

Electronic Calculator



1614P

Canon **Canola 1614P**

Instructions

English Edition

TO CANOLA 1614P USERS :

Thank you for selecting Canon's CANOLA 1614P Electronic Desk-top Programmable Calculator.

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

This instruction manual has been prepared to provide a reference to the basic calculating and programming logic of the 1614P. Its easily comprehensible 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.

Use your handy CANOLA 1614P programmable calculator to the fullest.

March 1972

CANON INC.

CONTENTS

1. Specifications 5-16

- 1. Specifications of Canola 1614P 6
- 2. Explanation of Keys 8
 - 2-1 Explanation of Keys 8
 - 2-2 Explanation of Switches 11
 - 2-3 Explanation of Lamps 11
 - 2-4 Explanation of Program Keys 11
 - 2-5 Explanation of Program Switches 12
 - 2-6 Explanation of Program Lamp 14

2. Explanation as an Ordinary

Calculator 17-34

- 1. General 18
- 2. Calculation Examples 19
 - 2-1 Addition and Subtraction 19
 - 2-2 Multiplication 20
 - 2-3 Division 22
 - 2-4 Mixed Calculation 22
 - 2-5 Constant Multiplication and Division 23
 - 2-6 Power Calculation 24
 - 2-6-1 Power Calculation 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

3. Explanation as a Programmable

Calculator 35-64

- 1. General 36
- 2. Before Commencing 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

3-10 Subroutine Function 54

3-11 Card System 59

3-12 Canon Program Card 59

3-13 Cross Reference of

Program Instruction and Cod 59

3-14 Input Data by Card 62

3-15 Correction & Clearing of Program

Instructions Written into Cards 64

4. Concept & Program Preparation Methods

With Actual Examples 65-80

- 1. General 66
- 2. Programming 67
- 3. Program Using ROUND Instruction (Calculation with Pythagorean theorem) 70
- 4. Program Using Decision Functions (Jump Functions) 71
 - 4-1 Program Using SENSE JUMP (Depreciation calculation) 71
 - 4-2 Program Using ENTRY JUMP (Calculation of Average Unit Price) 75
 - 4-3 Program Using MINUS JUMP (Calculation of a Factorial) 78
- 5. Program Using Subroutine (Calculation of the Velocity of a Flowing Fluid) 79

1

SPECIFICATIONS

1. Specifications of Canola 1614P

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 OF CANOLA 1614P

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 : May 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 ; may 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 may be accomplished.

- 7) Operation check

Step-by-step 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  key ; Automatic calculation of sums of products and quotients ;  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, etc. :

- 1) Size 310W x 395D x 153H (mm)
(1'3/16" x 1'3-9/16" x 6")
- 2) Weight 7.0 kg (15 lbs. 7 oz)
- 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

1-9 Numeric keys

Entry is accomplished as random keys are depressed one after another. For example, if numeric keys 2 3 4 are depressed in sequence, the 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.



Shift Key

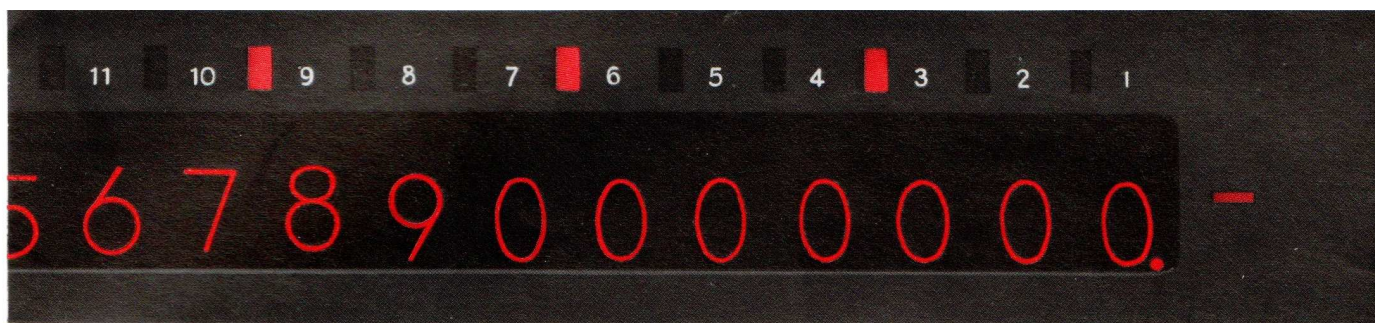
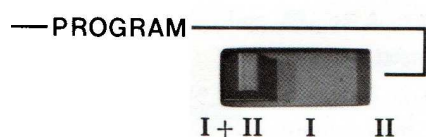
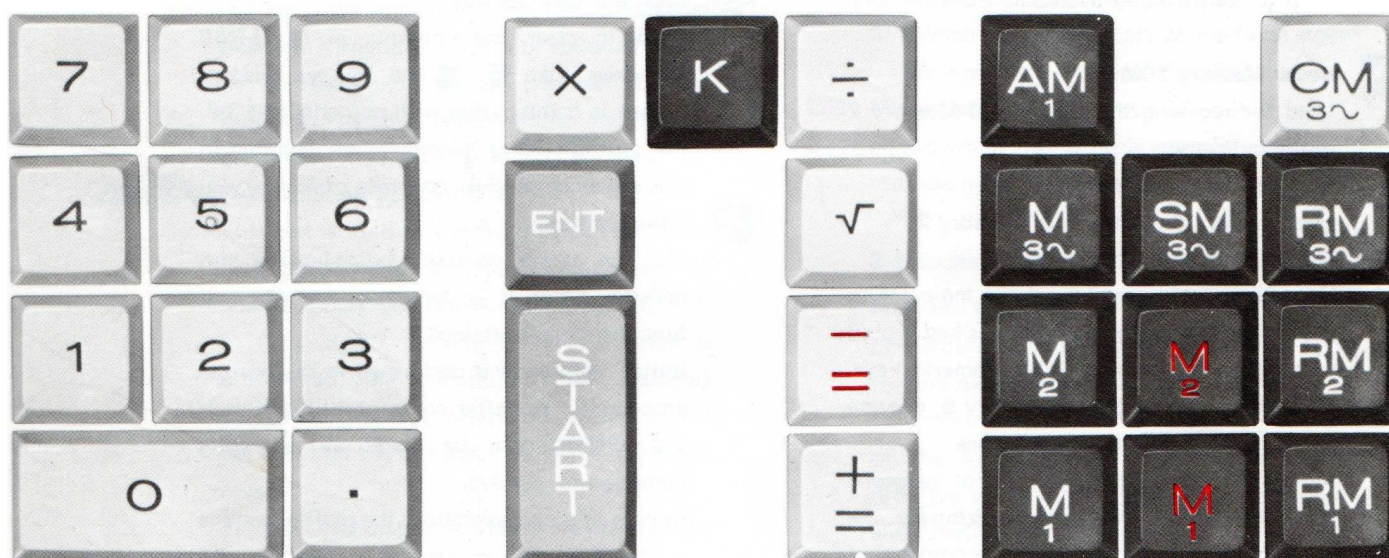
This key is provided to shift the indicated value one digit in the arrow indicated direction, and is mainly used when partially correcting accidentally entered digits.

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 $\mp$ instruction, to recall a of $a \times b$ where there is any possibility of 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 CHECK MODE (Program Mode

Switch is set in ^{PRO}_{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 at ^{OPE}_{CHE}), the program is returned to zero step.

CI Clear Indicator Key

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

AM₁ Accumulative Memory 1 Key

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

M₁ Memory 1 Key

Used to add displayed data to Memory 1.

 **Minus Memory 1 Key**
Used to subtract displayed data from Memory 1

 **Recall Memory 1 Key**
Used for recalling the contents of Memory 1 to the indicator.

 **Clear Memory 1 Key**
Used to clear the contents of Memory 1.

 **Memory 2 Key**
Used to add displayed data to Memory 2.

 **Minus Memory 2 Key**
Used to subtract displayed data from Memory 2.

 **Recall Memory 2 Key**
Used for recalling the contents of Memory 2 to the indicator.

 **Clear Memory 2 Key**
Used to clear the contents of Memory 2.

 **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 $\boxed{2}$, $\boxed{M}$ and $\boxed{5}$ in this sequence. Minus Memories 3-9 keys are not provided but, the  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 $\boxed{2}$, $\boxed{M}$, $\boxed{M}$ and $\boxed{5}$ 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   keys.

 **Storage Memory 3-9 key**
Similar to the  key, this key is used to read into No3 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 $\boxed{2}$,  and $\boxed{5}$ are operated in sequence, the contents of Memory 5 is changed to 2 from 6.

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

For example, to recall the contents of Memory 5, operate  and $\boxed{5}$ in this sequence.

 **Clear Memory 3-9 Key**
Used to clear the contents in No.3-No9 memories. Like ,  and  keys, this key is used in combination with numeric keys 3-9. For example, keys  and $\boxed{5}$ are operated in this sequence to clear contents of Memory 5.

 **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 $\boxed{\pm}$ 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 $\boxed{\pm}$ key $(n-1)$ times. Thus, $a \boxed{\times} \boxed{\pm} \boxed{\pm} \boxed{\pm} \boxed{\pm}$ gives a^5 .

 **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		-a
-a		a
0		0

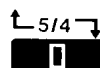
(Where $a \neq 0$)

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 off, up or rounded 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

Glow 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

Glow 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 Decimal Point Selector Dial.



Overflow Lamp

Glow 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

Glow 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.

SJ Sense Jump Key

This program key is used for both manual program storage and executing a program.

1) When storing a program

If this key is manually depressed, a sense jump instruction SJ is stored in the program.

2) When executing a program

If this key is manually depressed at the SJ instruction in the program, the program will jump to the specified program step. If the  key is depressed the program will proceed to the following step. The ENTRY lamp lights at this instruction.

EJ Entry Jump Key

Like MJ and UJ instructions, this key is used for manual program storage.

When this key is depressed with the Program Mode Switch in LRN position, an entry jump instruction EJ is stored in the program. During execution of a program at this instruction, the program stops to await entry of data and the ENTRY lamp lights. When an entry is made and the  key is depressed, a jump is accomplished to the specified program step ; when the  key is depressed without entry, the program proceeds without jumping.

MJ Minus Jump Key

Used for manual storing of programs. When this key is depressed, a minus jump instruction MJ is stored in the program.

Minus jump instruction automatically judges the calculation results at the step immediately before this instruction is programmed ; if the value is zero or positive it proceeds as it is, and if it is negative, it jumps to the specified program step.

UJ Unconditional Jump Key

This key is used to store a manual program and to execute programmed calculation.

1) Storing program

With the program mode switch set to LRN, depressing this key during manual operation stores the unconditional jump

(UJ) instruction in the program.

2) Executing programmed calculation

With the program mode switch set to OPE, pushing UJ *n* (entry keys —) causes a jump to the jump destination specified in the FJ *n* step instruction. This key is used for program subdivision and in programs involving a relatively large number of decisions.

Refer to information on decision functions for details.

FJ Flag Jump Key

This key is used in manual storing of programs in combination with , , , and  keys. If this key is depressed with Program Mode Switch set in LRN position, a flag for jump instruction is given, and a jump destination specifying instruction FJ is stored in the program. Each FJ instruction and the corresponding FJ, MJ, SJ, or UJ instruction are accompanied by an identical flag code.

START Start Key

This key is used for execution of program calculation and to check the contents of a program.

1) Execution of program calculation

If this key is depressed with the Program Mode Switch set in OPE position, the program is advanced to ENT, SJ and EJ instructions in the program.

2) Program Checking

If this key is depressed with the Program Mode Switch set in ^{PRO}CHE position, the contents of the program can be checked. The program is advanced one step each time the  key is depressed, and simultaneously the immediate step number and program instruction code are indicated together.

3) Checking program calculation

^{OPE}
With the program mode switch set to CHE, depressing this key checks the program calculation. Each operation of this key advances the program one step, followed

by arithmetic operation and display of the result.

2-5 Explanation of Program Switches



Program Mode Switch

Positions LRN, OPE, PRO, and OPE have the following functions.

1) LRN (Learn)

When storing a program, manually using the keyboard, and when a program card is read into the calculator or when clearing a program, the switch is set in this position.

2) OPE (Operation)

- (i) The switch is set in this position when a program calculation is executed using actual data.
- (ii) The switch is also set in this position when an ordinary manual calculation is made.

