the actual example below.

(Ex.) To input as data the answer (18.84) of  $2 \times 3 \times 3.14$ .

(i) Operate keys  $\square$   $\times$   $\square$   $\times$   $\square$   $\cdot$   $\square$   $\square$   $\square$  in this sequence and obtain the answer (18.84).

(ii) Then by operating the  $\square$  key, the answer (18.84) of  $2 \times 3 \times 3.14$  is input as data.

This is because the function of the  $\square$  key used in execution of program calculation and the function of the  $\square$  key used in an ordinary manual calculation are completely separated and each is independent.

#### 2) Interpolated calculation in the step (zero step) for manual calculation

If power is ON with the Program Mode Switch set in OPE position, the  $\square$  key is depressed, or the Program Mode Switch is changed over to OPE position from either LRN or PRO CHE position, the program is reset to the area for manual calculation; and to execute program calculations it is necessary to draw out the program to the area of program calculation by depressing the  $\square$  key as a preparatory step. The above was discussed earlier.

It was also explained that in this step for manual

calculation, ordinary manual calculations can be freely executed, that is entry can be freely accomplished.

From the above facts, it is possible to regard that in some of the programs commencing with entry instructions in the 1st step the entry instruction of the 1st step is a cut-in calculation in the step (area) for manual calculation.

Description is given below using an actual example in comparison with an ordinary case.

(Ex.) The program of  $a \times b \times c =$  and the procedure for execution of program calculation are described in correlation with an ordinary case. (Provided that  $a, b, c \neq 0$ )

Step	Regular		Cut-in		Remarks
	Program instruction	Key operation	Program instruction	Key operation	
					Step for manual calculation
1 2 3 4 5 6	E X E X E ±		X E X E ± *(S), (E)		

As seen from the cross-reference table, the program instruction E which corresponds to variable *a* is omitted in the case of cut-in, and it is processed as a cut-in calculation in the step for manual calculation. Consequently, operation of the key as a preparation to draw the program out of the step for manual calculation is also omitted.

\* In this case, however, S (stop) instruction will be required at the last (6th) step in order to prevent the program to return automatically to the 1st step and then to proceed to the 2nd step where E instruction for entry of the 2nd variable. Thus, in execution of the 2nd and subsequent program calculations, this S instruction also functions as E instruction for entry of the 1st variable.

### 3-8 ROUND Instructions and Printer Instructions

These instructions are used exclusively in the Card system.

#### 3-8-1 ROUND Instructions ( $\uparrow$ , 5/4, $\downarrow$ )

There are 3 kinds of ROUND instructions, employed respectively for round-up (  $\uparrow$  ), round-off ( 5/4 ), and round-down (  $\downarrow$  ) according to the programmed number of decimal places, and which take preference over the setting of the Decimal Point Selector Dial.

Round-up	$\uparrow$	<i>n</i>
Round-off	5/4	<i>n</i>
Round-down	$\downarrow$	<i>n</i>

Each ROUND instruction is always followed by a number (0 through 14) which specifies the number of decimal places. Thus a ROUND instruction is a 2-step Instruction.

step	:	round-up	:	round-off	:	round-down
M	$\uparrow$		5/4		$\downarrow$	
M + 1		<i>n</i>		<i>n</i>		<i>n</i>
	:		:		:	
	:		:		:	

*n* denotes the specified number of decimal places and can be any number from 0 to 14. The punching of a card is made as shown here.

		CANON PROGRAM CARD						
Step	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1	40	20	10	8	4	2	1
		1		■	■	■	□	□
2		■	■	■	□	□	□	■
3		■	■	■	□	□	■	□
4		■	■	■	□	□	■	■
5		■	■	■	□	■	□	□
6		■	■	■	□	■	□	■
7		■	■	■	□	■	■	□
8		■	■	■	□	■	■	■
9		■	■	■	■	□	□	□
10		■	■	■	■	□	□	■
11		■	■	■	■	□	■	□
12		■	■	■	■	□	■	■
13		■	■	■	■	■	□	□
14		■	■	■	■	■	□	■
15		■	■	■	■	■	■	□

(1) ROUND instructions in programs are valid for results of multiplication, division, addition, subtraction

tion, and square root operations during program execution.

However, the function of a ROUND instruction varies according to the type of the calculation ;

multiplication, division . . . . . decimal point specification and round-up, round-off or round-down

addition, subtraction, square root

. . . . . decimal point specification only

Step	Program	Key operation	Display	Dial
20	E (a)	$\boxed{5}$	5.	
21	M <sub>1</sub>		5.	↓
22	÷		5.	
23	E (b)	$\boxed{3}$	3.	
24	5/4		3.	
25	6		3.	↓
26	$\pm$		1.666667	
27	RM <sub>1</sub>		5.00	

The Round-off instruction at the 24th step and 25th step functions under the  $\pm$  instruction at the 26th step taking preference over the settings of the Decimal Point Selector Dial and the Round-off Switch. However, the setting of the Round-off Switch and the setting of the Decimal Point Selector Dial take effect again at the 27th and subsequent steps.

(2) ROUND instructions ahead of a stop instruction (ENT, EJ, or SJ) are valid until it comes to the part  $\pm$ ,  $\equiv$ , unless in this example there is a  $\sqrt{\quad}$  instruction during the manual input.

In this manner, the ROUND instruction in programmed operation takes effect.

Step	Program	Key operation	Display	Dial
11	E (a)	$\boxed{5}$	5.	
12	M <sub>1</sub>		5.	↓
13	$\nabla$		5.	
14	2		5.	
15	E (b)	$\boxed{3}$	3.	
		$\boxed{r}$	1.73	↓
16	X		1.73	
17	RM <sub>1</sub>		5.000000	
18	$\pm$		8.700000	

At step 15 of the Round-down instruction of the 13th and 14th steps takes effect by the execution of a square root. Hence this Round-down instruction does not function under the  $\pm$  instruction at step 18. Consequently, it is advisable to program ROUND instructions just ahead of  $\pm$ ,  $\equiv$ , and  $\sqrt{\quad}$  instructions. (3) ROUND instructions for addition and subtraction during execution of a program function only to specify the decimal point, and are valid until the next  $\pm$  instruction after the ROUND instruction.

Step	Program	Key operation	Display	Dial
20	E (a)	$\boxed{1} \boxed{.} \boxed{2} \boxed{3} \boxed{4} \boxed{5}$	1.2345	
21	$\pm$		1.2345	↓
22	E (b)	$\boxed{4} \boxed{.} \boxed{3} \boxed{2} \boxed{1} \boxed{5}$	4.3215	
23	$\pm$		5.5560	
24	E (c)	$\boxed{.} \boxed{2} \boxed{2} \boxed{2} \boxed{8}$	0.2228	
25	$\equiv$		5.3332	↓
26	E (d)	$\boxed{1} \boxed{.} \boxed{3} \boxed{3} \boxed{3} \boxed{4}$	1.3334	
27	$\uparrow$		1.3334	↓
28	2		1.3334	
29	$\pm$		6.66	↓
30	E (e)	$\boxed{1} \boxed{.} \boxed{2} \boxed{3} \boxed{4} \boxed{5}$	1.2345	
31	$\pm$		7.8945	


The round-up instruction at steps 27 and 28 functions only under the  $\pm$  instruction at the 29th step, so that it is not valid at the  $\pm$  instruction of step 31. Thus, in order to obtain a total of  $a+b-c+d+e$  to 2 decimal places with the Dial setting at 3 or more, it is necessary to program a ROUND instruction between steps 30 and 31. However, if round-up, round-off, or round-down of a total is required, the following 5 steps are needed just after the last  $\pm$  or  $\equiv$  instruction to obtain the total;

$\pm$	7.9011	(total of $a+b-c+d+e$ )
X	7.9011	
1	1.	
$\uparrow$	1.	
2	1.	
$\pm$	7.91	

### 3-8-2 Printer Instructions

Two instructions for printing and paper feeding are employed in Canola 1614P when optional Canon Printer Unit P-1 is connected.

#### (1) Print Instruction ( $\diamond$ )

With the Printer Unit P-1 connected and power ON, displayed numbers are printed when the program arrives at a Print instruction step. After printout, the program operation automatically proceeds to the next step. When the Printer Unit is not connected or its power switch is OFF, the program operation stops at the Print instruction step. To advance the program it is necessary to depress the  key. Thus, a Print instruction can be used as a Stop ( S ) instruction perhaps to read a calculation result in the display when the Printer Unit is not utilized.

#### (2) Paper Feed Instruction (FD)

With the Printer Unit P-1 connected and power ON, the roll paper is advanced by one line. When the program arrives at a Paper Feed instruction step, and after paper feeding, the program automatically proceeds to the next step. When the Printer Unit is not connected or its power switch is OFF, the Paper Feed instruction is disregarded and nothing occurs at this instruction step.

For greater detail on the optional instructions just described and on the printout function, refer to the Instruction Manual for the Canon Printer Unit P-1.



### 3-9 Decision Functions (Jump Functions)

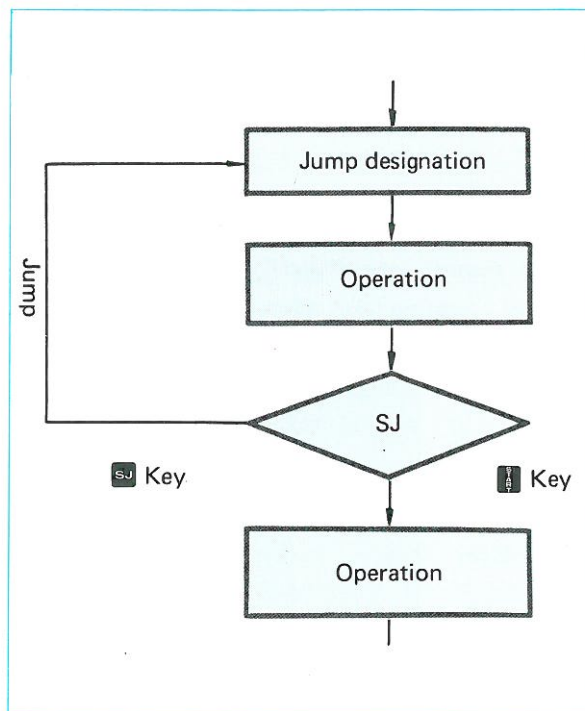
Normally during execution of program calculation, the program calculation advances sequentially starting from the 1st step of the stored program. However, the 1614P is provided with a function to stop the program flow in the middle, to decide the "then" conditions and to execute once again the program which has been just executed or to execute the program in a required portion with a certain range jumped off.

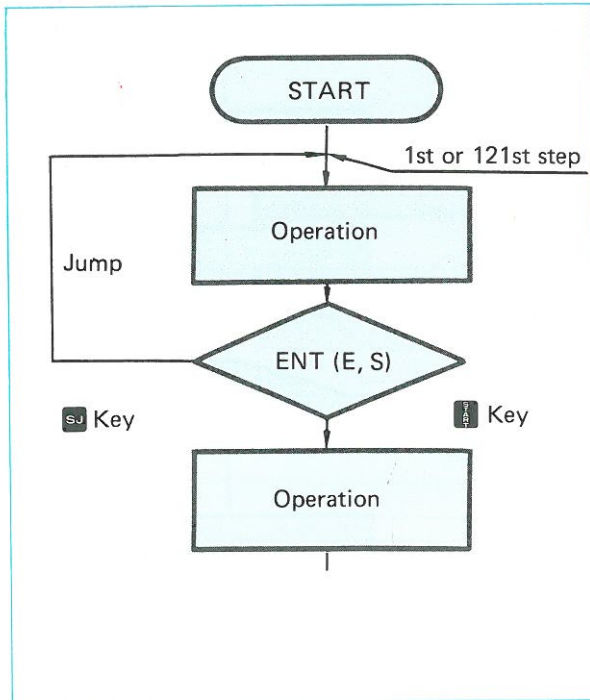
Such a function is called a decision function (jump function).