3) PRO (Program Check)

- (i) The switch is set in this position when checking the contents of a stored program.
- (ii) The switch is also set in this position when partial correction to the program is to be made.
(For further details on program checking and correction refer to III-3-4 Method of Checking and Correcting Program Contents.)

4) OPE (Operation Check)

Use this position for effecting calculation at each step in accordance with the contents of the stored program, so that results may be checked at each step.



Program Selector Switch

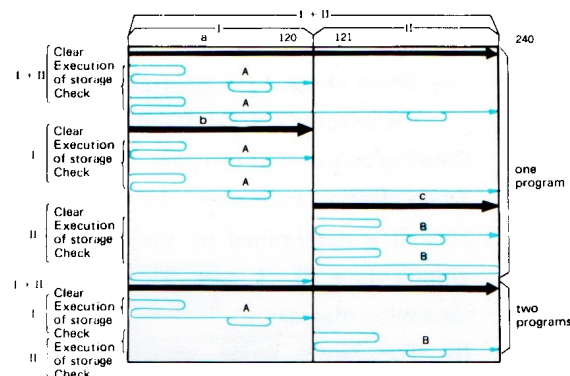
When storing a program, executing a program calculation, and checking or clearing a program, program ranges are selected by this switch.

1) Program storage

- (i) When two independent programs are stored at the same time, a program

within 120 steps is stored on side I and a program within 120 steps is stored on side II.

- (ii) When one program of over 120 steps is stored, the switch is set in the I + II position.



(Refer to information on UJ for program details.)

- (iii) When the respective programs at I and II positions exceed 120 steps and 120 steps respectively, the excess is stored automatically in II or I.

This relationship is illustrated above.

2) Execution of a program

The program is also classified by the changeover among I + II, I and II in the case of program calculation execution in accordance with the stored program. This will be more easily understood by following the schematic diagram above.

- (i) To execute a program by using the contents of program A, set the switch in I + II or I position.
- (ii) To execute a program by using the contents of program B, set the switch in II position.
- (iii) When the two programs are stored and they are used as required, execute program A with the switch set to I position, and execute program B with the switch set in II position.

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 a dual program, 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 being read into the calculator by the operation of entry keys and the  key, 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.

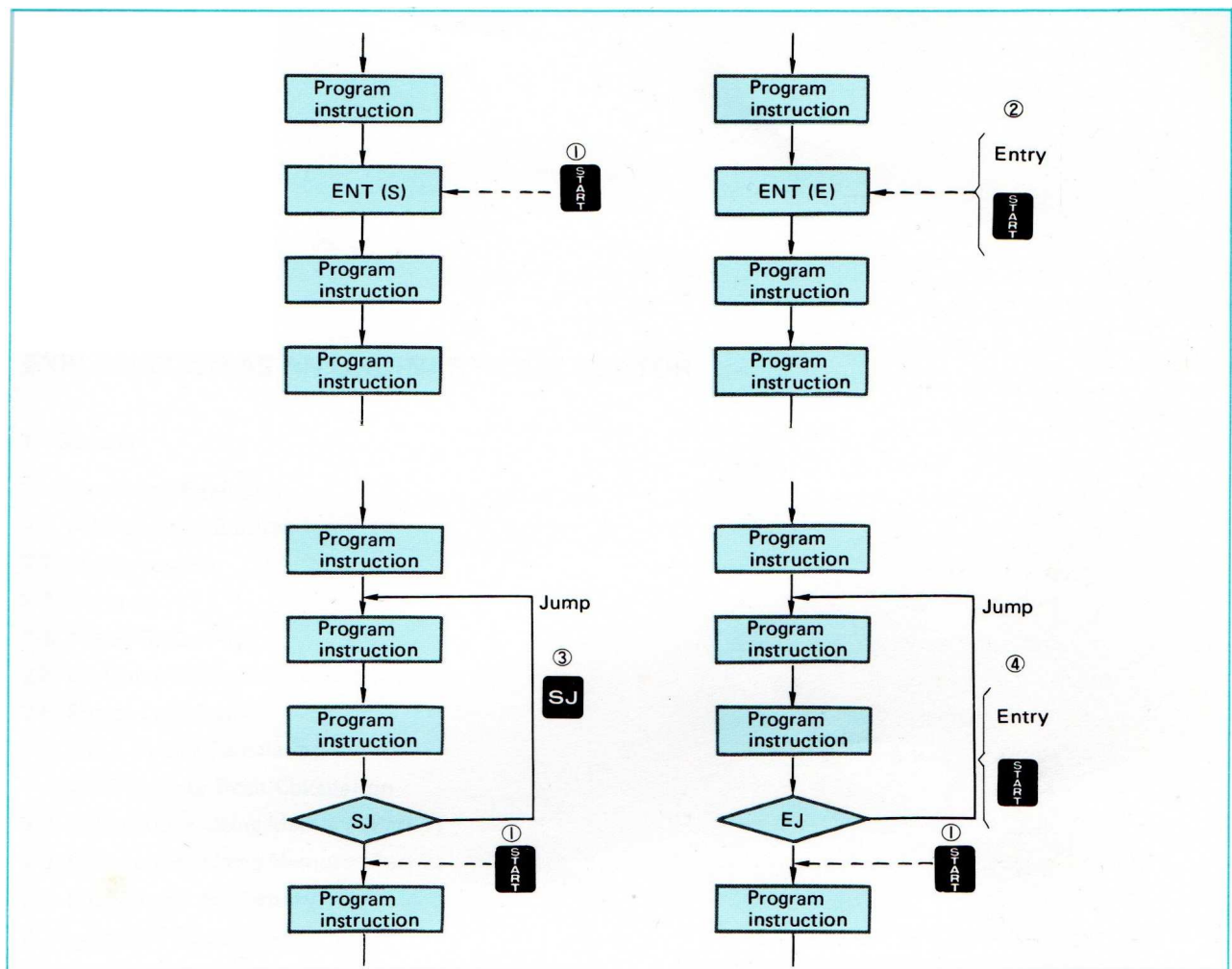
2-6 Explanation of Program Lamp



Entry Lamp

This lamp is used when executing a program. The lamp glows 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) Data is read in by the operation of entry keys and the  key, 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 Power Calculation

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

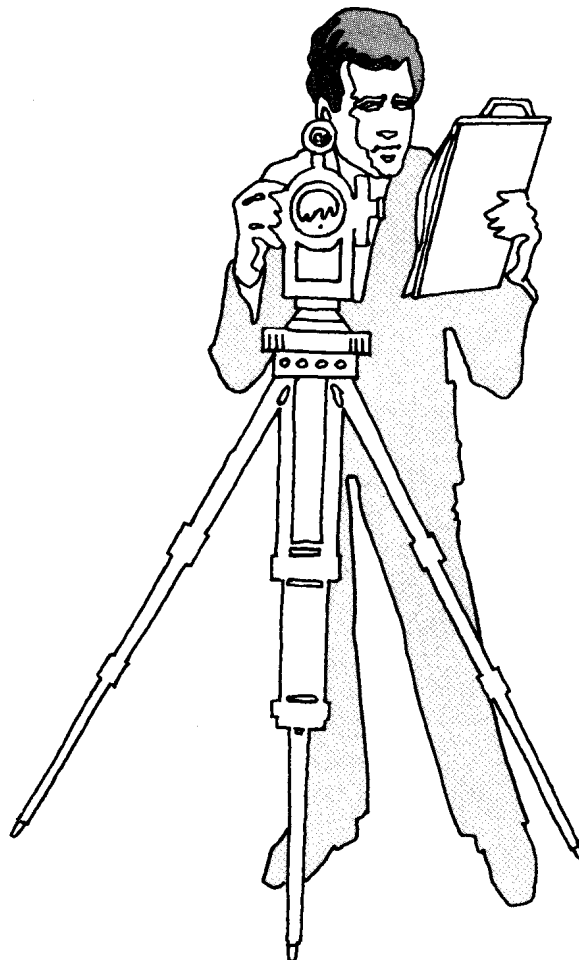
2-10 Applied Calculations

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 PRO_{CHE} 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 $=$	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 $=$	12.340 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 $=$	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 $=$ 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 RV key is now depressed, the multiplicand is recalled. Depress the RV 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 press the 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 M key is pressed.
- 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 CH 3 $\times$ 4 3.45 5 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 CH key.
- For example (2), the key operation 1.2 CH $\times$ 3.45 CH $\pm$ is also available.
- With settings at CH and 5/4 , the result is obtained to four decimal places, with rounding off of the fifth decimal digit.
- With settings at CH and 5/4 , the fifth decimal digit is dropped off and the result becomes 97,504.1907. With the Round-off Switch set at 5/4 rounding up occurs and the result is the same as in (3).

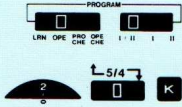
Calculation example	Setting	Key operation	Display
4) (Chain Multiplication) $123.456 \times 789.456 \times 456.123$ $= 44,455,125.4096$		1 123.456 2 $\times$ 3 789.456 4 $\times$ 5 456.123 6 $\pm$	123.456 123.456 789.456 97,463.079936 456.123 44,455,152.4096

- (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.

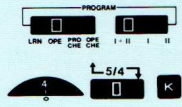
Calculation example	Setting	Key operation	Display
5) (Floating Calculation) 12.345×789.456 $= 9745.8342$		1 12.345 2 $\times$ 3 789.456 4 $\pm$	12.345 12.345 789.456 9745.83432

- (1) If the Decimal Point Selector Dial is set at 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.

2-5 Constant Multiplication and Division

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

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

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

- (1) In division, the second figure entered after $\mathbf{K}$ 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.

2-6 Power Calculation

2-6-1 Involution

Calculation example	Setting	Key operation	Display
2) (Squaring a Number) $3.14159^2 = 9.869588$		1 3.14159 2 $\times$ 3 $\pm$	3.14159 3.14159 9.869588
2) (Raising to n Powers) $3^8 = 6,561$		1 3 2 $\times$ 3 $\pm$ 4 $\pm$ 5 $\pm$ 6 $\pm$ 7 $\pm$ 8 $\pm$ 9 $\pm$	3. 3. 9. 27. 81. 243. 729. 2,187. 6,561.

- (1) The following method can also be used for squaring and raising to the 4th and 8th power : 3
 $(\times) (\pm)$ (square) $(\times) (\pm)$ (4th power) $(\times) (\pm)$ (8th power)

2-6-2 Square Root Calculation

Calculation exmaple	Setting	Key operation	Display
$\sqrt{2} = 1.414213562373095$		1 2 2 $\sqrt{}$	2. 1.414213562373095

- (1) When the Decimal Point Selector Dial is set at F, all 16 effective digits are displayed.
 (2) When the decimal point selector is set at other positions, the result is obtained to the designated number of decimal places.

2-7 Calculations Using Memory (Part 1)

Calculation example	Setting	Key operation	Display
1) (Sum of Products of Chain Multiplication) $243 \times 18 \times 0.86 = 3,761.64$ $576 \times 32 \times 0.78 = 14,376.96$ $387 \times 29 \times 0.88 = 9,876.24$ <hr/> Total 28,014.84		1 243 2 $\times$ 3 18 4 $\times$ 5 .86 6 $\pm$ 7 576 8 $\times$ 9 32 10 $\times$ 11 .78 12 $\pm$ 13 387 14 $\times$ 15 29 16 $\times$ 17 .88 18 $\pm$ 19 RM	243. 243. 18. 4,374. 0.86 3,761.64 576. 576. 32. 18,432. 0.78 14,376.96 387. 387. 29. 11,223 0.88 9,876.24 28,014.84

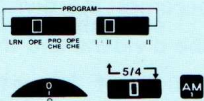
When the AM key is locked and the $\pm$ or $\mp$ key is depressed, the result is automatically added to or subtracted from Memory 1. To recall the contents of this memory, depress the RM key; to erase the contents, depress the CM key.

(1) When calculating the difference of products, depress the $\mp$ key instead of the $\pm$ key.

(2) Figures are added to the first memory only when the $\pm$ key is depressed. Depressing the $\pm$ key between steps 3 and 4 by mistake would result in the intermediate product being added to the memory.

(3) The position of the decimal point in figures entered in the memory corresponds to the setting of the Decimal Point Selector Dial when they are entered, regardless of the subsequent resetting of the Decimal Point Selector Dial.

Calculation example	Setting	Key operation	Display
2) (Sum of Products of Constant Multiplication) $0.85 \times 123 = 104.55$ $0.85 \times 456 = 387.60$ $0.85 \times 789 = 670.65$ <hr/> Total 1,162.80		1 .85 2 $\times$ 3 123 4 $\pm$ 5 456 6 $\pm$ 7 789 8 $\pm$ 9 RM	0.85 0.85 123. 104.55 456. 387.60 789. 670.65 1,162.80

Calculation example	Setting	Key operation	Display
2) (Subtotals and Grand Total of Products)		1 123	123.
Quantity		2 $\times$	123.
$123 \times 45 = 5,535$		3 45	45.
$456 \times 78 = 35,568$		4 $\pm$	5,535.
Subtotal 41,103		5 456	456.
		6 $\times$	456.
		7 78	78.
		8 $\pm$	35,568.
$789 \times 12 = 9,468$		9 $\overline{RM}$	41,103.
$147 \times 25 = 3,675$		10 $\overline{M}$	41,103.
Subtotal 13,143		11 $\overline{CM}$	41,103.
Grand total 54,246		12 789	789.
		13 $\times$	789.
		14 12	12.
		15 $\pm$	9,468.
		16 147	147.
		17 $\times$	147.
		18 25	25.
		19 $\pm$	3,675.
		20 $\overline{RM}$	13,143.
		21 $\overline{M}$	13,143. ... Add to second memory
		22 $\overline{RM}$	54,246. ... Recall grand total

2-9 Supplement to Memories

Model 1614P has fourteen memory registers.

(1) Memories 1 and 2

These memories are accumulative types. Contents are modified with the plus and minus memory keys.

(2) Memories 3 through 14

- Memories 3 through 14 include accumulative types and temporary storage types.
- Memories 10 through 14 cannot be used for manual operation, as they are only appli-

cable to card programming.

- There are no independent control keys for memories 3 through 9. Access is gained by combinations of the $\overline{M}$ $\overline{SM}$ $\overline{RM}$ $\overline{CM}$ keys and keys $\boxed{1}$ through $\boxed{9}$
- There is no minus memory key for memories 3 through 14. To subtract from these memories, use the $\overline{CM}$ key to change the sign of the entry prior to storage.

	Memory 1	2	3	4	5	6	7	8	9	10	11	12	13	14
Cumulative addition	$\overline{M}$	$\overline{M}$			$\overline{M}$	$\boxed{3}$		$\boxed{9}$		Used only for programmed calculation by card. Functions same as those of memories 3 through 9.				
Cumulative subtraction	$\overline{M}$	$\overline{M}$		$\overline{CM}$	$\overline{M}$	$\boxed{3}$		$\boxed{9}$						
Temporary storage					$\overline{SM}$	$\boxed{3}$		$\boxed{9}$						
Recall	$\overline{RM}$	$\overline{RM}$			$\overline{RM}$	$\boxed{3}$		$\boxed{9}$						
Cancellation	$\overline{CM}$	$\overline{CM}$			$\overline{CM}$	$\boxed{3}$		$\boxed{9}$						

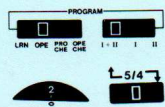
2-10 Applied Calculations

Exaplnation of **RV** key

The **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 $\text{divident} \div \text{divisor} \times 100$, since the total becomes the divisor. Since the **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 **RV** key to reverse the two figures. If the **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)																					
Obtaining the sales percentage of each store against total sales :																					
<table><tr><th>Store</th><th>Sales (\$)</th><th>%</th></tr><tr><td>A</td><td>3,524,500</td><td>(22.43)</td></tr><tr><td>B</td><td>5,163,200</td><td>(32.86)</td></tr><tr><td>C</td><td>4,048,600</td><td>(25.77)</td></tr><tr><td>D</td><td>2,975,800</td><td>(18.94)</td></tr><tr><td>Total</td><td>(15,712,100)</td><td>(100.00)</td></tr></table>	Store	Sales (\$)	%	A	3,524,500	(22.43)	B	5,163,200	(32.86)	C	4,048,600	(25.77)	D	2,975,800	(18.94)	Total	(15,712,100)	(100.00)			
Store	Sales (\$)	%																			
A	3,524,500	(22.43)																			
B	5,163,200	(32.86)																			
C	4,048,600	(25.77)																			
D	2,975,800	(18.94)																			
Total	(15,712,100)	(100.00)																			
Formula : $\frac{\text{Sales of each store}}{\text{Total sales}} \times 100$																					
$= \frac{\text{Sales of each store}}{15,7121}$																					
		1 3524500	3,524,500.																		
		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 AM	157,121.00																		
		13 K	157,121.00																		
		14 $\div$	157,121.00																		
		15 3524500	3,524,500.																		
		16 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 RM	100.00																		

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

simply by depressing the $\pm$ key. Thereafter composition ratios can be obtained by repeatedly entering the various figures and depressing the $\pm$ key.

The RV key can also be used for checking registered figures in multiplications and divisions, especially multiplicands and dividends. These can be recalled by depressing the RV key once ; depressing it once more

returns the figures to their original position. In addition and subtraction, the RV key recalls the numeral to be calculated.

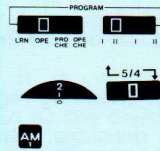
For example : $a + b + b = d$

Operation : $a \pm b \pm \text{RV} \pm$

Note :

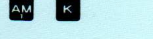
The operation of $a \times b \pm \text{RV}$ recalls b instead of a , only when a is a zero.

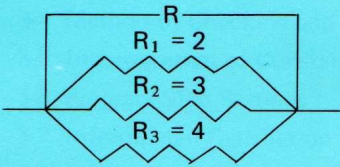
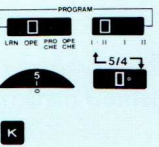
Calculation example	Setting	Key operation	Display
2) (Proportional Allotment Calculation)			
Distributing a total budget			
of \$1,583,000 among 4 stores			
according to their sales records :			
		1 1985	1,985.
		2 $\pm$	1,985.
		3 4623	4,623.
		4 $\pm$	6,608.
		5 2748	2,748.
		6 $\pm$	9,356.
		7 6136	6,136.
		8 $\pm$	15,492.
		9 $\div$	15,492.
		10 1583	1,583.
		11 RV	15,492. . .
		12 $\times$	0.102181771236767
		13 ΔM	0.102181771236767
		14 K	0.102181771236767
		15 1985	1,985.
		16 $\pm$	203.
		17 4623	4,623.
		18 $\pm$	472.
		19 2748	2,748.
		20 $\pm$	281.
		21 6136	6,136.
		22 $\pm$	627.
		23 RM	1,583.
			Interchanging denominator and numerator

Calculation example						Setting	Key operation	Display	
3) (Sales and Gross Profit Calculation)							1 1234	1,234.	
							2 $\times$	1,234.	
							3 23	23.	
							4 $\times$	28,382.	
							5 .2	0.2	
							6 $\pm$	5,676.40	
							7 M_2	5,676.40	
							8 RV	28,382.	
							9 $\pm$	34,058.40	
							10 4567	4,567.	
							11 $\times$	4,567.	
							12 12	12.	
							13 $\times$	54,804.	
							14 .25	0.25	
							15 $\pm$	13,701.00	
							16 M_2	13,701.00	
							17 RV	54,804.	
							18 $\pm$	68,505.00	
							19 789	789.	
							20 $\times$	789.	
							21 45	45.	
							22 $\times$	35,505.	
							23 .3	0.3	
							24 $\pm$	10,651.50	
							25 M_2	10,651.50	
							26 RV	35,505.	
							27 $\pm$	46,156.50	
							28 RM_1	148,719.90 (Sales)	
							29 RM_2	30,028.90 (Gross profit)	
							30 $\pm$	118,691.00 (Total cost price)	
Cost price	Quantity	Total cost price	Installment 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)				

Calculation example	Setting	Key operation	Display
4) Calculation of Redemption by Annual Installments Using RV Key) 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.		1 .06	0.06
		2 M	0.06
		3 1	1
		4 $\frac{+}{-}$	1.060000
		5 $\times$	1.060000
		6 $\frac{+}{-}$	1.123600 ... square
		7 $\times$	1.123600
		8 $\frac{+}{-}$	1.262476 ... 4th power
		9 $\times$	1.262476
		10 $\frac{+}{-}$	1.593845 ... 8th power
		11 $\div$	1.593845
		12 1	1.
		13 RV	1.593845
		14 $\frac{+}{-}$	0.627413 -
		15 1	1.
		16 $\frac{+}{-}$	0.372587
		17 $\div$	0372487
		18 RM	0.060000
		19 RV	0.372587
		20 $\times$	0.161036214360672
		21 2000000	2,000,000.
		22 $\frac{+}{-}$	322,072,428721 (Redemption)

$$\text{Formula: } 2,000,000 \times \frac{0.06}{1 - \frac{1}{(1 + 0.06)^8}}$$

Calculation example				Setting	Key operation	Display
5) (Depreciation Calculation) Fifth year depreciation					1 350000	350,000. . . Acquisition cost
					2 	350,000.
					3 	350,000.
					4 .369	0.369
					5 	350,000.
					6 	129,150. — . . . First year depreciation
					7 	220,850. . . First year book value
					8 	81,494. — . . . Second year depreciation
					9 	139,356. . . Second year book value
					10 	51,422. — . . . Third year depreciation
					11 	87,934. . . Third year book value
					12 	32,448. — . . . Fourth year depreciation
					13 	55,486. . . Fourth year book value
					14 	20,474. — . . . Fifth year depreciation
					15 	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				

Calculation example	Setting	Key operation	Display
6) (Calculation of Automobile's Braking Distance) Finding the distance travelled by an automobile after the brakes are first applied until it stops completely : Formula : $X = \left(\frac{5}{18} V \right)^2 \times \frac{1}{19.6\mu}$ X = Braking distance V = Speed of automobile μ = Friction coefficient of tires and road surface Finding X when V = 40 km and $\mu = 0.4$		1 40 2 $\times$ 3 5 4 $\div$ 5 18 6 $\times$ 7 $\times$ 8 $\pm$ 9 $\div$ 10 19.6 11 $\div$ 12 .4 13 $\pm$	40. 40. 5. 200. 18. 11.11111111111111 11.11111111111111 123.4568 123.4568 19.6 6.298816326530612 0.4 15.7470
7) (Total Resistance of a Parallel Circuit)  $R = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}} = 0.92308$		1 2 2 $\div$ 3 $\pm$ 4 $\pm$ 5 $\mathbf{M}$ 6 3 7 $\div$ 8 $\pm$ 9 $\pm$ 10 $\mathbf{M}$ 11 4 12 $\div$ 13 $\pm$ 14 $\pm$ 15 $\mathbf{M}$ 16 $\mathbf{RM}$ 17 $\div$ 18 $\pm$ 19 $\pm$	2. 2. 1.00000 0.50000 0.50000 3. 3. 1.00000 0.33333 0.33333 4. 4. 1.00000 0.25000 0.25000 1.08333 1.08333 1.00000 0.92308

- (1) By depressing $\div$ $\pm$ the displayed figure a is divided by itself ($a \div a = 1$).
 Depressing the $\pm$ key once more automatically results in a reciprocal calculation of $1 \div a = \frac{1}{a}$

3

EXPLANATION AS A PROGRAMMABLE CALCULATOR

1. General

2. Before Commencing Program Operation

2-1 Explanation of Terms

2-2 Explanation of Symbols Used

3. For Program Operationn

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 Commencing 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
- 3) Programming
- 4) Program storage
- 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, M1, M2, 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 Card Codes.)

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 Card Codes" 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., M₁, 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.
 - i) 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  and  keys.

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

If keys , , , ,  and  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  key, like       , 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 , ,  and .

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  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  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  key or equivalent.