#### 3-9-1 Sense Jump (SJ)

During execution of program calculation the program temporarily stops advancement at the SJ instruction. At this point the operator decides whether to cause the program to jump to the specified program step or to cause the program to advance to the next step without jumping.

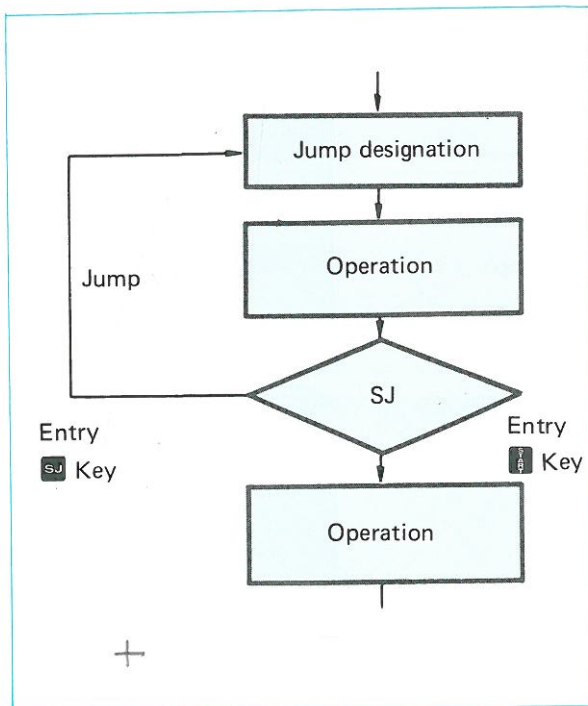
If it is decided to cause the program to jump, the  key is manually depressed, and the  key is depressed if it is intended to cause the program to advance without jumping.





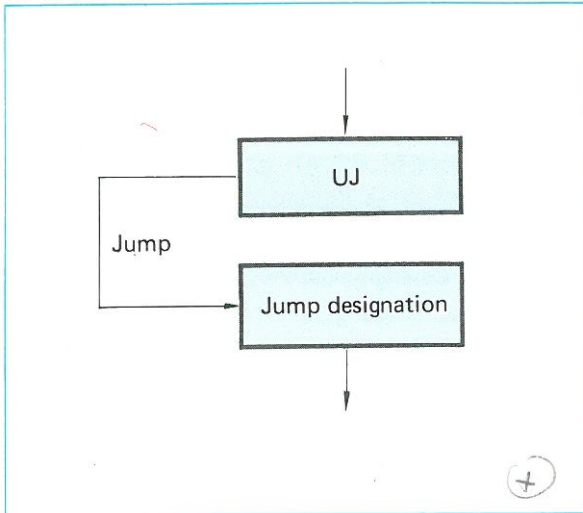
[ Supplement to Sense Jump ]

1) Program jumping by the operation of the **SJ** key should be accomplished at the SJ instruction in the program as a rule. However, it is possible to cause it at the ENT (E, S) instruction in the program if required. However, jump designation is limited to the 1st step if the Program Selector Switch is in I + II or I position, and to the 121st step if the same is in II position.



2) Further, it is possible to make an entry as required at the SJ instruction. In this case, depressing the **SJ** key after entry causes the program to jump to a specified step, or depressing the **ENT** key causes the program to advance without jumping.





### 3-9-4 Unconditional Jump (UJ)

As the program advances and comes across this program instruction, unlike MINUS JUMP, the program is caused to unconditionally jump to the specified program step regardless of the then calculated value.

(Supplement to Unconditional Jump)

Minor division of the program by UJ

Major division of the program area can be made into two groups of 120 steps by the Program Selector Switch.

Using the unconditional jump function, these two groups of 120 steps or the 240 steps as a whole can be subdivided, thereby allowing several independent programs to be stored simultaneously.

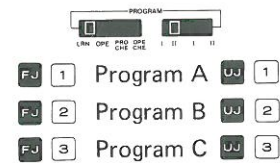
Access to each of the independent programs stored simultaneously is made by depressing the **UJ** key and

one of the numeral keys corresponding to the symbol number of the programs.

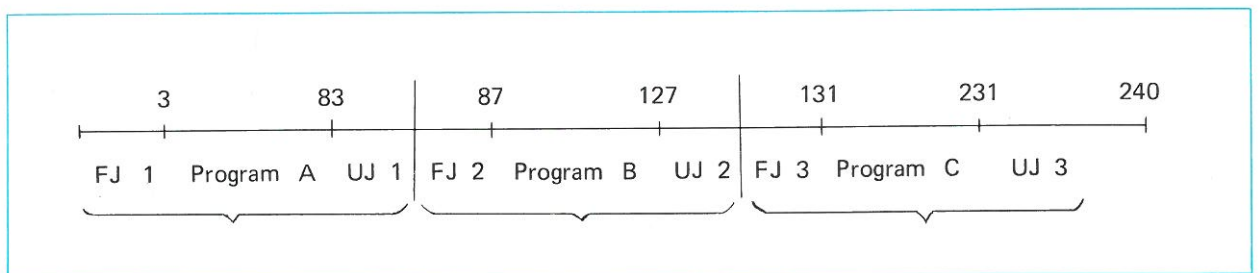
Example : Three independent programs

- Program A 80 steps
- Program B 40 steps
- Program C 100 steps

Operation :



Read-in takes place in the above sequence.



Programs A, B, and C occupy, respectively, steps of #3 through #82, #87 through #126, and #131 through #230.

### Execution of Program A

In executing Program A, operation is the same as that for ordinary program calculation. At the end of Program A, the UJ instruction at step 83 followed by 1, the UJ symbol, at step 84 returns the program to the steps of the FJ instruction and 1 at steps 1 and 2.

### Execution of Program B

Operation of **UJ** and **2** keys under the manual mode effects an automatic jump to FJ and 2 of steps 85 and 86. Program B is then executed.

At the end of Program B calculation, the program is returned to FJ and 2 at the beginning of Program B by means of UJ and 2 in steps 127 and 128.

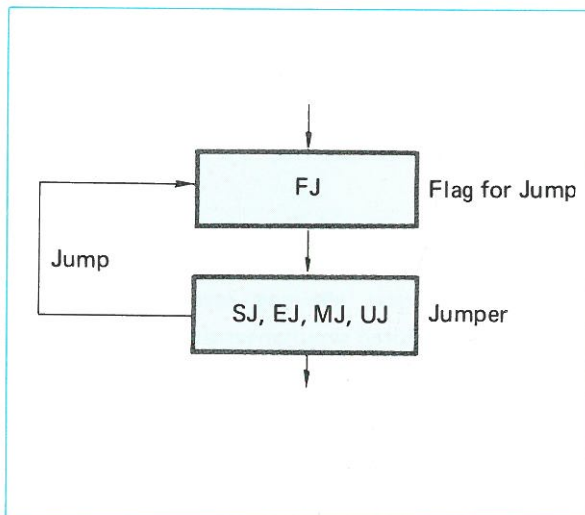
### Execution of Program C

Depressing **UJ** and **3** under the manual mode results in execution similar to that described for programs A and B.

Examples of use of the above mentioned decision functions (jump functions) will be explained in Chapter IV Concept and Program Preparation Methods with actual examples.

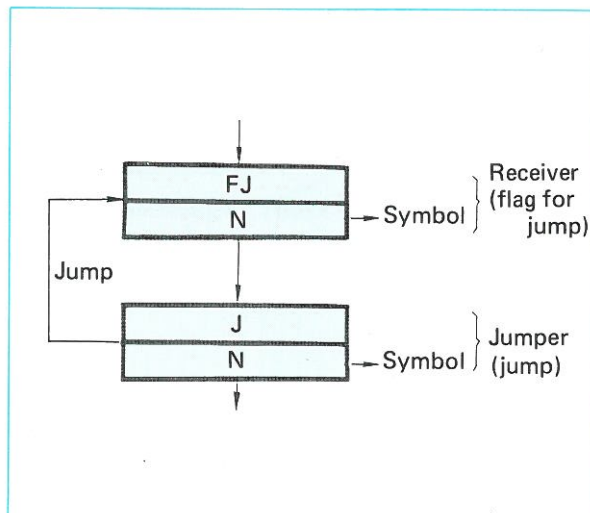
### 3-9-5 Specification of Jump Destination (FJ).

In this calculator, FLAG FOR JUMP instruction is used to specify the destination of jumping when respective jump SJ, EJ, MJ and UJ are to be accomplished, for the purpose of indicating to which particular step the program is caused to jump. That is, jumping is accomplished from the SJ, EJ, MJ and UJ instructions to the FJ instruction.



When several jump instructions are used, each FJ instruction must have its particular symbol relating it to a particular jump instruction.

That is, both jumper and receiver are provided with symbols; the jumper seeks in the program a FLAG FOR JUMP which has a symbol identical to that of the jumper and jumps to its step. Also, because of this action, from the standpoint of hardware the step (one instruction) which immediately follows J (SJ, EJ, MJ and UJ) or FJ is always interpreted as a symbol for jumping.



### 3-9-6 Symbol of Destination of Jumping

Numerals 0–14 are used as symbols for destination of jumping.

In manual system, however, no keys being available which correspond to 10–14 (i.e., if keys  $\boxed{1}$   $\boxed{0}$  are used to represent 10, two steps are required and only 1 is regarded as a symbol with 0 being stored as a constant), symbols are limited to 0 through 9. In the card system 10–14 are specified as binary numbers so that two steps are not required for a symbol.

#### 10–14 by Binary System

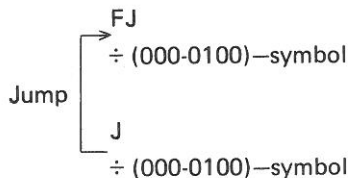
Symbol	Punch hole location	(Decimal)	(Binary)
	CANON PROGRAM CARD		
	40 20 10 8 4 2 1		
10	■ ■ ■ ■ □ ■ □	10	→ 1010
11	■ ■ ■ ■ □ ■ ■	11	→ 1011
12	■ ■ ■ ■ □ ■ □	12	→ 1100
13	■ ■ ■ ■ □ ■ ■	13	→ 1101
14	■ ■ ■ ■ □ ■ ■	14	→ 1110

### 3-9-7 Supplement to Jump Destination Symbols

#### 1) Selection of symbol

As described earlier, the step (one instruction) which follows J or FJ is always interpreted as a symbol for jumping. Consequently, in practice it is possible to use all the combinations of seven punch hole locations (7 bits) on one horizontal line (one column) on a program card as symbols for jump destination. That is,  $2^7 = 128$  kinds of symbols can be considered.

Thus, in practice



the program instruction which is commonly used for division can also be used as a symbol for specification of jump destination.

However, it is not possible to use all of the 128 symbols (kinds), as those which have the following punch hole locations are not used.

Combination of punch hole locations not used as symbols

		CANON PROGRAM CARD							
		40	20	10	8	4	2	1	
1	□	□	□	□	□	□	□	□	Without punch hole locations
2	■	■	■	■	■	■	■	■	
3	□	□	□	■	■	■	■	■	
4	□	□	■	■	■	■	■	■	When 8, 4, 2 and 1 are punched
5	□	■	□	■	■	■	■	■	
6	□	■	■	■	■	■	■	■	
7	■	□	□	■	■	■	■	■	
8	■	□	■	■	■	■	■	■	Combination punch of 10 and 8, 4, 2, 1
9	■	■	□	■	■	■	■	■	
10	□	□	■	□	□	□	□	□	
11	□	□	■	□	□	□	□	■	
12	□	□	■	□	□	□	■	□	
13	□	□	■	□	□	■	■	■	
14	□	□	■	□	■	□	□	□	
15	□	□	■	□	■	□	■	■	
16	□	□	■	□	■	■	□	□	
17	□	□	■	□	■	■	■	□	
18	□	□	■	□	□	□	□	□	
19	□	□	■	□	□	□	□	■	
20	□	□	■	□	□	■	□	□	
21	□	□	■	□	□	■	■	■	
22	□	□	■	□	■	■	□	□	
23	□	□	■	□	■	■	■	■	
24	□	□	■	□	■	■	■	□	

Considering such conditions, actual usable symbols come to only 104.