3) Then subsequent answers can be easily obtained by data input and simple operation of the  and  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  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 cleared
(iii)	II	121 - 240 steps	

(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-

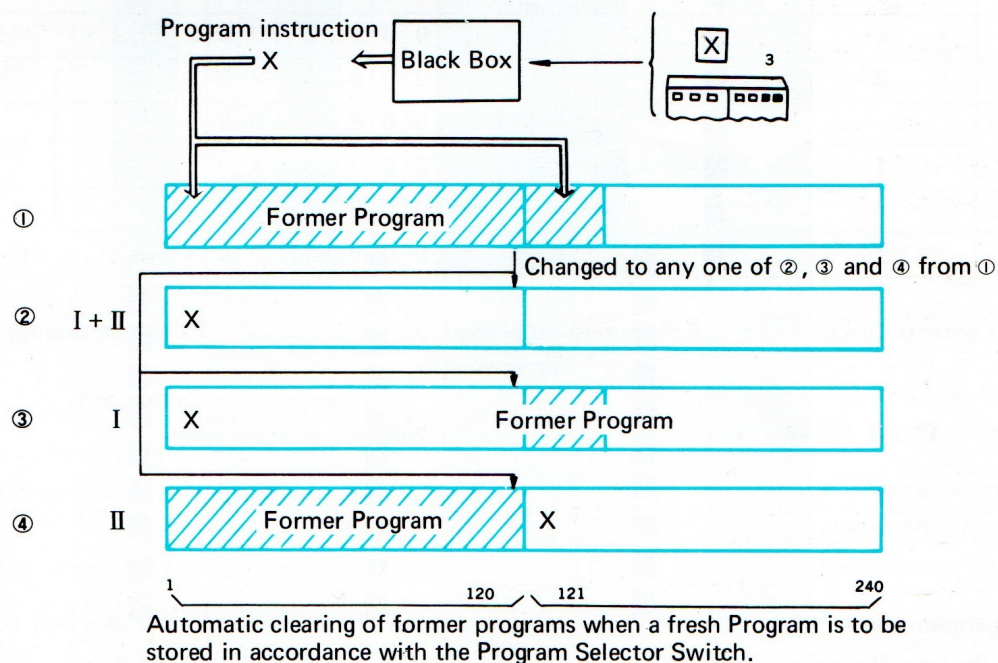
matically return to the 1st or 121st step in accordance with the Program Selector Switch.

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 PRO_{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 Contents	Key Operation	Display												Remarks		
1	E								0	0	1			1	0	1	r
2	x								0	0	2			0	0	3	
3	$\frac{+}{=}$								0	0	3			0	0	1	r ²
4	x								0	0	4			0	0	3	
5	3								0	0	5			7	0	3	} π
6	.								0	0	6			6	0	0	
7	1								0	0	7			7	0	1	
8	4								0	0	8			7	0	4	
9	$\pm$								0	0	9			0	0	1	πr^2

Program Contents Check

6 digits zero-suppressed
program step

2 digits zero-suppressed
program instruction code

2 digits zero-suppressed

3-4-2 Operation check

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

The program check described previously checks the content 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 Contents	Key Operation	Display	Remarks
0			0	
1	E		2	
2	X		2	r
3	$\pm$		4.00	
4	X		4.00	r^2
5	3		3	}
6	•		3	
7	1		3.1	
8	4		3.14	
9	$\pm$		12.56	πr^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	Code	Key Operation	Display	Key Operation	Display	Key Operation	Display	Remarks
0			0					
1	E		3					
2	SM3		3					
3	1		1					
4	SM4		1					
5	FJ							
6	1							
7	RM3		3		2		1	
8	X		3		2		1	
9	RM4		1		3		6	
10	$\pm$		3		6		6	
11	SM4		3		6		6	
12	1		1		1		1	
13	SC		- 1		- 1		- 1	If a figure becomes minus at step 16, a jump takes place to step 15.
14	M3		- 1		- 1		- 1	
15	RM3		2		1		0	
16	SC		- 2		- 1		0	
17	MJ							Upon completion of the loop, advance to the next step takes place.
18	1							
19	RM4						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 $\div$.

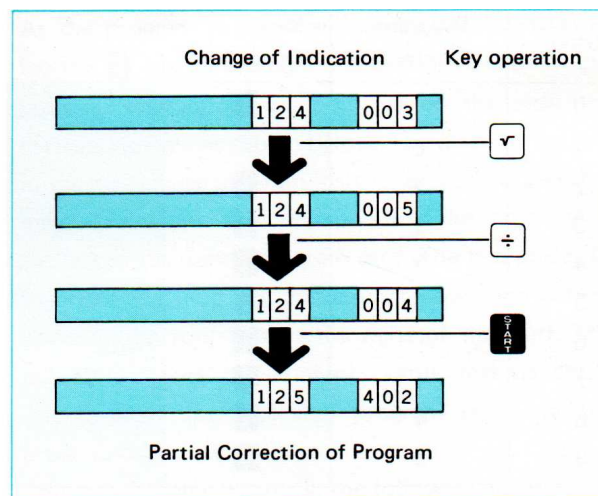
In other words, the wrong program is

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

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 $[x]$ 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 $[STOP]$ 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	Check 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 ₁	
62.	RM ₂	
63.	E	
64.	$\pm$	
65.	RM ₃	
66.	X	
67.	E	OPE $\rightarrow$ PRO CHE
68.	$\pm$	
69.	$\sqrt{}$	
70.	$\div$	
71.	RM ₁	
72.	$\pm$	
73.	$\sqrt{}$	

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 π (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	Keys to be operated	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

- Convert 3.14 into codes
- Punch a card in accordance with the converted codes.

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

Code	Constant
703 $\rightarrow$	3
600 $\rightarrow$	.
701 $\rightarrow$	1
704 $\rightarrow$	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 [C].

1) LRN

(i) Set the Program Mode Switch in LRN position, turn the power ON, or depress the [C] key. Thus the program is automatically reset to the 1st or 121st step as specified by the Program Selector Switch. If the [C] 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 [C] 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

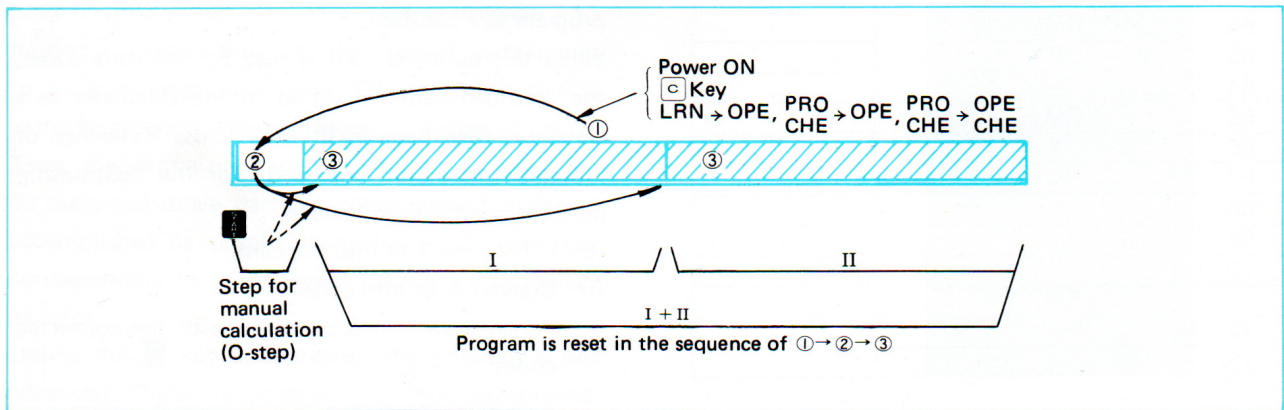
0-step before the 1st step, exclusive for manual calculation) instead of being reset to the 1st or 121st step as when the switch is in LRN position.

(ii) When the Program Mode Switch is set in either LRN or PRO_{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 [I] key once.

Operation of the [I] 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 [I] 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) PRO_{CHE}

(i) If the power is turned ON or the [C] key is depressed with the Program Mode Switch set in PRO_{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 PRO_{CHE} position when it was set in either OPE or OPE_{CHE} position, the program goes to the next step to the step where the switch was in OPE or OPE_{CHE} position.

4) OPE_{CHE}

Step-settings of OPE_{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,  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          in this sequence and obtain the answer (18.84).

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

This is because the function of the  key used in execution of program calculation and the function of the  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  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  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	Program calculation execution procedure	Program Instruction	Program calculation execution procedure	
0				a	Step for Manual calculation
1	E	a	X		
2	X		E	b	
3	E	b	X		
4	X		E	c	
5	E	c	$\pm$		
6	$\pm$		*(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 2d 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 ($\nwarrow$, 5/4, $\swarrow$)

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

Round-up	$\nwarrow$	<i>n</i>
Round-off	5/4	<i>n</i>
Round-down	$\swarrow$	<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	$\nwarrow$ round-up	$\nwarrow$ round-off	$\nwarrow$ round-down
M	$\nwarrow$	5/4	$\swarrow$
M + 1	<i>n</i>	<i>n</i>	<i>n</i>
	:	:	:
	:	:	:

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

Step	CANON PROGRAM CARD					
	40	20	10	8	4	2
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

(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	Manual	Display	Dial
20	E (a)	5	5.	
21	M1		5.	
22	+		5.	
23	E (b)	3	3.	
24	5/4		3.	
25	6		3.	
26	$\pm$		1.666667.	
27	RM1		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 the part $\pm$, $\mp$, unless in this example there is a $\sqrt{}$ instruction during the manual input.

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

Step	Program	Manual	Display	Dial
11	E (a)	5	5.	
12	M1		5.	
13	∇		5.	
14	2		5.	
15	E (b)	3	3.	
		$\sqrt{}$	1.73	
16	X		1.73	
17	RM1		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 $\pm$, $\mp$, and $\sqrt{}$ 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	Manual	Display	Dial
20	E (a)	1 . 2 3 4 5	1.2345	
21	$\pm$		1.2345	
22	E (b)	4 . 3 2 1 5	4.3215	
23	$\pm$		5.5560	
24	E (c)	. 2 2 2 8	0.2228	
25	$\mp$		5.3332	
26	E (d)	1 . 3 3 3 4	1.3334	
27	$\uparrow$		1.3334	
28	2		1.3334	
29	$\pm$		6.66	
30	E (e)	1 . 2 3 4 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 step31. 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 $\mp$ 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 (◇)

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 proceed 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)

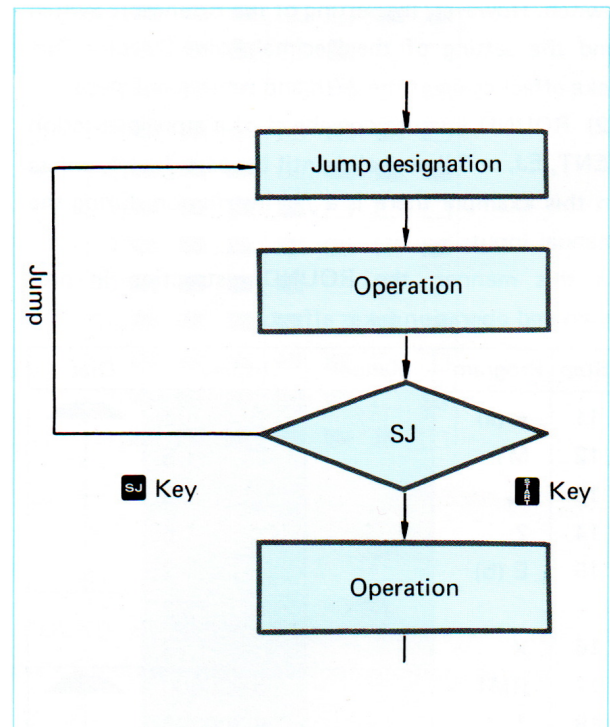
Normally during execution of program calculation, the program calculation advances sequentially starting from the 1st step of the stored program. However, the 1614P is provided with a function to stop the program flow in the middle, to decide the “then” conditions and to execute once again the program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

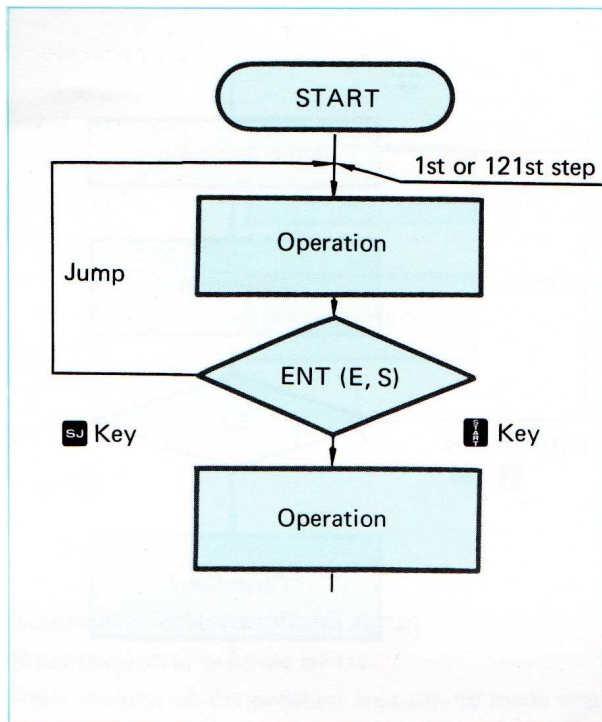
Such a function is called a decision function (jump function).

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.

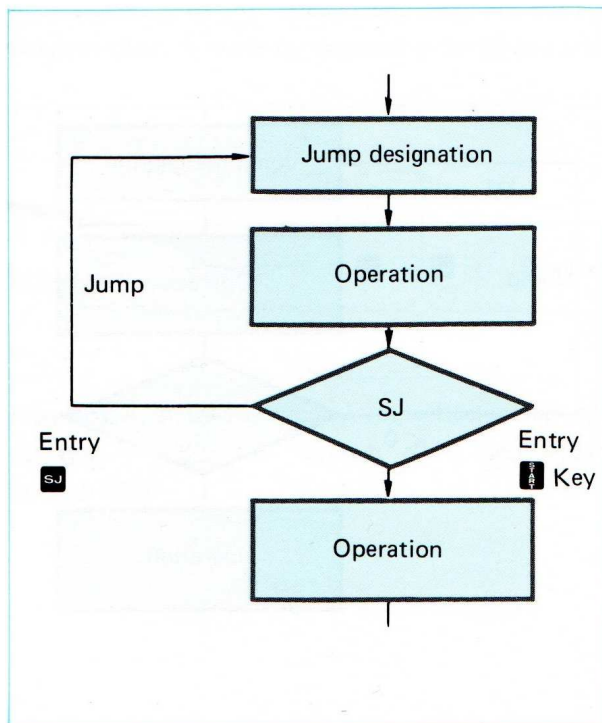
If it is decided to cause the program to jump, the  key is manually depressed, and the  key is depressed if it is intended to cause the program to advance without jumping.





[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.

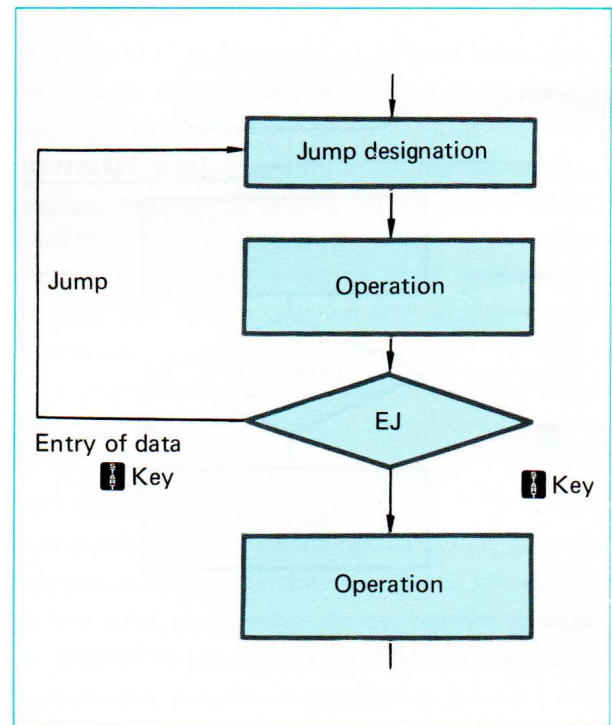


2) Further, it is possible to make an entry as required at the SJ instruction. In this case, depressing the **SJ** key after entry causes the program to jump to a specified step, or depressing the **ENT** key causes the program to advance without jumping.

3-9-2 Entry Jump (EJ)

In execution of program calculation, the program stops at the EJ instruction, as is the case of Sense Jump. When data is entered at this step by numeric key operation or by punched cards through the card-reader, the program jumps to the specified step by depressing the  key.

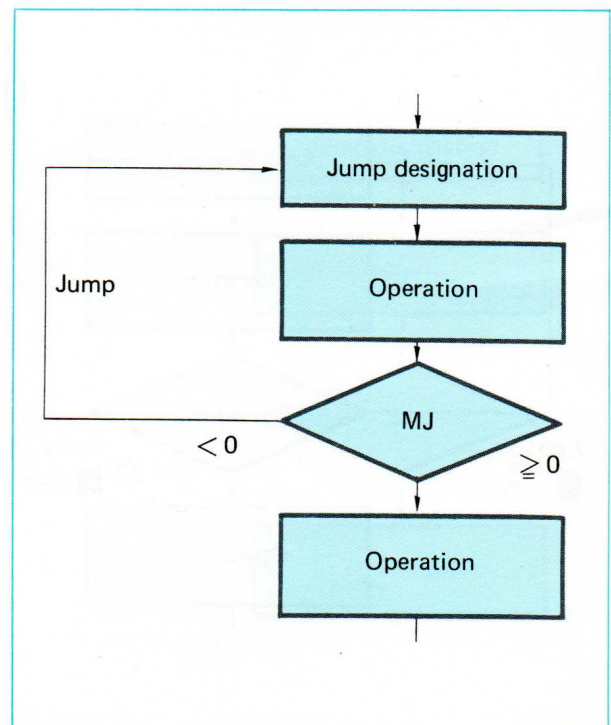
If the  key is depressed without data input, the program will advance without jumping.

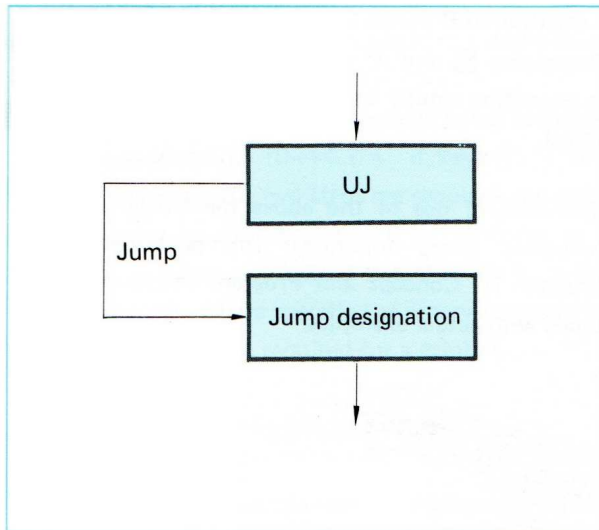


3-9-3 Minus Jump (MJ)

As the program advances and comes across the instruction, the positive or negative sign of the "then" calculation value is automatically judged; if the calculation value is negative, the program is caused to jump to the specified program step and if the value is positive or zero, the program is caused to advance without jumping.

If the MINUS JUMP and SIGN CHG instructions are used together, it is possible to automatically select the different portion of the program in accordance with three conditions, i.e., positive, negative and zero to execute program calculation.





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 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.

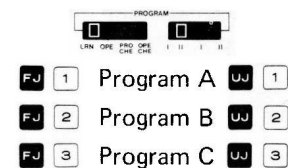
Access to each of the independent programs stored simultaneously is made by depressing the **UJ** key and

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

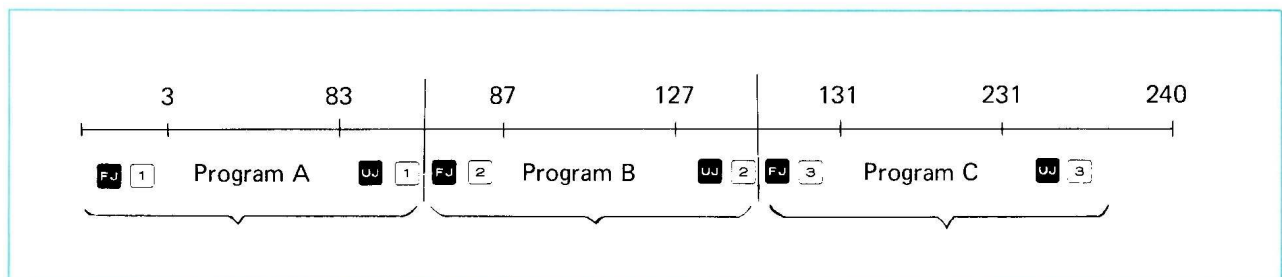
Example : Three independent programs

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 step 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, control is returned to FJ and 2 at the beginning of the program (B) by means of UJ and 2 in steps 127 and 128.

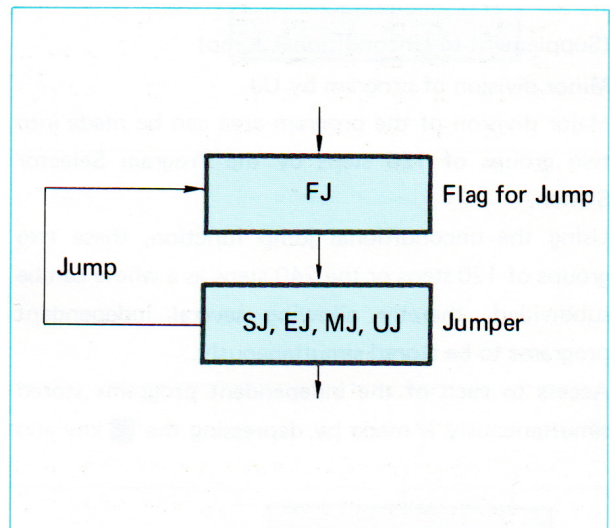
Execution of Program C

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.

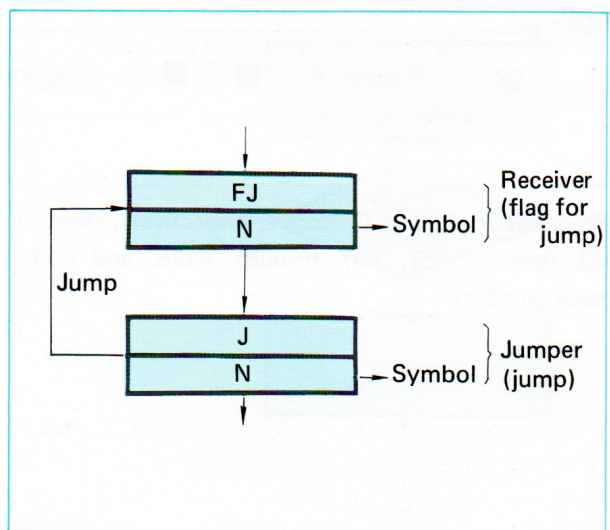
3-9-5 Specification of Jump Destination (FJ).

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

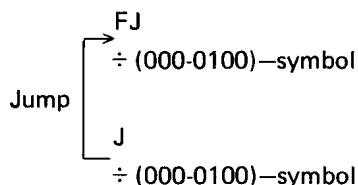
Symbol	Punch hole location	(Decimal)	(Binary)
	CANON PROGRAM CARD		
	STEP 40 20 10 8 4 2 1		
1	☒ ☒ ☒ ☒ ☐ ☒ ☐	10	→ 1010
2	☒ ☒ ☒ ☒ ☐ ☒ ☒	11	→ 1011
3	☒ ☒ ☒ ☒ ☒ ☐ ☐	12	→ 1100
4	☒ ☒ ☒ ☒ ☒ ☐ ☒	13	→ 1101
5	☒ ☒ ☒ ☒ ☒ ☒ ☐	14	→ 1110
6	☐ ☐ ☐ ☐ ☐ ☐ ☐		

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

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	☐	☐	☒	☒	☐	☒	☒
22	☐	☐	☒	☒	☒	☐	☐
23	☐	☐	☒	☒	☒	☐	☒
24	☐	☐	☒	☒	☒	☒	☐

Without punch
hole locations
Fully punched

When 8, 4, 2
and 1 are
punched

Combination
punch of 10
and 8, 4, 2, 1

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.

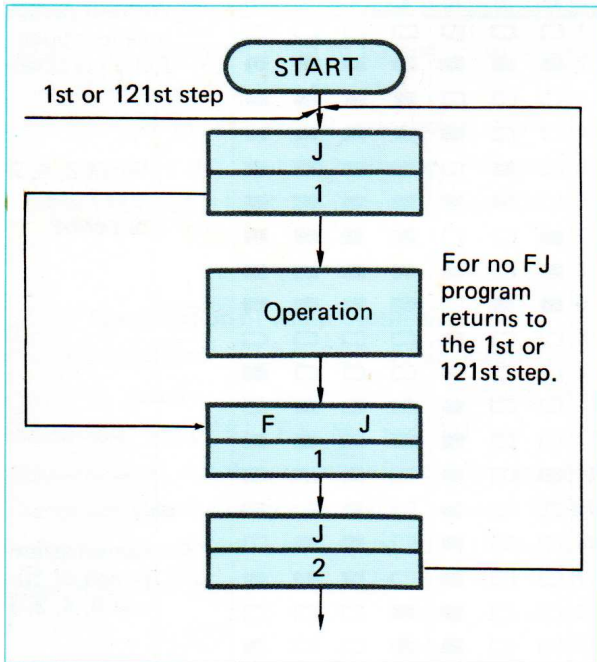
2) Jump when no receiver (FLAG FOR JUMP) is incorporated.