Among those which are not naturally used as program instructions, there are some which do not have corresponding codes. Care should be exercised on this point when checking the program.

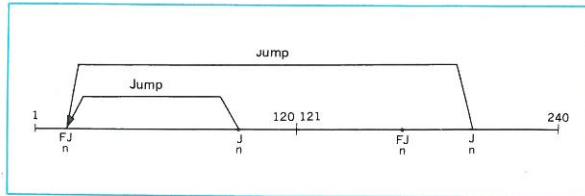
2) Jump when no receiver (FLAG FOR JUMP) is incorporated.

As described previously, jumping in a program is accomplished in such a manner that the jumper with a symbol at the succeeding step seeks a FLAG FOR JUMP which has the same symbol at the succeeding step, but there may be an instance where a jumper is incorporated in a program but no corresponding

FLAG FOR JUMP is available.

In such a case, the program is caused to jump to the 1st or 121st step as specified by the Program Selector Switch.

Consequently, if a jump is required only to the 1st or 121st step, the FJ instruction and the succeeding symbol (receiver) may be omitted.



Consequently, where two programs A and B are simultaneously stored, care should be exercised to avoid using a receiver (FLAG FOR JUMP) identical to that of program A stored in I position, when making program B which is to be stored in II position.

### 3-10 Subroutine Function

#### 3-10-1 Subroutine and Main Routine

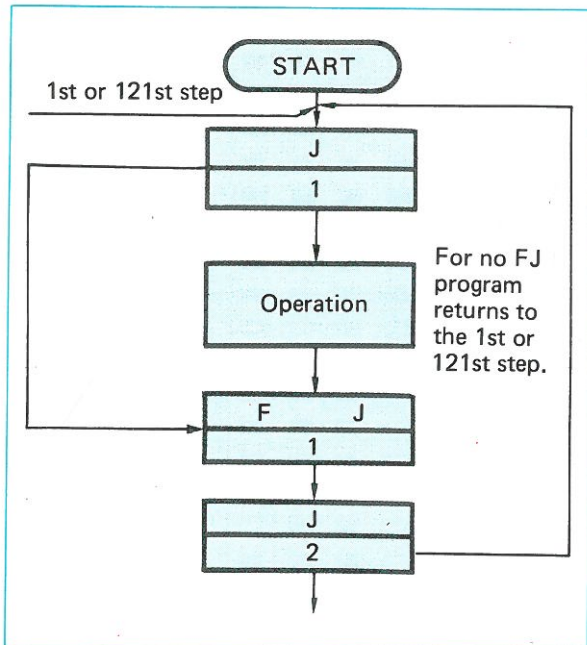
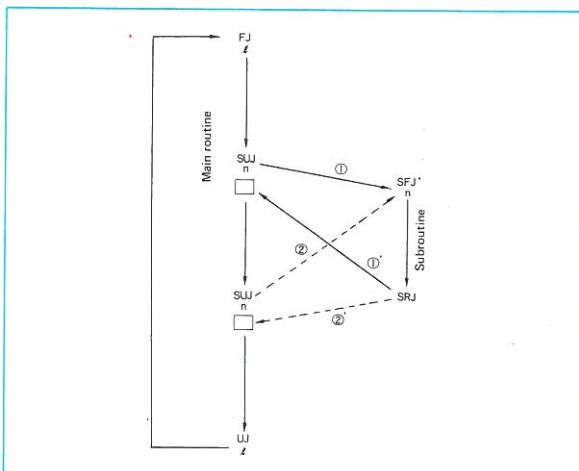
Commonly used operations, which can be repeated within a program, are known as subroutines and are separated from the main routine. Model 1614P provides for three types of subroutine instructions :

SUJ (SUBROUTINE UNCONDITIONAL JUMP)

SFJ (SUBROUTINE FLAG FOR JUMP)

SRJ (SUBROUTINE RETURN JUMP)

SUJ is an unconditional jump from main routine to subroutines. When the program arrives at this instruction, a jump to SFJ at the initial step of subroutine, with corresponding symbol  $n$  occurs. SRJ is inserted at the end of the subroutine to effect return to the main routine. When the subroutine arrives at SRJ, execution of the main routine resumes at the step following the SUJ and its symbol.



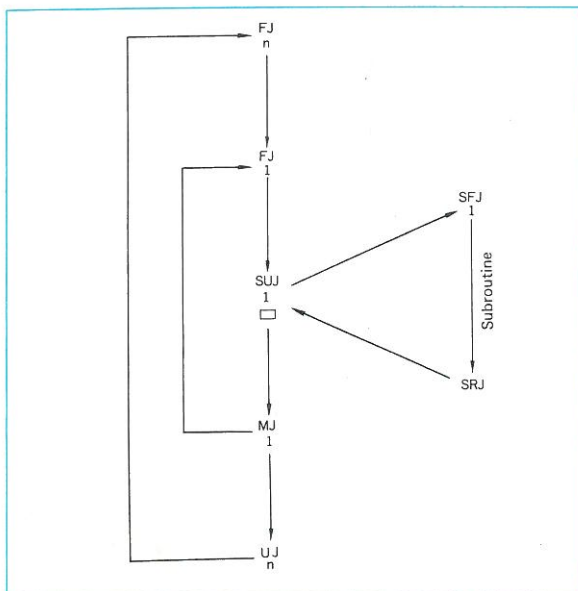
3) Jump when several receivers (FLAG FOR JUMP) are incorporated.

It has previously been explained that a jump in a program is accomplished between a jumper (J) and a flag for jump (FJ) having the same symbol, but there may be a case where more than one flag jump are available for a single jumper.

For example, when two programs A and B are simultaneously stored in the calculator and jump flags with the same symbol are used in two programs, jumping is accomplished in the following manner.

If there are several jumps for a single jumper in the range between the 1st step and the 240th step, a jump is accomplished to the smallest step as viewed from the 1st step regardless of the linear sequence of steps or specifications such as the Program Selector Switch.

Jumping from a main routine to a subroutine is effected with symbol n. Jumping for return to the main routine is made by the SRJ instruction. Therefore, a subroutine can be used any number of times. There is no danger of random return to the main routine. As use of the normal jump symbol differs from that of the subroutine jump symbol within the same program, both symbols may be employed for a subroutine jump without concern. There is no danger of confusion.



An UJ is inserted at the end of the main routine and an FJ at the beginning of the main routine when a subroutine is used, in order to separate the main routine and subroutine. This action returns the program to step 1 following completion of operation of the main routine. When feeding the program card, the sequence of main routine followed by subroutines must be used.

### 3-10-2 Use of the Program Card Package

Four types of prepunched program cards are available, the ordinary subroutine card, the group function card (used in place of the function keys), the statistics card, and the check program card.

#### (1) Mono subroutine card

This card is used in conjunction with the main routine. Subroutines are recalled automatically by a sub-

routine unconditional jump corresponding to a subroutine flag for jump symbol.

#### (2) Group function card

This card is not used to correlate subroutines and other programs. Instead, it is used as a replacement for the function keys.

Depressing the **UJ** key under the manual mode specifies the flag for jump symbol corresponding to the applicable subroutine.

FJ1 sin x UJ1 FJ2 cos x UJ2 FJ3 tan x UJ3

Example : **UJ** 2 → cos x

#### (3) Statistics card

This card accommodates a grouped program for calculation of statistics. It is frequently used in programmed calculation.

#### (4) Check program card

This card is used to check the major functions of the 1614P.

**Mono-Subroutine**

File	Program	SFJ Symbol used	Accuracy	Number of program steps	Number of cards	Memories used
MS-1	$\sin x$	201	10	65	2	14, 13, 12
MS-2	$\cos x$	202	10	65	2	14, 13, 12
MS-3	$\tan x$	203	10	94	3	14, 13, 12, 11, 10, 9
MS-4	$\text{Sin}^{-1} x$	204	10	79	2	14, 13, 12, 11
MS-5	$\text{Cos}^{-1} x$	205	10	82	3	14, 13, 12, 11
MS-6	$\text{Tan}^{-1} x$	206	10	83	3	14, 13, 12, 11
MS-7	$a^n$	207	10	54	2	14, 13, 12
MS-8	$a^x$	208	10	71	2	14, 13, 12, 11
MS-9	$10^x$	209	10	45	2	14, 13, 12, 11
MS-10	$\log_{10} x$	210	10	58	2	14, 13, 12, 11
MS-11	$e^x$	211	10	39	1	14, 13, 12, 11
MS-12	$\log_e x$	212	10	44	2	14, 13, 12, 11
MS-13	$\sinh x$	213	10	38	1	14, 13, 12
MS-14	$\cosh x$	214	10	31	1	14, 13, 12
MS-15	$\tanh x$	301	10	74	2	14, 13, 12, 11, 10
MS-16	$\sinh^{-1} x$	302	10	68	2	14, 13, 12, 11, 10
MS-17	$\cosh^{-1} x$	303	10	49	2	14, 13, 12, 11
MS-18	$\tanh^{-1} x$	304	10	68	2	14, 13, 12, 11, 10
MS-19	Polar $\rightarrow$ Rectangular	305	10	95	3	14, 13, 12, 11, 10, 9
MS-20	Rectangular $\rightarrow$ Polar	306	10	151	4	14, 13, 12, 11, 10, 9
MS-21	Degree, Minute, Second $\rightarrow$ Degree	307		18	1	14, 13, 12
MS-22	Degree $\rightarrow$ Degree, Minute, Second	308		39	1	14, 13
MS-23	Degree $\rightarrow$ Radian	309		21	1	
MS-24	Radian $\rightarrow$ Degree	310		22	1	
MS-25	N!	311	16	22	1	14, 13, 12

FJ Symbol codes used	Input/Output Condition
200	$-10^{16} < x < 10^{16}$ (x : Degree) Input
201	$-10^{16} < x < 10^{16}$ (x : Degree) Input
202, 203, 204	$-10^{16} < x < 10^{16}$ (x : Degree) Input
205, 206, 207	Input : $-1 \leq x \leq 1$ Output : $-90^\circ \leq \text{Sin}^{-1} x \leq 90^\circ$
208, 209, 210	Input : $-1 \leq x \leq 1$ Output : $0^\circ \leq \text{Cos}^{-1} x \leq 180^\circ$
211, 212, 213	Input : $-10^7 < x < 10^7$ Output : $-90^\circ < \text{Tan}^{-1} x < 90^\circ$
214, 300, 301, 302, 303	n : integer a : 14th memory n : 13th memory
304, 305	$a > 0$ a : 14th memory x : 13th memory
306	$-16 \leq x \leq 16$
307	$0 < x < 10^{16}$
308	$-32 \leq x \leq 32$
309	$0 < x < 10^{16}$
310	$-32 \leq x \leq 32$
311	$-32 \leq x \leq 32$
312, 313, 314, 400, 401	$-10^{16} < x < 10^{16}$
402, 403, 404	$-10^8 < x < 10^8$
405	$1 \leq x < 10^8$
406, 407, 408	$-1 < x < 1$
409, 410, 411	$\left\{ \begin{array}{l} \text{Input } r : 14\text{th memory } \theta : 13\text{th memory } (-10^3 \leq \theta \leq 10^3) \\ \text{Output } x : 14\text{th memory } y : 13\text{th memory} \end{array} \right.$
412, 413, 414, 500, 501, 502, 503, 504	$\left\{ \begin{array}{l} \text{Input } x : 14\text{th memory } y : 13\text{th memory} \\ \text{Output } r : 14\text{th memory } \theta : 13\text{th memory } (0^\circ \leq \theta < 360^\circ) \end{array} \right.$
	Input Degree : 14th memory Minute : 13th memory Second : 12th memory
	Print instruction included
505	$0 \leq N \leq 18$