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.

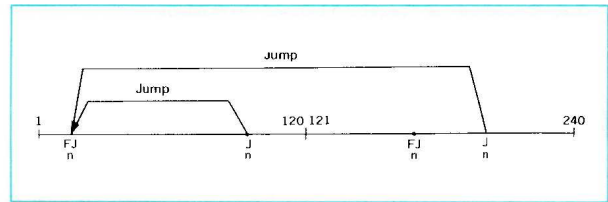


3) Jump when several receivers (FLAG FOR JUMP) are incorporated.

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.



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 I' position.

3-10 Subroutine Function

3-10-1 Subroutine and Main Routine

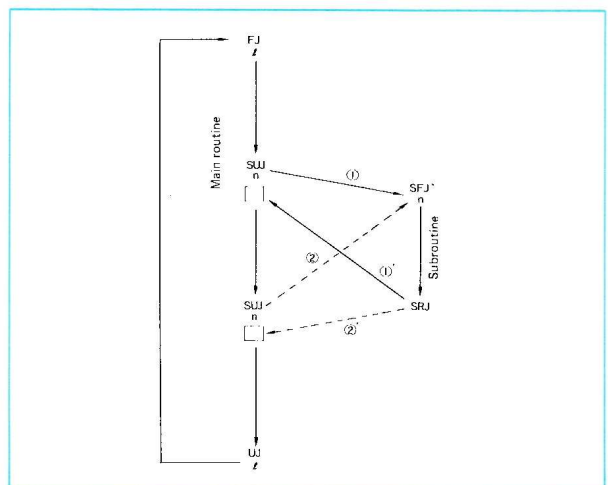
Commonly used operations, which can be repeated within a program, are known as subroutines and are separated from the main routine. Model 1614P provides for three types of subroutine instructions :

SUJ (SUBROUTINE UNCONDITIONAL JUMP)

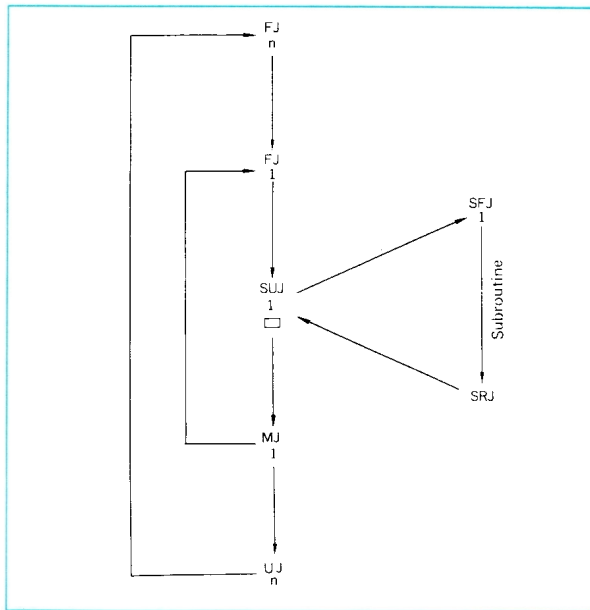
SFJ (SUBROUTINE FLAG FOR JUMP)

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.



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

FJ1 sin x UJ1 FJ2 cosx UJ2 FJ3 tanx UJ3

Example :   cosx

(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^n	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-Radian	309		21	1	
MS-24	Radian-Degree	310		22	1	
MS-25	N!	311	16	22	1	14, 13, 12

FJ Symbol code used	Input/Output Condition
200	$\left. \begin{array}{l} -10^{16} < x < 10^{16} \quad (x : \text{Degree}) \\ -10^{16} < x < 10^{16} \quad (x : \text{Degree}) \\ -10^{16} < x < 10^{16} \quad (x : \text{Degree}) \end{array} \right\} \text{Input}$
201	
202, 203, 204	
205, 206, 207	Input : $-1 \leq x \leq 1$ Output : $-90^\circ \leq \sin^{-1} x \leq 90^\circ$
208, 209, 210	Input : $-1 \leq x \leq 1$ Output : $0^\circ \leq \cos^{-1} x \leq 180^\circ$
211, 212, 213	Input : $-10^7 < x < 10^7$ Output : $-90^\circ < \tan^{-1} x < 90^\circ$
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	$\left\{ \begin{array}{ll} \text{Input} & r : 14\text{th memory} \quad \theta : 13\text{th memory } (-10^3 \leq \theta \leq 10^3) \\ \text{Output} & x : 14\text{th memory} \quad y : 13\text{th memory} \end{array} \right.$
412, 413, 414, 500, 501, 502, 503, 504	$\left\{ \begin{array}{ll} \text{Input} & x : 14\text{th memory} \quad y : 13\text{th memory} \\ \text{Output} & r : 14\text{th memory} \quad \theta : 13\text{th memory } (0^\circ \leq \theta < 360^\circ) \end{array} \right.$ <p>Input Degree : 14th memory Minute : 13th memory Second : 12th memory</p> <p>Print instruction included</p>
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 : π	$\left. \begin{array}{l} -10^{16} < x < 10^{16} \\ (x: \text{Degree}) \end{array} \right\}$	Constant
GF-2	$\sin^{-1} x, \cos^{-1} x,$ $\tan^{-1} x, \pi$	6 + 3	UJ 1 : $\sin^{-1} x$ UJ 2 : $\cos^{-1} x$ UJ 3 : $\tan^{-1} x$ UJ 4 : π	$\left. \begin{array}{l} -1 \leq x \leq 1 \\ -10^{15} \leq x \leq 10^{15} \end{array} \right\}$	$-90^\circ \leq x \leq 90^\circ$ $0^\circ \leq x \leq 180^\circ$ $-90^\circ < x < 90^\circ$ 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	$\left. \begin{array}{l} -32 \leq x \leq 32 \\ -10^{15} \leq x \leq 10^{15} \\ -32 \leq x \leq 32 \end{array} \right\}$	Constant
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 : $\log_e x$	$\left. \begin{array}{l} -10^{15} \leq x \leq 10^{15} \\ 1 \leq x \leq 10^{15} \\ -1 < x < 1 \end{array} \right\}$	
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	$\left. \begin{array}{l} 0 < x < 10^{16} \\ 0 < x < 10^{16} \\ -15 \leq x \leq 15 \\ -32 \leq x \leq 32 \end{array} \right\}$	
GF-6	$a^x, \sqrt[3]{x}$	6 + 4	UJ 1 : a^x UJ 2 : $\sqrt[3]{x}$	$\left. \begin{array}{l} a > 0 \\ 0 \leq x < 10^{16} \end{array} \right\}$	

Statistic program

File	Program	Number of Program steps	Number of cards	FJ symbol code 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 Paired 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 ₁	M ₂	M ₃	M ₄	M ₅	M ₆	M ₇	M ₈	M ₉	M ₁₀	M ₁₁	M ₁₂	M ₁₃	M ₁₄
M SM	30		M ₁	M ₂	SM ₃	SM ₄	SM ₅	SM ₆	SM ₇	SM ₈	SM ₉	SM ₁₀	SM ₁₁	SM ₁₂	SM ₁₃	SM ₁₄
RM	40		RM ₁	RM ₂	RM ₃	RM ₄	RM ₅	RM ₆	RM ₇	RM ₈	RM ₉	RM ₁₀	RM ₁₁	RM ₁₂	RM ₁₃	RM ₁₄
C	50	CI	CM ₁	CM ₂	CM ₃	CM ₄	CM ₅	CM ₆	CM ₇	CM ₈	CM ₉	CM ₁₀	CM ₁₁	CM ₁₂	CM ₁₃	CM ₁₄
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 M₃ 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	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
002	≡	2	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
003	X	3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
004	÷	4	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
005	√	5	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

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

101	ENT	6	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
102	SJ	7	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
103	EJ	8	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
105	MJ	9	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
106	UJ	10	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
107	FJ	11	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
108	SUJ	12	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
109	SFJ	13	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
110	SRJ	14	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

3) 20 (Plus memory instructions : PLUS MEMORY)

201	M ₁	1	☐	☒	☐	☐	☐	☐	☒
202	M ₂	2	☐	☒	☐	☐	☐	☒	☐
203	M ₃	3	☐	☒	☐	☐	☐	☒	☒
204	M ₄	4	☐	☒	☐	☐	☒	☐	☐
205	M ₅	5	☐	☒	☐	☐	☒	☐	☒
206	M ₆	6	☐	☒	☐	☐	☒	☒	☐
207	M ₇	7	☐	☒	☐	☐	☒	☒	☒
208	M ₈	8	☐	☒	☐	☒	☐	☐	☐
209	M ₉	9	☐	☒	☐	☒	☐	☐	☒
210	M ₁₀	10	☐	☒	☐	☒	☐	☒	☐
211	M ₁₁	11	☐	☒	☐	☒	☐	☒	☒
212	M ₁₂	12	☐	☒	☐	☒	☒	☐	☐
213	M ₁₃	13	☐	☒	☐	☒	☒	☐	☒
214	M ₁₄	14	☐	☒	☐	☒	☒	☒	☐

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

301	M ₁	1	☐	☒	☒	☐	☐	☐	☒
302	M ₂	2	☐	☒	☒	☐	☐	☒	☐
303	SM ₃	3	☐	☒	☒	☐	☐	☒	☒
304	SM ₄	4	☐	☒	☒	☐	☒	☐	☐
305	SM ₅	5	☐	☒	☒	☐	☒	☐	☒
306	SM ₆	6	☐	☒	☒	☐	☒	☒	☐
307	SM ₇	7	☐	☒	☒	☐	☒	☒	☒
308	SM ₈	8	☐	☒	☒	☒	☐	☐	☐
309	SM ₉	9	☐	☒	☒	☒	☐	☐	☒
310	SM ₁₀	10	☐	☒	☒	☒	☐	☒	☐
311	SM ₁₁	11	☐	☒	☒	☒	☐	☒	☒
312	SM ₁₂	12	☐	☒	☒	☒	☒	☐	☐
313	SM ₁₃	13	☐	☒	☒	☒	☒	☐	☒
314	SM ₁₄	14	☐	☒	☒	☒	☒	☒	☐

5) 40 (Recall memory instructions : RECALL MEMORY)

401	RM ₁	1	☒	☐	☐	☐	☐	☐	☒
402	RM ₂	2	☒	☐	☐	☐	☐	☒	☐
403	RM ₃	3	☒	☐	☐	☐	☐	☒	☒
404	RM ₄	4	☒	☐	☐	☐	☒	☐	☐
405	RM ₅	5	☒	☐	☐	☐	☒	☐	☒
406	RM ₆	6	☒	☐	☐	☐	☒	☒	☐
407	RM ₇	7	☒	☐	☐	☐	☒	☒	☒
408	RM ₈	8	☒	☐	☐	☒	☐	☐	☐
409	RM ₉	9	☒	☐	☐	☒	☐	☐	☒
410	RM ₁₀	10	☒	☐	☐	☒	☐	☒	☐
411	RM ₁₁	11	☒	☐	☐	☒	☐	☒	☒
412	RM ₁₂	12	☒	☐	☐	☒	☒	☐	☐
413	RM ₁₃	13	☒	☐	☐	☒	☒	☐	☒
414	RM ₁₄	14	☒	☐	☐	☒	☒	☒	☐

6) 50 (Clear instructions : CLEAR)

500	CI	1	☒	☐	☒	☐	☐	☐	☐
501	CM ₁	2	☒	☐	☒	☐	☐	☐	☒
502	CM ₂	3	☒	☐	☒	☐	☐	☒	☐
503	CM ₃	4	☒	☐	☒	☐	☐	☒	☒
504	CM ₄	5	☒	☐	☒	☐	☒	☐	☐
505	CM ₅	6	☒	☐	☒	☐	☒	☐	☒
506	CM ₆	7	☒	☐	☒	☐	☒	☒	☐
507	CM ₇	8	☒	☐	☒	☐	☒	☒	☒
508	CM ₈	9	☒	☐	☒	☒	☐	☐	☐
509	CM ₉	10	☒	☐	☒	☒	☐	☐	☒
510	CM ₁₀	11	☒	☐	☒	☒	☐	☒	☐
511	CM ₁₁	12	☒	☐	☒	☒	☐	☒	☒
512	CM ₁₂	13	☒	☐	☒	☒	☒	☐	☐
513	CM ₁₃	14	☒	☐	☒	☒	☒	☐	☒
514	CM ₁₄	15	☒	☐	☒	☒	☒	☒	☐

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	■	■	□	■	□	□	■

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	■	■	■	■	■	■	□

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

that when entry of data is made through the keyboard.

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

CANON PROGRAM CARD									
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.

CANON PROGRAM CARD									
40	20	10	8	4	2	1			
1	■	■	■	□	□	■	□	2	
2	■	■	■	□	□	■	■	3	
3	□	■	■	□	□	■	■	SM ₃	
4	■	■	■	□	□	□	■	1	
5	■	■	■	□	□	■	□	2	
6	□	■	■	□	■	□	□	SM ₄	

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 M₁ 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.

* 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          or   
and store 1 2 3 . 4 6 into respective memories.
- 2) Depress  to recall contents of Memory 2. The recalled contents are 1 2 3.
- 3) Depress  and  to recall contents of Memory 3. The recalled contents are 1 2 3 . 4 6.

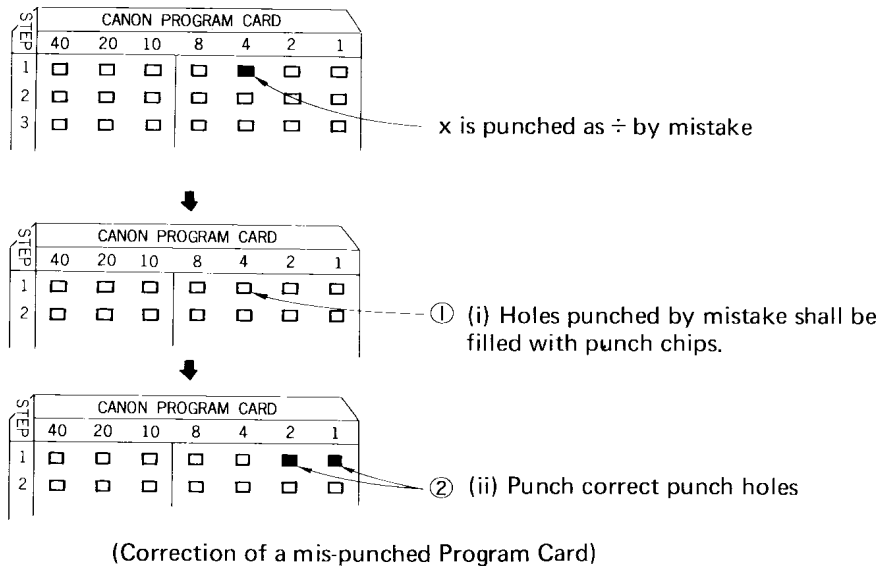
3-15 Correction & Clearing of Program Instruction

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

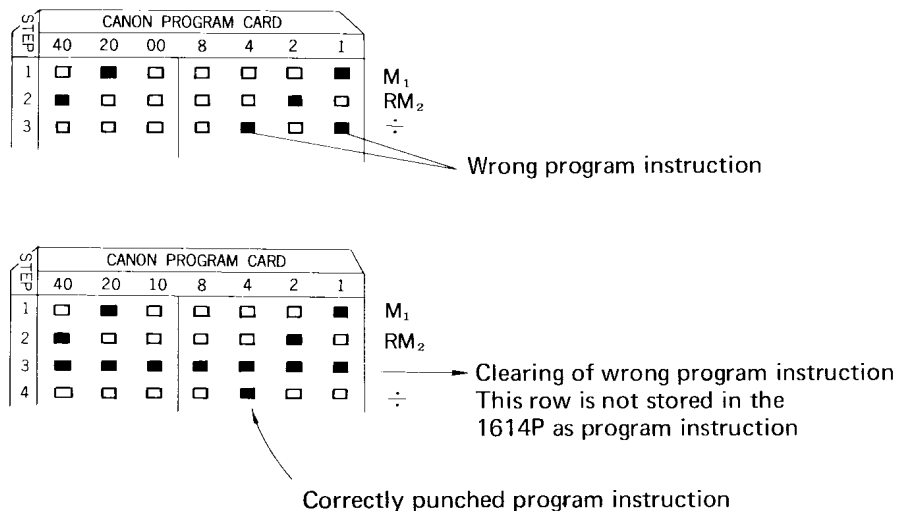
- (i) First, fill the hole punched by mistake with punch chips (punched out pieces).
- (ii) Then, punch correct punch holes.



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

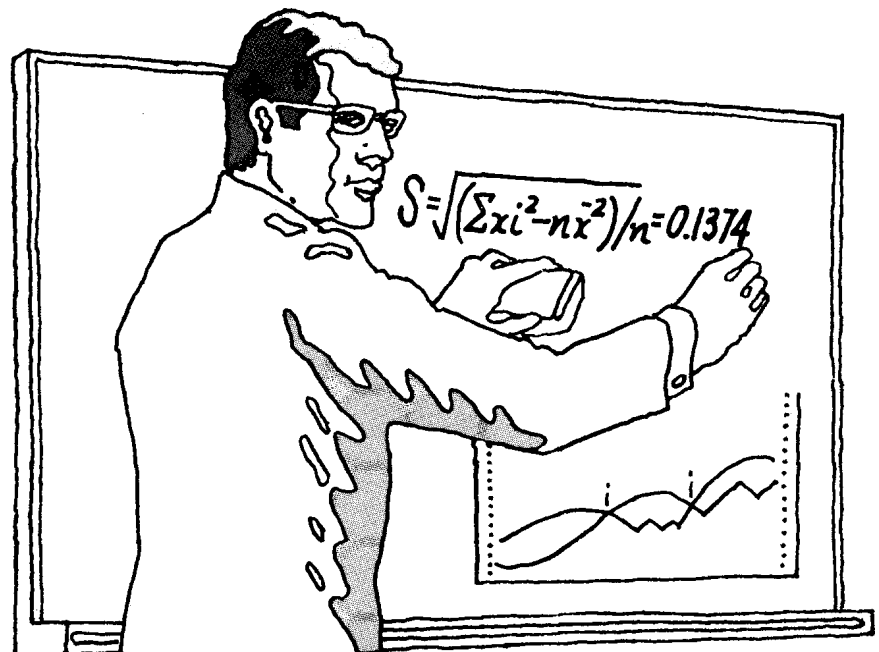
CONCEPT & PROGRAM PREPARATION METHODS WITH ACTUAL EXAMPLES

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 a 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.

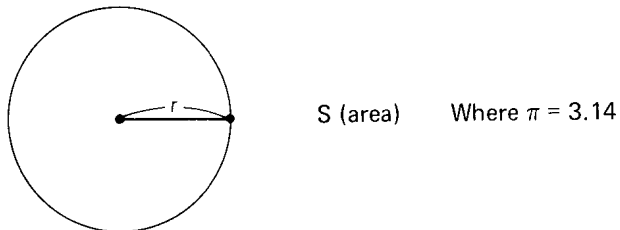


2. Programming

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.



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

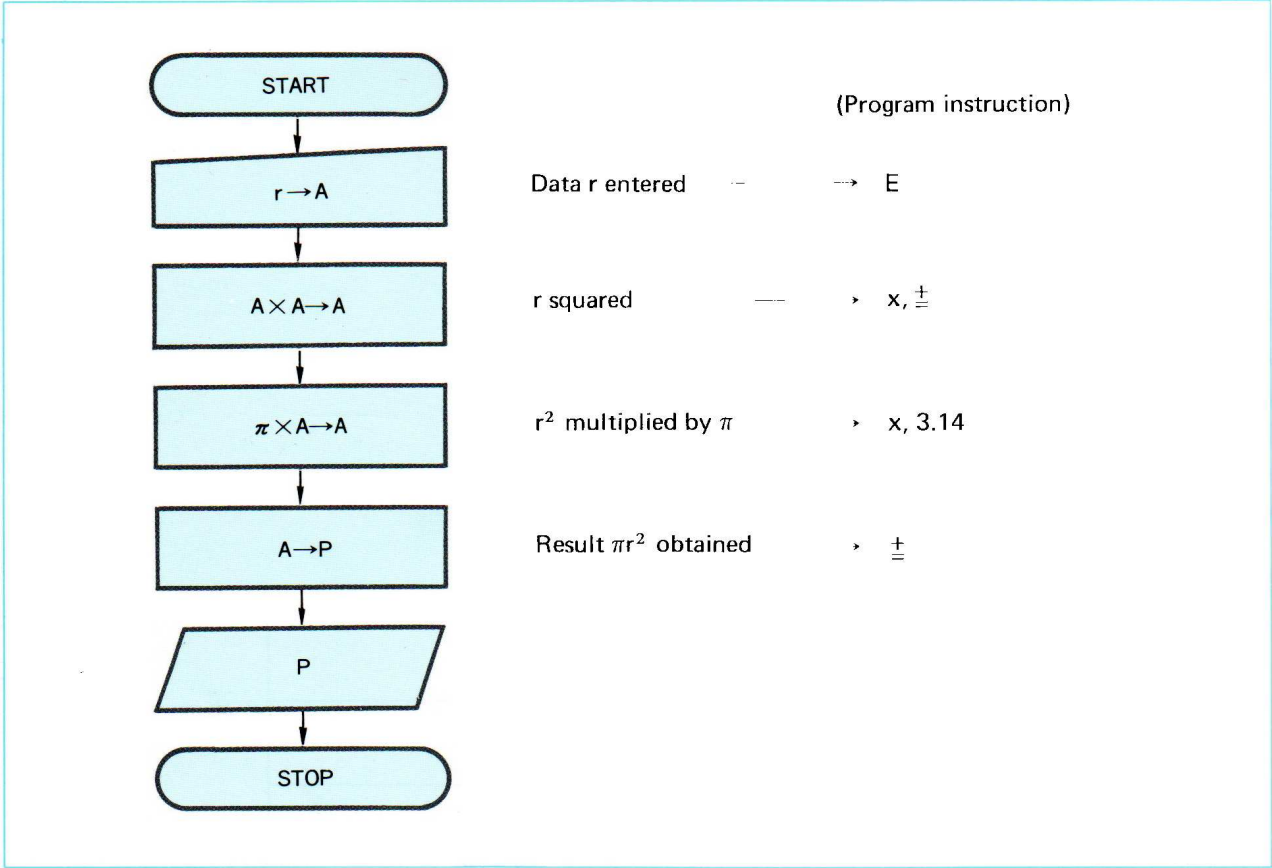
The flow chart is useful as a means of arranging elements of a complex problem in readily understandable form. It shows the problem at a glance.

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.

Step	CANON PROGRAM CARD						
	40	20	10	8	4	2	1
1							
2							
3							
4							
5							
6							
7							
8							
9							

(Code)	(Program Instruction)
10 + 01 = 101	→ E
00 + 03 = 003	→ x
00 + 01 = 001	→ $\frac{+}{=}$
00 + 03 = 003	→ x
70 + 03 = 703	→ 3
60 + 00 = 600	→ .
70 + 01 = 701	→ 1
70 + 04 = 704	→ 4
00 + 01 = 001	→ $\frac{+}{=}$

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

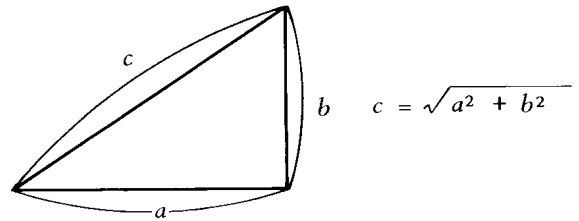
3. Program Using ROUND Instruction (Calculation with Pythagorean theorem)

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 Contents	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			
4	M1	49			Square of b taken and added to memory 1, thereby giving the sum of the squares of a and b in this memory.
5	E	6			
6	X	6			
7	$\pm$	36			
8	M1	36			
9	RM1	85			Sum of the squares of a and b called, ROUND instruction specified, and root extracted.
10	5/4	85			
11	4	85			
12	$\sqrt{\quad}$	9.2195			Square root of $a^2 + b^2$, obtained to 4 decimal places.