### Group function program

File	Program	Number of cards program + data	Operation	Input	Output
GF-1	$\sin x, \cos x, \tan x, \pi$	4 + 3	UJ 1 : $\sin x$ UJ 2 : $\cos x$ UJ 3 : $\tan x$ UJ 4 : $\pi$	$\left. \begin{array}{l} -10^{16} < x < 10^{16} \\ (x: \text{Degree}) \end{array} \right\}$	Constant
GF-2	$\text{Sin}^{-1}x, \text{Cos}^{-1}x,$ $\text{Tan}^{-1}x, \pi$	6 + 3	UJ 1 : $\text{Sin}^{-1}x$ UJ 2 : $\text{Cos}^{-1}x$ UJ 3 : $\text{Tan}^{-1}x$ UJ 4 : $\pi$	$\left. \begin{array}{l} -1 \leq x \leq 1 \\ -10^{15} \leq x \leq 10^{15} \end{array} \right\}$	$-90^\circ \leq \text{Sin}^{-1}x \leq 90^\circ$ $0^\circ \leq \text{Cos}^{-1}x \leq 180^\circ$ $-90^\circ < \text{Tan}^{-1}x < 90^\circ$ Constant
GF-3	$\sinh x, \cosh x, \tanh x,$ $e^x$	6 + 3	JU 1 : $\sinh x$ JU 2 : $\cosh x$ UJ 3 : $\tanh x$ UJ 4 : $e^x$	$\left. \begin{array}{l} -32 \leq x \leq 32 \\ -10^{15} \leq x \leq 10^{15} \\ -32 \leq x \leq 32 \end{array} \right\}$	
GF-4	$\sinh^{-1}x, \cosh^{-1}x,$ $\tanh^{-1}x, \log_e x$	6 + 3	UJ 1 : $\sinh^{-1}x$ UJ 2 : $\cosh^{-1}x$ UJ 3 : $\tanh^{-1}x$ UJ 4 : $\log_e x$	$\begin{array}{l} -10^{15} \leq x \leq 10^{15} \\ 1 \leq x \leq 10^{15} \\ -1 < x < 1 \\ 0 < x < 10^{16} \end{array}$	
GF-5	$\log_{10}x, \log_e x, 10^x, e^x$	6 + 4	UJ 1 : $\log_{10}x$ UJ 2 : $\log_e x$ UJ 3 : $10^x$ UJ 4 : $e^x$	$\begin{array}{l} 0 < x < 10^{16} \\ 0 < x < 10^{16} \\ -15 \leq x \leq 15 \\ -32 \leq x \leq 32 \end{array}$	
GF-6	$a^x, \sqrt[3]{x}$	6 + 4	UJ 1 : $a^x$ UJ 2 : $\sqrt[3]{x}$	$\begin{array}{l} a > 0 \\ 0 \leq x < 10^{16} \end{array}$	



### Statistic program

File	Program	Number of program steps	Number of cards	FJ symbol codes used	Memories used
ST-1	Mean, Variance, Standard Deviation	40	1	610	14, 13, 12
ST-2	Linear Regression	79	2	611, 612	14, 13, 12, 11, 10, 9
ST-3	Correlation Coefficient	62	2	613	14, 13, 12, 11, 10, 9, 8
ST-4	Summations of Palred Data $(x_i, y_i)$ $n, \sum x,$ $\sum x^2, \sum y, \sum y^2, \sum xy$	40	1	614	14, 13, 12, 11, 10, 9, 8

### Check program

### 3-11 Card System

This is a system designed to facilitate program storage ; a program once developed is translated into codes, transferred (punched) on to Canon program cards, and then into the calculator storage through the card-reader which reads the punched codes.

This program storage capability greatly enhances and expands calculation ability. Once frequently occurring calculations are programmed and punched on cards, they can be filed in a program card library for immediate use. Programs are stored in the 1614P simply by the insertion of these cards into the card-reader, and the answers can be quite simply obtained by only keyboard entry of variables and depression of the  and  keys.

### 3-12 Canon Program Card

This is generally known as "PUNCHED CARD". The punch hole locations are preformed to facilitate punching operation with simple tools and does not require an elaborate punching machine.

Accordingly, the Canon program card can be used freely as a means of program input.

A card comprises seven punch hole locations (7 bits) in one horizontal line (one column), and so one program instruction is entered by the combination of these seven punch hole locations.

Seven punch holes weighted as 40, 20, 18, 8, 4, 2 and 1 starting from the left, and 95 kinds of program instructions may be represented by combinations of these 7 holes.

The three left side locations (headed 40, 20, 10) are for the set program instruction codes. The four right side locations (8, 4, 2, 1) are for individual program codes.

(See Program Instruction Card Codes for details).

One card has forty rows of punch hole locations (each with 7 punch hole locations) longitudinally. This means it is possible to write 40 program instructions (40 program steps) in one card.

When one program exceeds 40 steps, write the 41st step and subsequent steps on another card sequentially from the 1st step.

When one program exceeds 80 steps, three cards shall be used in the same manner and, accordingly four cards shall be used if a program exceeds 120 steps.

To store this program in the calculator, insert these cards one after another in correct sequence starting from 1 into the card-reader.

### 3-13 Cross Reference of Program Instruction and Code

In the card system the programmed program is transferred to cards.

For this purpose it is necessary to once convert the program represented by program instructions into codes.

For the 1614P the corresponding relationship between program instructions and codes is shown in a table and classified into groups to facilitate understanding.

### 3-13-1 Program Instruction ↔ Code Cross Reference Table

		00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
F	00		±	≡	×	÷	√									
J	10		ENT	SJ	EJ		MJ	UJ	FJ	SUJ	SFJ	SRJ				
M	20		M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	M <sub>5</sub>	M <sub>6</sub>	M <sub>7</sub>	M <sub>8</sub>	M <sub>9</sub>	M <sub>10</sub>	M <sub>11</sub>	M <sub>12</sub>	M <sub>13</sub>	M <sub>14</sub>
M SM	30		M <sub>1</sub>	M <sub>2</sub>	SM <sub>3</sub>	SM <sub>4</sub>	SM <sub>5</sub>	SM <sub>6</sub>	SM <sub>7</sub>	SM <sub>8</sub>	SM <sub>9</sub>	SM <sub>10</sub>	SM <sub>11</sub>	SM <sub>12</sub>	SM <sub>13</sub>	SM <sub>14</sub>
RM	40		RM <sub>1</sub>	RM <sub>2</sub>	RM <sub>3</sub>	RM <sub>4</sub>	RM <sub>5</sub>	RM <sub>6</sub>	RM <sub>7</sub>	RM <sub>8</sub>	RM <sub>9</sub>	RM <sub>10</sub>	RM <sub>11</sub>	RM <sub>12</sub>	RM <sub>13</sub>	RM <sub>14</sub>
C	50	CI	CM <sub>1</sub>	CM <sub>2</sub>	CM <sub>3</sub>	CM <sub>4</sub>	CM <sub>5</sub>	CM <sub>6</sub>	CM <sub>7</sub>	CM <sub>8</sub>	CM <sub>9</sub>	CM <sub>10</sub>	CM <sub>11</sub>	CM <sub>12</sub>	CM <sub>13</sub>	CM <sub>14</sub>
PRS	60	.	RV	→	SIGN CHG		↙	5/4	↖	◇	FD					
N	70	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Numerals 10 through 14 are used as jump instruction symbols. These cannot be regarded as constants.

### 3-13-2 How to Read Code Table & Explanation of Contents

It has already been established that the set instruction codes (hereinafter SIC) of program instruction are represented by the combination of three codes on the left side of the program card, i.e., 40, 20, 10 ; the individual program codes (hereinafter IPC) are represented by the combination of four codes on the right side, i.e., 8, 4, 2, 1,

When reading the code for a respective program instruction, the set instruction codes and the individual program codes which correspond to the subject instruction shall be added. For example M<sub>3</sub> is a sum of SIC 20 and IPC 03. That is 20 + 03 = 203. The corresponding relationship between program instruction and code is shown below as classified by groups. Become fully acquainted with this relationship.

- 1) 00 (Four rules of arithmetic operation and square root calculation instructions : FUNCTION)

001	±	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
002	≡	2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
003	X	3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
004	÷	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
005	√	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- 2) 10 (Jump, entry, and stop instructions : JUMP)

101	ENT	6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
102	SJ	7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
103	EJ	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
105	MJ	9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
106	UJ	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
107	FJ	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
108	SUJ	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
109	SFJ	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
110	SRJ	14	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3) 20 (Plus memory instructions : PLUS MEMORY)

201	M <sub>1</sub>	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
202	M <sub>2</sub>	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
203	M <sub>3</sub>	3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
204	M <sub>4</sub>	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
205	M <sub>5</sub>	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
206	M <sub>6</sub>	6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
207	M <sub>7</sub>	7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
208	M <sub>8</sub>	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
209	M <sub>9</sub>	9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
210	M <sub>10</sub>	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
211	M <sub>11</sub>	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
212	M <sub>12</sub>	12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
213	M <sub>13</sub>	13	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
214	M <sub>14</sub>	14	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

4) 30 (Minus memory, and store memory instructions : MINUS MEMORY & STORE MEMORY)

301	M <sub>1</sub>	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
302	M <sub>2</sub>	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
303	SM <sub>3</sub>	3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
304	SM <sub>4</sub>	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
305	SM <sub>5</sub>	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
306	SM <sub>6</sub>	6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
307	SM <sub>7</sub>	7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
308	SM <sub>8</sub>	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
309	SM <sub>9</sub>	9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
310	SM <sub>10</sub>	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
311	SM <sub>11</sub>	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
312	SM <sub>12</sub>	12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
313	SM <sub>13</sub>	13	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
314	SM <sub>14</sub>	14	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

5) 40 (Recall memory instructions : RECALL MEMORY)

401	RM <sub>1</sub>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
402	RM <sub>2</sub>	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
403	RM <sub>3</sub>	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
404	RM <sub>4</sub>	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
405	RM <sub>5</sub>	5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
406	RM <sub>6</sub>	6	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
407	RM <sub>7</sub>	7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
408	RM <sub>8</sub>	8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
409	RM <sub>9</sub>	9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
410	RM <sub>10</sub>	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
411	RM <sub>11</sub>	11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
412	RM <sub>12</sub>	12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
413	RM <sub>13</sub>	13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
414	RM <sub>14</sub>	14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

6) 50 (Clear instructions : CLEAR)

500	CI	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
501	CM <sub>1</sub>	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
502	CM <sub>2</sub>	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
503	CM <sub>3</sub>	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
504	CM <sub>4</sub>	5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
505	CM <sub>5</sub>	6	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
506	CM <sub>6</sub>	7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
507	CM <sub>7</sub>	8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
508	CM <sub>8</sub>	9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
509	CM <sub>9</sub>	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
510	CM <sub>10</sub>	11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
511	CM <sub>11</sub>	12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
512	CM <sub>12</sub>	13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
513	CM <sub>13</sub>	14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
514	CM <sub>14</sub>	15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

7) 60 (Other instructions: POINT, REVERSE, SHIFT & SIGN CHANGE)

600	.	1	■	■	□	□	□	□	□
601	RV	2	■	■	□	□	□	□	■
602	→	3	■	■	□	□	□	■	□
603	SIGN CHG	4	■	■	□	□	□	■	■
605	↵	5	■	■	□	□	■	□	■
606	5/4	6	■	■	□	□	■	■	□
607	↶	7	■	■	□	□	■	■	■
608	◇	8	■	■	□	■	□	□	□
609	FD	9	■	■	□	■	□	□	■

that when entry of data is made through the keyboard.

(Ex.) To enter 348 as data by a card.

		CANON PROGRAM CARD							
STEP	40	20	10	8	4	2	1		
1	■	■	■	□	□	■	■	3	
2	■	■	■	□	■	□	□	4	
3	■	■	■	■	□	□	□	8	

When this card is read by the card-reader, 348 is entered as data and appears on the indicator.

3-14-2 Input of Data into Memories

Instead of storing data into the memories by means of the numeric entry keys it is possible to store data into any memory as required using the program card on which data and memory instructions for the data have been punched.

8) 70 (Value : NUMERAL)

700	0	10	■	■	■	□	□	□	□
701	1	11	■	■	■	□	□	□	■
702	2	12	■	■	■	□	□	■	□
703	3	13	■	■	■	□	□	■	■
704	4	14	■	■	■	□	■	□	□
705	5	15	■	■	■	□	■	□	■
706	6	16	■	■	■	□	■	■	□
707	7	17	■	■	■	□	■	■	■
708	8	18	■	■	■	■	□	□	□
709	9	19	■	■	■	■	□	□	■
710	10	20	■	■	■	■	□	■	□
711	11	21	■	■	■	■	□	■	■
712	12	22	■	■	■	■	■	□	□
713	13	23	■	■	■	■	■	□	■
714	14	24	■	■	■	■	■	■	□

		CANON PROGRAM CARD							
STEP	40	20	10	8	4	2	1		
1	■	■	■	□	□	■	□	2	
2	■	■	■	□	□	■	■	3	
3	□	■	■	□	□	■	■	SM <sub>3</sub>	
4	■	■	■	□	□	□	■	1	
5	■	■	■	□	□	■	□	2	
6	□	■	■	□	■	□	□	SM <sub>4</sub>	

When this card is read by the card reader, 23 is stored in Memory 4.

If the data involves decimal points, care should be exercised using the Decimal Point Selector Dial when storing into Memory 1, Memory 2, and Memories 3-9. For example, if a program card punched as 1.5 M<sub>1</sub> is read with the Decimal Point Selector Dial set as 0, 1.5 is indicated as it is on the indicator, but only 1 is stored in Memory 1.

3-14 Input Data by Card

In the 1614P it is possible to input data as an entry or to input data into the memories by means of program cards (when the Program Mode Switch is set in OPE position).

3-14-1 Data Input as Entry

When the 1614P is in a state of stand-by for entry, instead of entering data through the keyboard, program cards are read into the 1614P by the cardreader.

Thus, entry is accomplished in an identical manner as

\* Difference of Decimal Point Positions in Memories  
Due to M & SM.

When storing values in a memory, remember, the position of the decimal point in the value to be stored varies even in the same memory as to whether M is to be used or SM is used.

That is, when M is used the value indicated is stored in accordance with the position of the decimal point specified by the Decimal Point Selector Dial, and when SM is used the indicated value is stored as it is regardless of the specifications by the Decimal Point Selector Dial.

The best way to explain this capability is through an example.

(Ex.) Store 1 2 3 . 4 6 in Memory 2 using M, and into Memory 3 using SM with the Decimal Point Selector Dial set at 0 and check the contents of the respective memories.

- 1) Operate  $\boxed{1} \boxed{2} \boxed{3} \boxed{\cdot} \boxed{4} \boxed{6} \boxed{M} \boxed{SM} \boxed{3}$  or  $\boxed{SM} \boxed{3} \boxed{M}$  and store 1 2 3 . 4 6 into respective memories.
- 2) Depress  $\boxed{RM} \boxed{2}$  to recall contents of Memory 2. The recalled contents are 1 2 3.
- 3) Depress  $\boxed{RM}$  and  $\boxed{3}$  to recall contents of Memory 3. The recalled contents are 1 2 3 . 4 6.

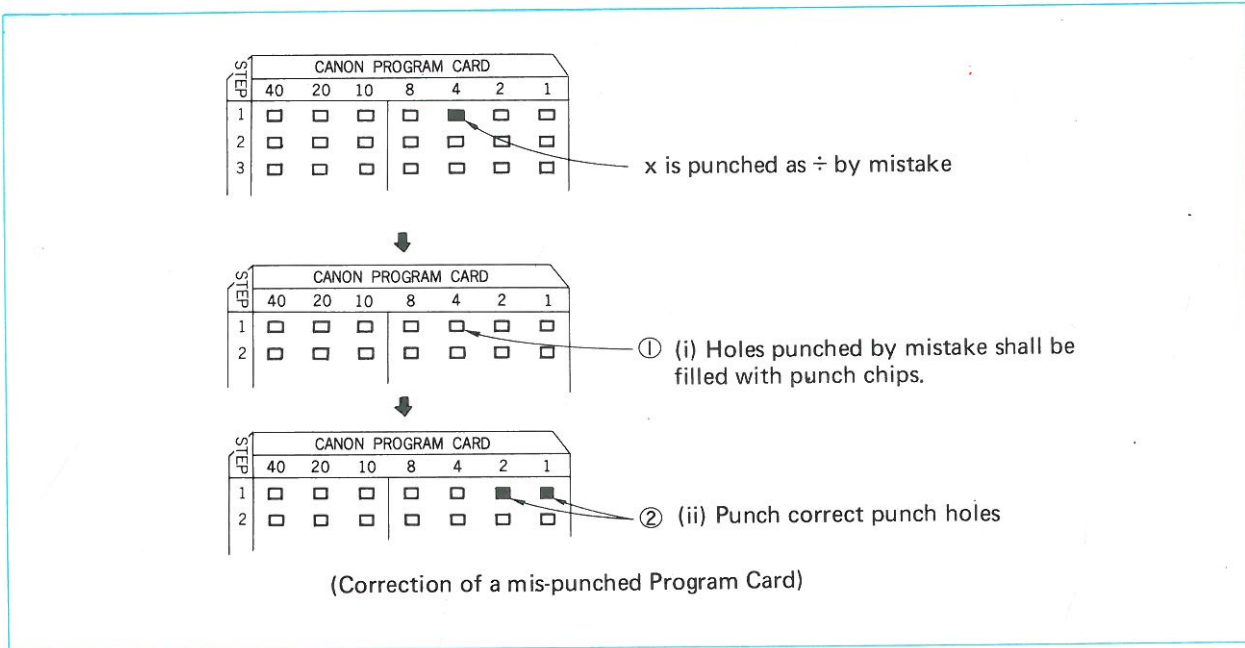
### 3-15 Correction & Clearing of Program Instruction

#### Written into Cards

If a program instruction is punched by mistake on a card, correction and clearing shall be accomplished in the following manner.

#### 1) Correction

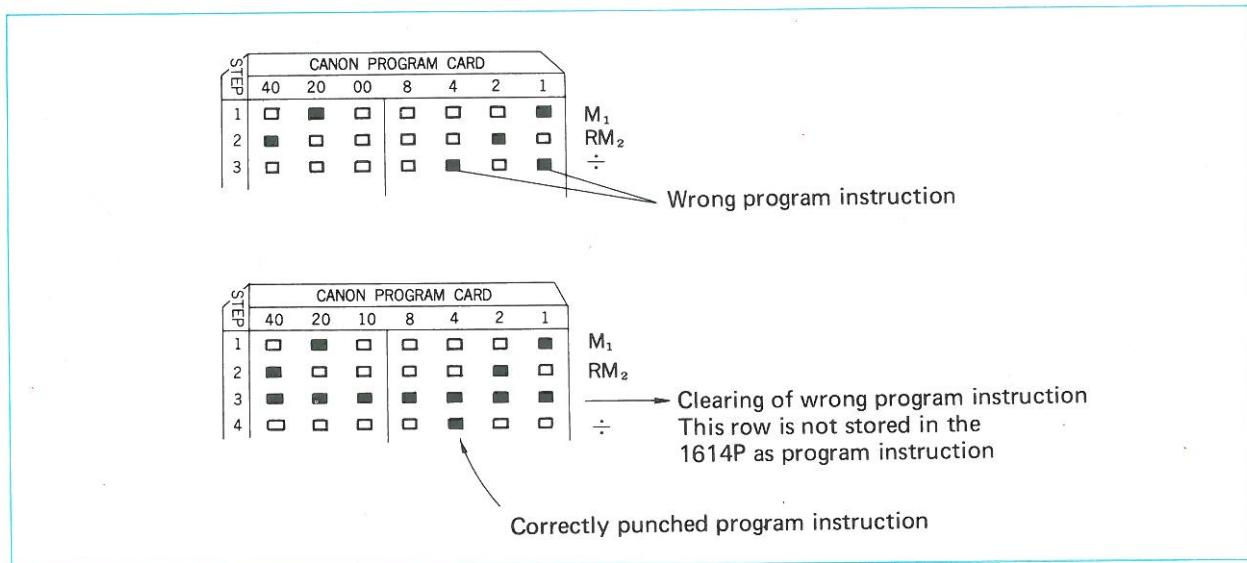
- (i) First, fill the hole punched by mistake with punch chips (punched out pieces).
- (ii) Then, punch correct punch holes.



#### 2) Clearing

It is possible to correct a wrong program instruction in the procedure described above. It is also possible to

clear seven punch hole positions on the whole lateral line in which the wrong program instruction is written.



# 4

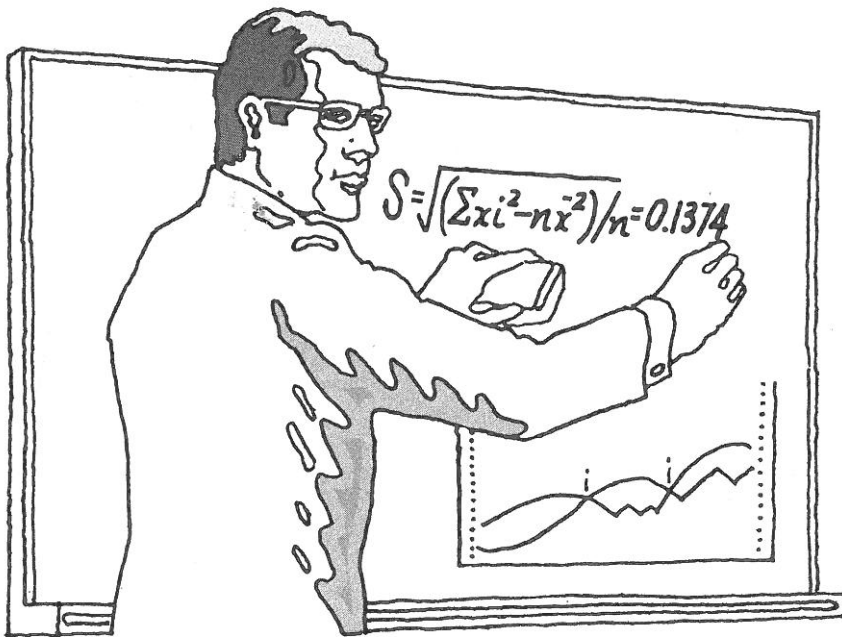
## CONCEPT & PROGRAM PREPARATION METHODS WITH ACTUAL EXAMPLES

1. General
2. Programming
3. Program Using ROUND Instructions  
(Calculation using Pythagorean theorem)
4. Program Using Decision Functions  
(Jump Functions)
  - 4-1 Program Using SENSE JUMP  
(Depreciation calculation)
  - 4-2 Program Using ENTRY JUMP  
(Calculation of Average Unit Price)
  - 4-3 Program Using MINUS JUMP  
(Calculation of a Factorial)
5. Program Using Subroutine  
(Calculation of the Velocity of Flowing Fluid)

## 1. General

The performance and methods of operation required to execute program calculation by using the 1614P have been described.

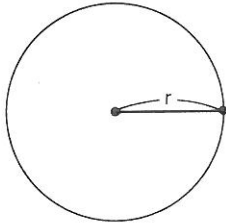
In this chapter we will describe how programs are actually prepared based on preceding explanations of operation, as well as cover how answers are obtained. As a rule, descriptions will be made in accordance with the card system.



## 2. Programming

Let's explain programming with a simple example.

Example : Calculation of the area of a circle  
Find the area when the radius (r) is known.



$$S \text{ (area)} = \pi r^2$$

Where  $\pi = 3.14$

### 1) Modification of equation

First the nominal equation is modified to place the variable ahead of the constant, that is,  $r^2 \pi$ .

### 2) Preparation of flow chart

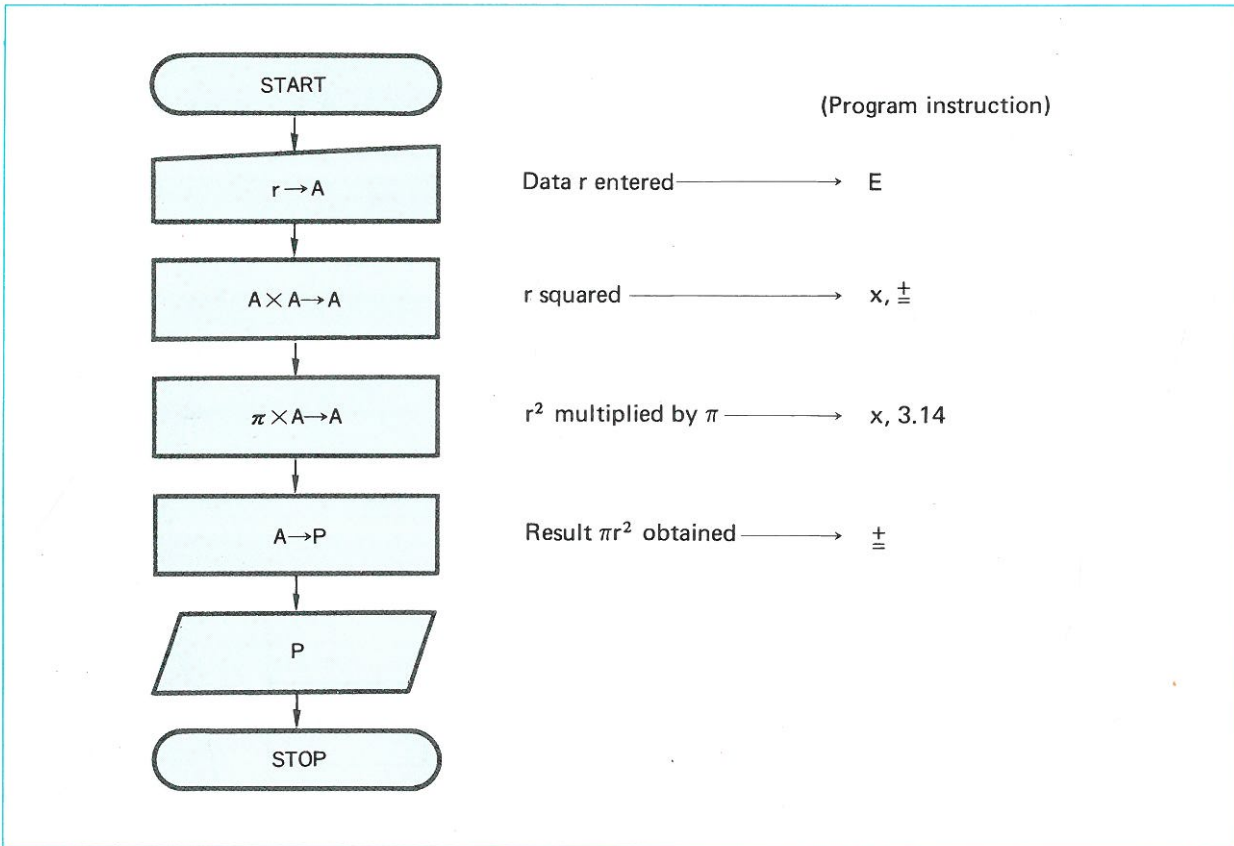
The flow chart is useful as a means of arranging elements of a complex problem in readily understandable form. It shows the problem at a glance.

Symbols used in this manual are shown here.

Symbol	Application and meaning
	Denotes start and end of program (  and  ).
	Denotes data input.
	Denotes data processing.
	Denotes branching to routine. The condition of branching is written inside the figure. (  )
	Denotes output of a computed result.
	Denotes routine connecting terminals. Figures with like designation are connected. (  )
	Denotes direction of program flow.

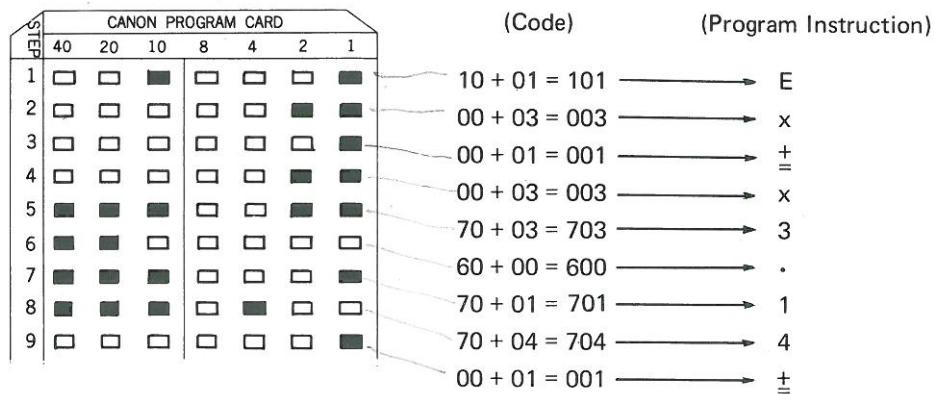
3) Programming by Model 1614P program instruction.

Programming is compiled with program instructions according to the prepared flow chart.





4) Preparation of program card

The completed program must be punched on the Canon program card in accordance with the program instruction and coding table.










5) Program storage and execution of calculation

The first procedure is the establishment of the program area with the program step switches. With the Program Mode Switch set to LRN, program card data can be read in.

Following completion of program card reading, the program mode switch is shifted from LRN to OPE. Subsequent operation of the  key advances the program to step 1 (program calculation area). Data is then entered with the numeral keys and program operation begun with the  key.

Preparation



Procedure	Key operation	Display	Remarks
1		0	Manual calculation area → Program calculation area
2		2	Input
3		12.56	Output
4		3	Input
5		28.26	Output
6		4	Input
7		50.24	Output

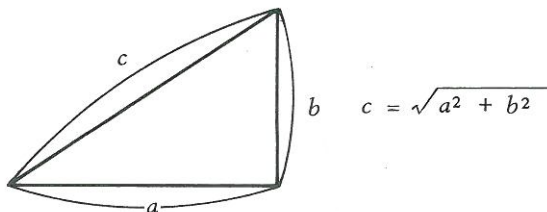
### 3. Program Using ROUND Instruction (Calculation with Pythagorean theorem)

Ordinarily, decimal point specification is accomplished with a dial or switch. This function is handled with the ROUND instruction in the case of program calculation. If the ROUND instruction is used, the decimal point selector should be set to floating, as this increases precision of interim operations.

Let's consider an example of calculation involving the Pythagorean theorem and the ROUND instruction.

Example : Calculation involving Pythagorean theorem

Find the length of hypotenuse  $c$  when sides  $a$  and  $b$  are known.



When  $a = 7$  and  $b = 6$ , find  $c$  rounded off to four places following the decimal point.

Step	Program instruction	Display	Decimal point selector	ROUND instruction	Remarks
1	E	7			Square of $a$ taken and the result stored in memory 1.
2	X	7			
3	$\pm$	49			Square of $b$ taken and added to memory 1, thereby giving the sum of the squares of $a$ and $b$ in this memory.
4	M <sub>1</sub>	49			
5	E	6			
6	X	6			Sum of the squares of $a$ and $b$ called, ROUND instruction specified, and root extracted.
7	$\pm$	36			
8	M <sub>1</sub>	36			Square root of $a^2 + b^2$ , obtained to 4 decimal places.
9	RM <sub>1</sub>	85			
10	$\downarrow$	85		$\downarrow$ Four places following decimal point.	
11	4	85			
12	$\sqrt{\quad}$	9.2195			

Preparation



Procedure	Key operation	Display	Remarks
1		0	Manual calculation area $\rightarrow$ Program calculation area
2		7	Input of $a_1$
3		49	
4		6	Input of $b_1$
5		9.2195	Output of $c_1$
6		3	Input of $a_2$
7		9	
8		4	Input of $b_2$
9		5.0000	Output of $c_2$

#### 4. Program Using Decision Functions (Jump Functions)

Examples of conventional programs have been described, and in this chapter decision functions possessed by the 1614P will be explained in actual examples to enable the reader to understand how decision functions are utilized.

##### 4-1 Program Using SENSE JUMP (Depreciation Calculation)

Calculate the total depreciation while obtaining the depreciation amount and the book value for the respective year.

Year	Purchase unit price \$	Amount of depreciation	Book value
1.	350,000	(129,150)	(220,850)
2.		( 81,494)	(139,356)
3.		( 51,422)	( 87,934)
4.		( 32,448)	( 55,486)
5.		( 20,474)	( 35,012)
		Total (314,988)	

Depreciation is based on the fixed rate method ; Term of depreciation is five years and depreciation rate is 36.9%.

Calculation formula :

Purchase price

(book value for preceding year)  $\times 0.369 =$  amount of depreciation

Purchase price            amount of  
(book value for    —    depreciation    =    book value  
preceding year)    for this year        for this year

##### (Programming)

1) Programming for this example is made in the following sequence.

(i) Enter the purchase price and store it in MEMORY 3. Use SM<sub>3</sub> to eliminate the preparation to depress  $\text{SM } 3$  for execution of program calculation. Enter the depreciation rate and store in MEMORY 4 by SM<sub>4</sub> which also functions to clear the previous memory contents.

(ii) Clear MEMORY 5 in advance by CM<sub>5</sub> as it will be used to obtain the total amount of depreciation.

(iii) Recall the purchase price (book value for the preceding year) and the depreciation rate and obtain the amount of depreciation. Temporarily stop the program and read the value.

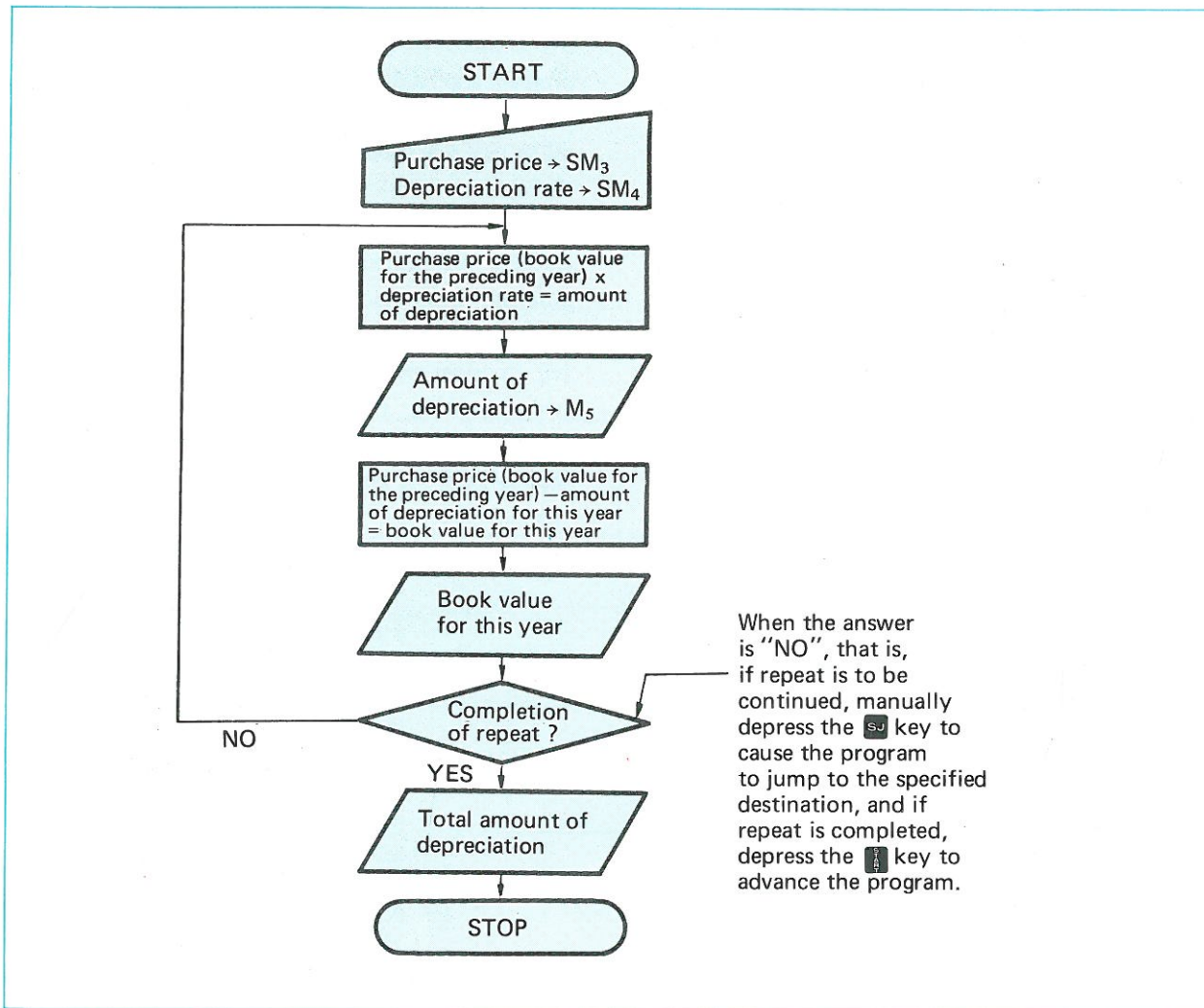
(iv) Store the obtained amount of depreciation in MEMORY 5 and calculate the total of the amounts of depreciation in the memory.

(v) Calculate in MEMORY 3 by subtracting the amount of depreciation for this year from the purchase price (book value for the preceding year), and read the value by recalling the contents.

(vi) Consider repeat of (iii)-(v).

(vii) Repeating (iii)-(v) to the required extent, advance the program ; recall the contents of MEMORY 5 and obtain the total amount of depreciation.

2) Flow Chart of Depreciation Calculation



3) The program is described by program instructions as follows in accordance with the above mentioned procedure.

E, SM<sub>3</sub>, E, SM<sub>4</sub>, CM<sub>5</sub>, FJ, 1, RM<sub>3</sub>, X, RM<sub>4</sub>,  
5/4, 0, ±, S, M<sub>5</sub>, SC, M<sub>3</sub>, RM<sub>3</sub>, SJ, 1, RM<sub>5</sub>,

4) The relationship between punch hole locatins on the card, codes and program instructions when the card system is used is shown in the next page for reference purposes.

CANON PROGRAM CARD							
STEP	40	20	10	8	4	2	1
1	□	□	■	□	□	□	■
2	□	■	■	□	□	■	■
3	□	□	■	□	□	□	■
4	□	■	■	□	■	□	□
5	■	□	■	□	■	□	■
6	□	□	■	□	■	■	■
7	■	■	■	□	□	□	■
8	■	□	□	□	□	■	■
9	□	□	□	□	□	■	■
10	■	□	□	□	■	□	□
11	■	■	□	□	■	■	□
12	■	■	■	□	□	□	□
13	□	□	□	□	□	□	■
14	□	□	■	□	□	□	■
15	□	■	□	□	■	□	■
16	■	■	□	□	□	■	■
17	□	■	□	□	□	■	■
18	■	□	□	□	□	■	■
19	□	□	■	□	□	■	□
20	■	■	■	□	□	□	■
21	■	□	□	□	■	□	■

(Code)	(Program Instruction)
10 + 01 = 101	E
30 + 03 = 303	SM <sub>3</sub>
10 + 01 = 101	E
30 + 04 = 304	SM <sub>4</sub>
50 + 05 = 505	CM <sub>5</sub>
10 + 07 = 107	FJ
70 + 01 = 701	1
40 + 03 = 403	RM <sub>3</sub>
00 + 03 = 003	X
40 + 04 = 404	RM <sub>4</sub>
60 + 06 = 606	5/4
70 + 00 = 700	0
00 + 01 = 001	±
10 + 01 = 101	S
20 + 05 = 205	M <sub>5</sub>
60 + 03 = 603	SC
20 + 03 = 203	M <sub>3</sub>
40 + 03 = 403	RM <sub>3</sub>
10 + 02 = 102	SJ
70 + 01 = 701	1
40 + 05 = 405	RM <sub>5</sub>

Program storage explanations are omitted here, because it is identical to what was described earlier.

**(Execution of Program Calculation)**

Program data is read in. The decimal point selector should be set to floating.

Procedure	Key operation	Display	Remarks
1		0	Manual calculation area → Program calculation area
2		350000	Input of value at time of acquisition
3		350000	
4		0.369	Input of depreciation rate
5		129150	Depreciation during 1st year
6		220850	Book value for 1st year
7		81494	Depreciation during 2nd year
8		139356	Book value for 2nd year
9		51422	Depreciation during 3rd year
10		87934	Book value for 3rd year
11		32448	Depreciation during 4th year
12		55486	Book value for 4th year
13		20474	Depreciation during 5th year
14		35012	Book value for 5th year
15		314988	Total depreciation



#### 4-2 Program Using ENTRY JUMP (Calculation of Average Unit Price)

Calculate the average unit price given respective amounts, total quantity and total of the amounts.

Quantity	Unit price \$	Amount	Average unit price
123	98.70	(12,140.10)	—
36	103.20	( 3,715.20)	—
85	101.50	( 8,627.50)	—
Total (244)		(24,482.80)	(100.34)

Calculation Formula :

Average unit price = total amount ÷ total quantity

#### (Programming)

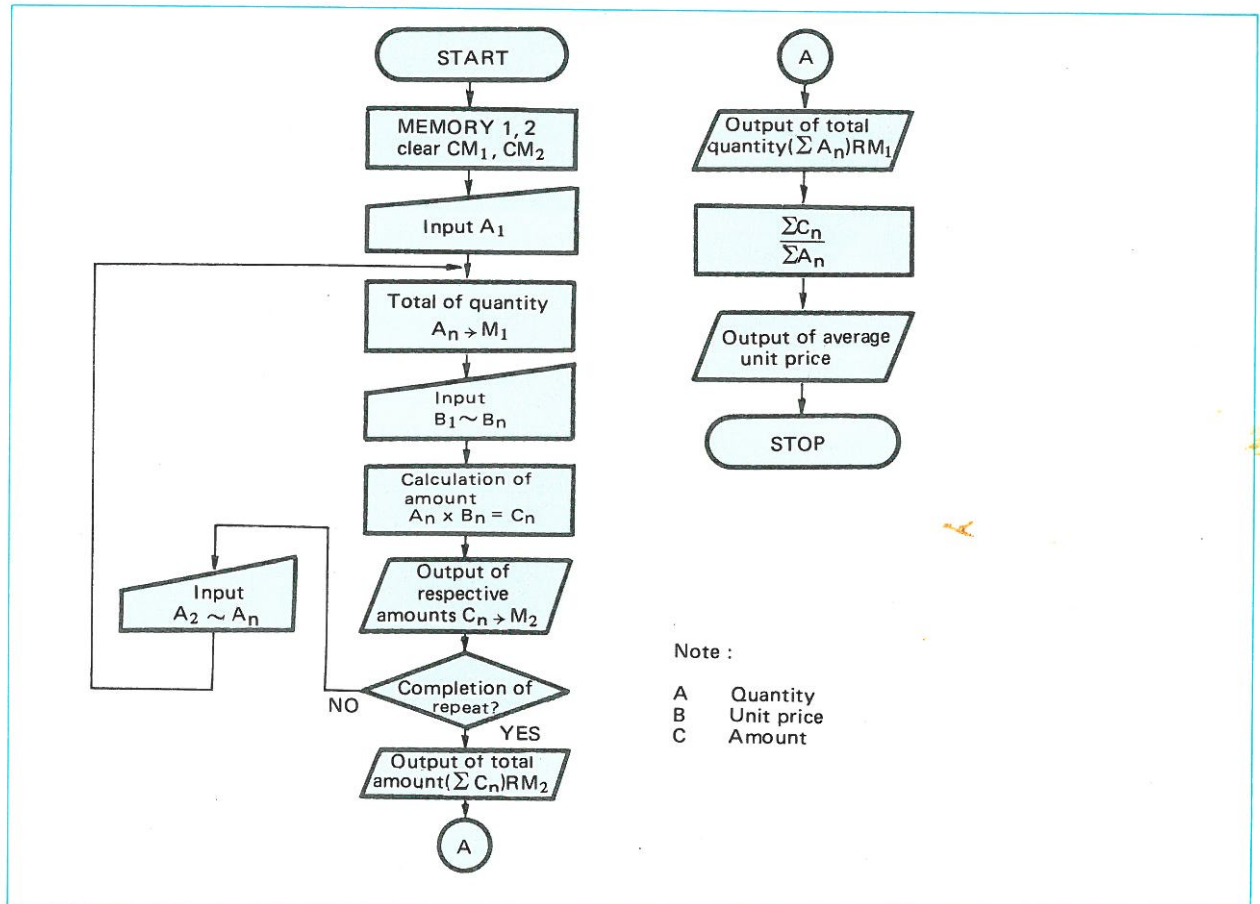
What is Entry Jump to be used for in this example? Explanation is repeated here in brief.

Entry Jump is a jump function that stops the program when it comes across this instruction, subsequently the calculator waits for instruction from the operator ; and if entry and the  $\square$  key operation is accomplished, the program jumps to the specified program step, and if the  $\square$  key only is depressed, it causes the program to execute the succeeding steps without jumping.



For this example, the program is prepared considering that a program jump is accomplished each time entry of a quantity and the  $\square$  key operation ia made.

1) For convenience of description, the outline of procedures will be explained in a flow chart, and then more basically described in correspondence with the program represented by program instructions.

Flow Chart for Calculation of Average Unit Price



2) The relationship between the program which has been prepared in accordance with the flow chart and the procedures are as shown below.

Step	Program instruction	Description of procedure
1	CM <sub>1</sub>	} Clear MEMORY 1 and MEMORY 2 to be used for totalling quantity and amount respectively in advance.
2	CM <sub>2</sub>	
3	E	
4	{ FJ	} Specified destination of jump when quantity is entered and the  key is depressed.
5	{ * 1	
6	M <sub>1</sub>	Total the quantity.
7	X	} Enter unit price. Calculate respective amounts.
8	E	
9	±	
10	M <sub>2</sub>	Total the amounts.
11	{ EJ	} Enter second and subsequent quantities. When entry of quantity is over, cause the program to advance by depressing the  key only.
12	{ * 1	
13	RM <sub>2</sub>	} When repeat of fixed type portion is complete, recall the total of amounts and read it with the progress of program stopped.
14	S	
15	÷	} Recall and read the total of quantity. Calculates the average unit price by dividing the total of by the total of quantity.
16	RM <sub>1</sub>	
17	S	
18	±	











\* In this program 1 is used as the symbol for specification of jump destination.

**(Execution of Program Calculation)**

Execute a practice program calculation using the prepared program.

Procedure for execution is as follows.

Preparation  

Pro- cedure	Key operation	Display	ENTRY lamp	Remarks
1		0		Manual calculation area → Program calculation area
2	1 2 3	123	Lights	First quantity
3		123	Lights	
4	9 8 . 7	98.7	Lights	First unit price
5		(12,140.10)	Lights	First amount
6	3 6	36	Lights	Second quantity
7		36	Lights	
8	1 0 3 . 2	103.2	Lights	Second unit price
9		(3,715.20)	Lights	Second amount
10	8 5	85	Lights	Third quantity
11		85	Lights	
12	1 0 1 . 5	101.5	Lights	Third unit price
13		(8,627.50)	Lights	Third amount
14		(24,482.80)	Lights	Total of amount
15		(244)	Lights	Total of quantity
16		(100.34)	Lights	Average unit price

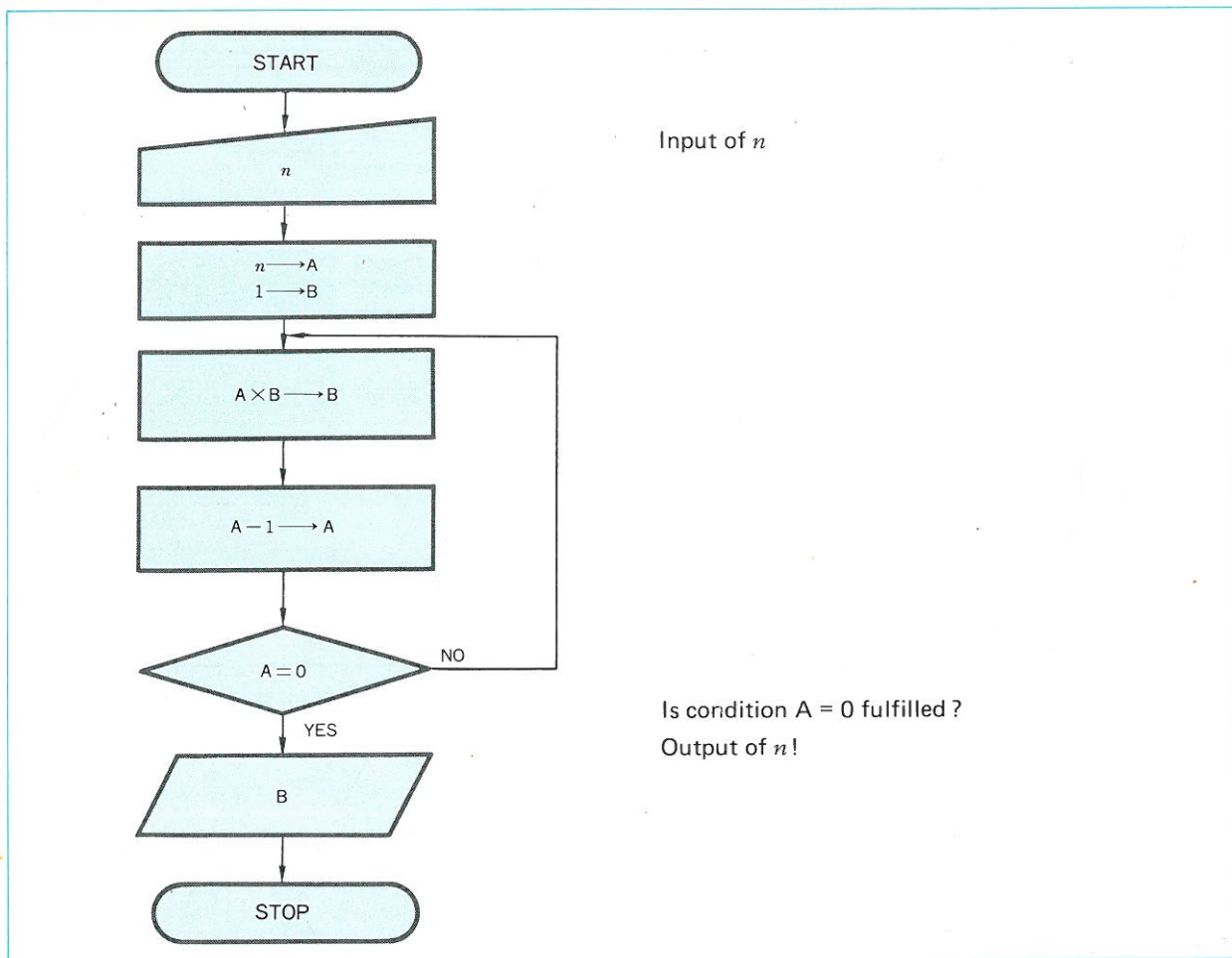
### 4-3 Program Using MINUS Jump (Calculation of a Factorial)

Factorial  $n!$

$$n! = n(n-1)(n-2) \dots 2 \cdot 1$$

Express the product of natural numbers from 1 to  $n$ .

2) Flow chart



Expressing the above flow in program instruction yields the following : E, SM<sub>3</sub>, 1, SM<sub>4</sub>, FJ, 1, RM<sub>3</sub>,

x, RM<sub>4</sub>,  $\frac{+}{-}$ , SM<sub>4</sub>, 1, SC, M<sub>3</sub>, RM<sub>3</sub>, SC, MJ, 1, RM<sub>4</sub>.  
As  $n$  is a natural number, ROUND instruction is not used.

Execution of program calculation

Preparation



Procedure	Key operation	Display	Remarks
1		0	Manual calculation area → Program calculation area
2		4	Input of $n$
3		24	Output of 4!
4		5	Input of $n$
5		120	Output 5!

## 5. Program Using Subroutine (Calculation of the Velocity of Flowing Fluid)

(Calculation of the velocity of flowing fluid)  
Flow calculation (Manning's equation)

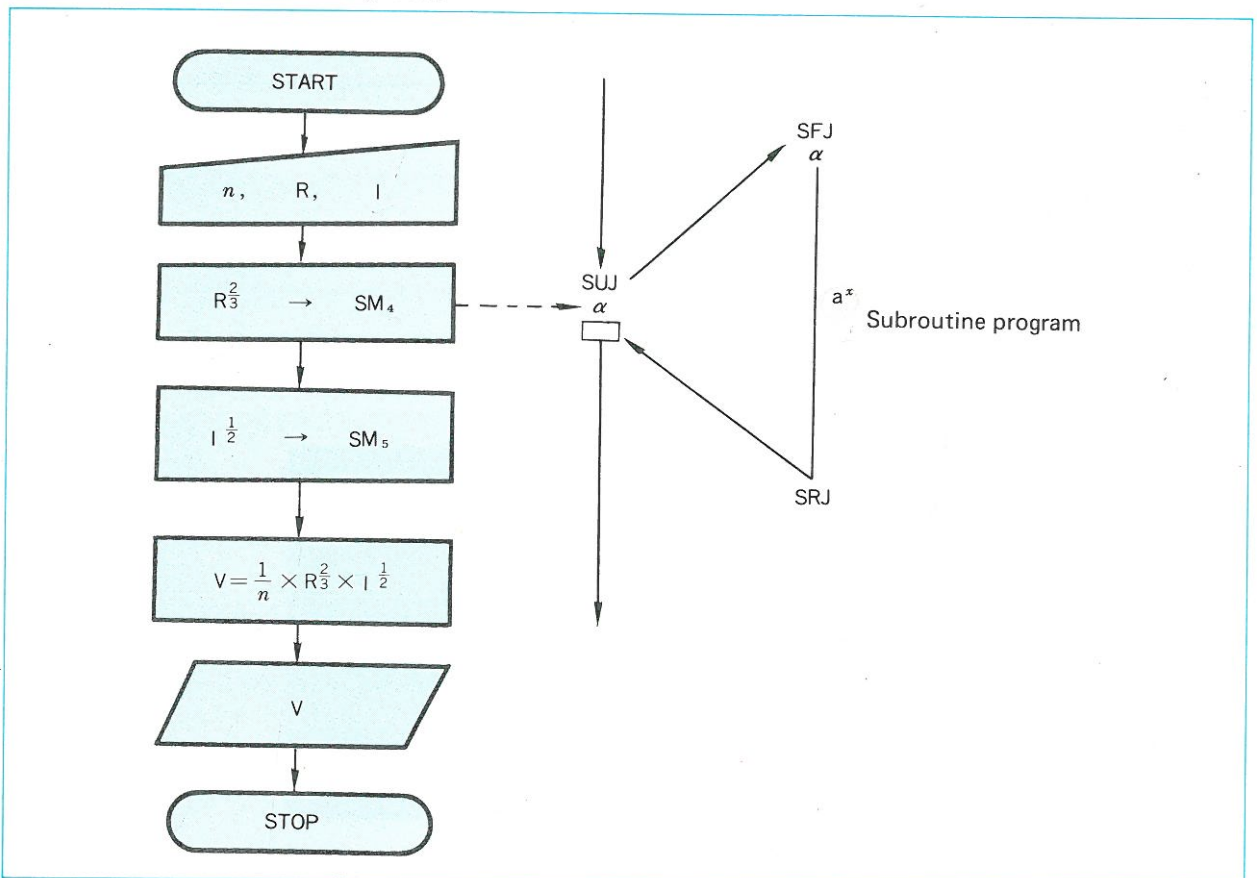
$$\text{Equation : } V = \frac{1}{n} R^{\frac{2}{3}} I^{\frac{1}{2}}$$

V : Rate of flow *velocity*

(Programming)

1) Manning's equation involves no more than multiplication and division, although interim calculation includes square and cube roots. Processing by means of the main routine alone proves difficult and complex. The use of subroutine simplifies the process.

The flow chart takes the following form.



2) Expressing the above procedures in program instruction yields the following.

Main routine: FJ, 1, E(n), SM<sub>3</sub>, E(R), SM<sub>4</sub>, E(I), √, SM<sub>5</sub>, RM<sub>4</sub>, SM<sub>14</sub>, 2, ÷, 3,  $\frac{1}{n}$ , SM<sub>13</sub>, SUJ, (α), X, RM<sub>5</sub>, ÷, RM<sub>3</sub>, 5/4, 1,  $\frac{1}{n}$ , UJ, 1,

Use memories designated by the subroutine card directives for storage of  $a$  and  $x$  values in the expression  $a^x$ , as well as the specified subroutine jump symbol code (α).

(Execution of Program Calculation)

Preparation



Procedure	Key operation	Display	Remarks
1		0	Manual calculation area → Program calculation area
2		30	Input of $n$
3		30	
4		0.747	Input of R
5		0.747	
6		50	Input of I
7		0.2	Output of V

As described under information on subroutines, the main routine must be identified with symbols FJ and UJ, in order to separate the main routine and subroutine. Program cards are fed in the sequence of main routine followed by subroutine programs.

**In conclusion**

This completes the description of your new Canola 1614P Calculator and associated programmed calculations.

Please refer to the instruction manual for the Canon Printer Unit P-1, as this manual gives additional and detailed information on optional instructions. If questions or problems arise, please contact your dealer or other Canon representatives.

# Canon

**CANON INC.**

9-9, Ginza 5-chome, Chuo-ku, Tokyo 104, Japan

**CANON U.S.A., INC.**

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 OPTICS & BUSINESS MACHINES CO., INC.**

3113 Wilshire Blvd., Los Angeles, California 90010, U.S.A.

**CANON AMSTERDAM N.V.**

Gebouw 70, Schiphol Oost, Holland

**CANON LATIN AMERICA, INC.**

Apartado 7022, Panama 5, Panama