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 amortization 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_3 to eliminate the preparation to depress $\boxed{CM} \boxed{3}$ for execution of program calculation. Enter the depreciation rate and store in MEMORY 4 by SM_4 which also functions to clear the previous memory contents.

(ii) Clear MEMORY 5 in advance by CM_5 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.

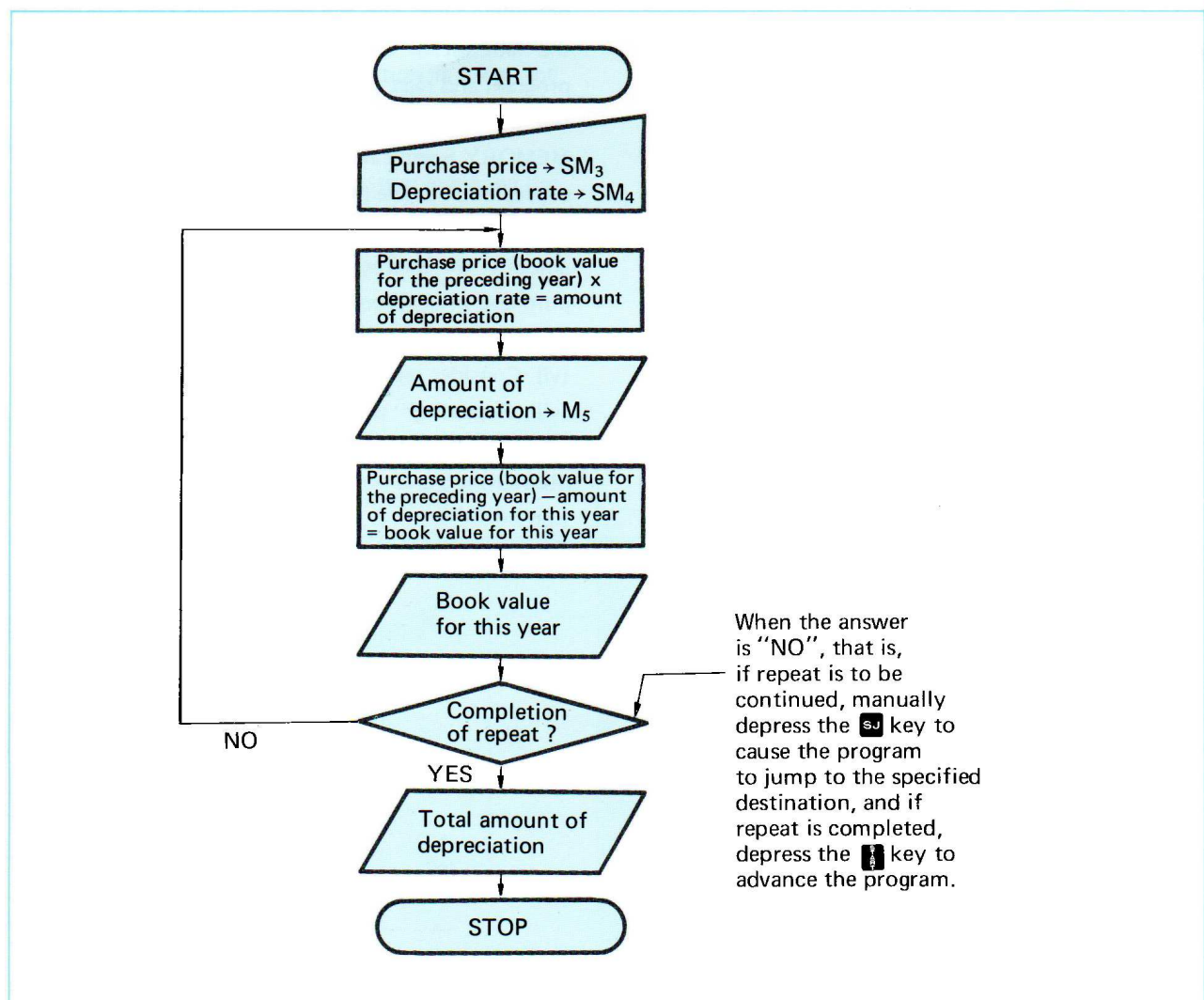
(iv) Store the obtained amount of depreciation in MEMORY 5 and calculate the total of the amounts of depreciation in the memory.

(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 SM₃ E SM₄ CM₅ FJ 1 RM₃ X RM₄ 5/4 O
 ±SM₅ SC M₃ RM₃ SJ 1 RM₅

4) The relationship between punch hole locations 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							(Code)	(Program Instruction)
Step	40	20	10	8	4	2		
1	☐	☐	☒	☐	☐	☐	10 + 01 = 101	E
2	☐	☒	☒	☐	☐	☒	30 + 03 = 303	SM ₃
3	☐	☐	☒	☐	☐	☐	10 + 01 = 101	E
4	☐	☒	☒	☐	☒	☐	30 + 04 = 304	SM ₃
5	☒	☐	☒	☐	☒	☐	50 + 05 = 505	CM ₅
6	☐	☐	☒	☐	☒	☒	10 + 07 = 107	FJ
7	☒	☒	☒	☐	☐	☐	70 + 01 = 701	1
8	☒	☐	☐	☐	☐	☒	40 + 03 = 403	RM ₃
9	☐	☐	☐	☐	☐	☒	00 + 03 = 003	X
10	☒	☐	☐	☐	☒	☐	40 + 04 = 404	RM ₄
11	☒	☒	☐	☐	☒	☐	60 + 06 = 606	5/4
12	☒	☒	☒	☐	☐	☐	70 + 00 = 700	0
13	☐	☐	☐	☐	☐	☒	00 + 01 = 001	+
14	☐	☐	☒	☐	☐	☒	10 + 01 = 101	S
15	☐	☒	☐	☐	☒	☒	20 + 05 = 205	M ₅
16	☒	☒	☐	☐	☐	☒	60 + 03 = 603	SC
17	☐	☒	☐	☐	☐	☒	20 + 03 = 203	M ₃
18	☒	☐	☐	☐	☐	☒	40 + 03 = 403	RM ₃
19	☐	☐	☒	☐	☐	☐	10 + 02 = 102	SJ
20	☒	☒	☒	☐	☐	☐	70 + 01 = 701	1
21	☒	☐	☐	☐	☒	☒	40 + 05 = 405	RM ₅

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 for 5th year
14		35012	Book value for 5th year
15		314988	Total depreciation

* The program which has advanced to step 21 will automatically return to step 1, and is stopped by the E instruction of step 1. If step 1 does not provide a step instruction such as E, the program will advance to the step which provides E,S,SJ, or EJ instruction.

** Carefully notice the process in which the program jumps toward FJ of step 6 by the **SJ** key operation.

*** 1 is used as the jump symbol in this example.

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 (224)		(24,482.80)	(100.34)

Calculation Formula :

Average unit price = total amount ÷ total quantity

(Programming)

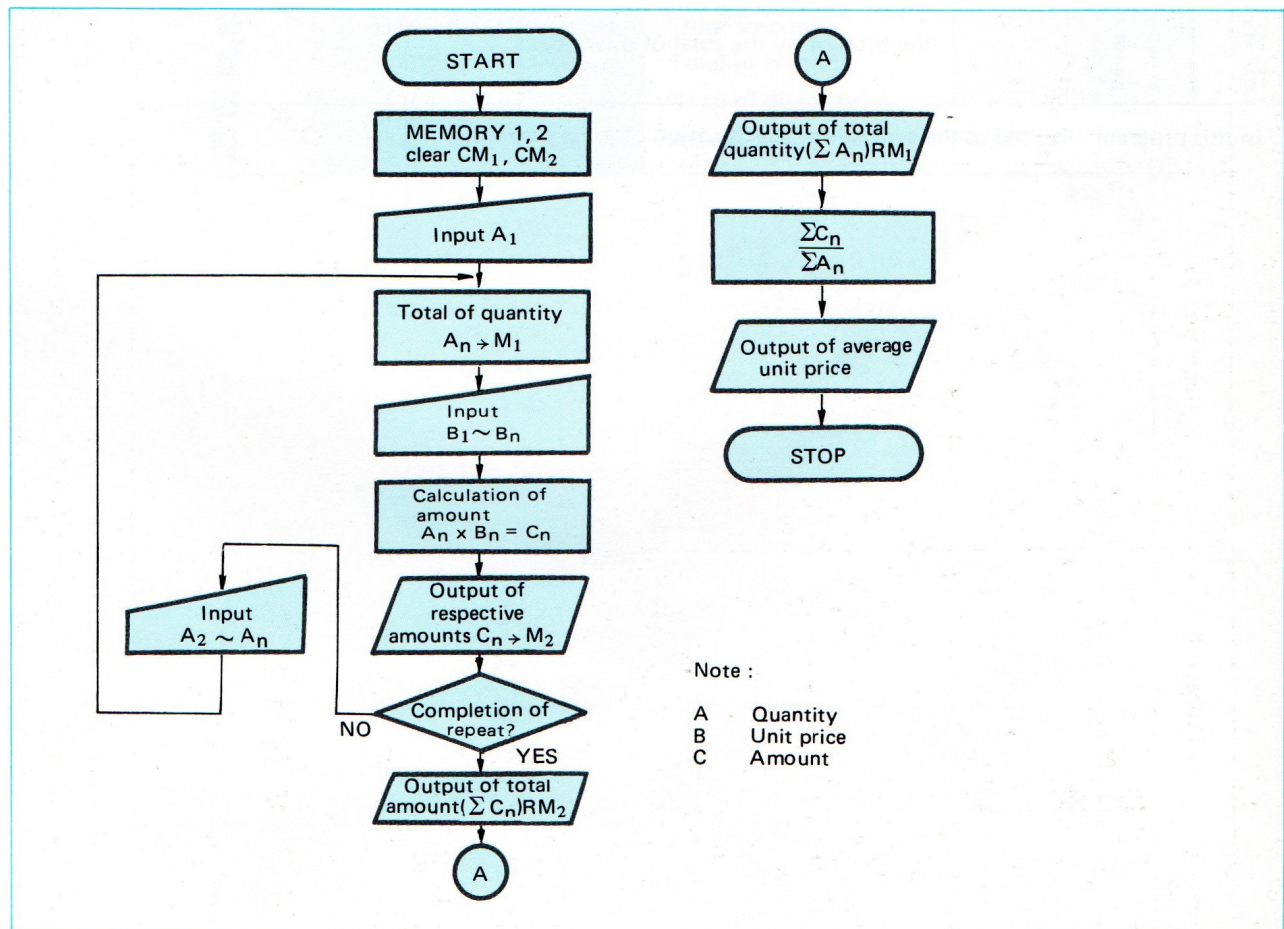
What is Entry Jump to be used for in this example?
Explanation is repeated here in brief.

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 $\frac{\Sigma}{A}$ key operation is accomplished, the program jumps to the specified program step, and if the $\frac{\Sigma}{A}$ 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 $\frac{\Sigma}{A}$ key operation is 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.

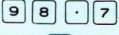
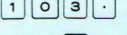
Step	Program instruction	Description of procedure
1	CM ₁	Clear MEMORY 1 and MEMORY 2 to be used for totalling quantity and amount respectively in advance.
2	CM ₂	
3	E	
4	{ FJ	Specified destination of jump when quantity is entered and the  key is depressed.
5	{ * 1	
6	M ₁	Total the quantity.
7	X	Enter unit price. Calculate respective amounts.
8	E	
9	$\pm$	Total the amounts.
10	M ₂	Enter second and subsequent quantities.
11	{ EJ	When entry of quantity is over, cause the program to advance by depressing the  key only.
12	{ * 1	
13	RM ₂	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	$\div$	Recall and read the total of quantity. Calculates the average unit price by dividing the total of by the total of quantity.
16	RM ₁	
17	S	
18	$\pm$	

* 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.

Pro- cedure	Operation	Contents of calculation	ENTRY lamp	Remarks
1		0		Step for manual calculation 3rd step
2		123	Glow	First quantity
3		123	Glow	
4		98.7	Glow	First unit price
5		(12,140.1)	Glow	First amount
6		36	Glow	Second quantity
7		36	Glow	
8		103.2	Glow	Second unit price
9		(3,715.2)	Glow	Second amount
10		85	Glow	Third quantity
11		85	Glow	
12		101.5	Glow	Third unit price
13		(8,627.5)	Glow	Third amount
14		(24,482.8)	Glow	Total of amount
15		(244)	Glow	Total of quantity
16		(100.34)	Glow	Average unit price

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

Factorial $n!$

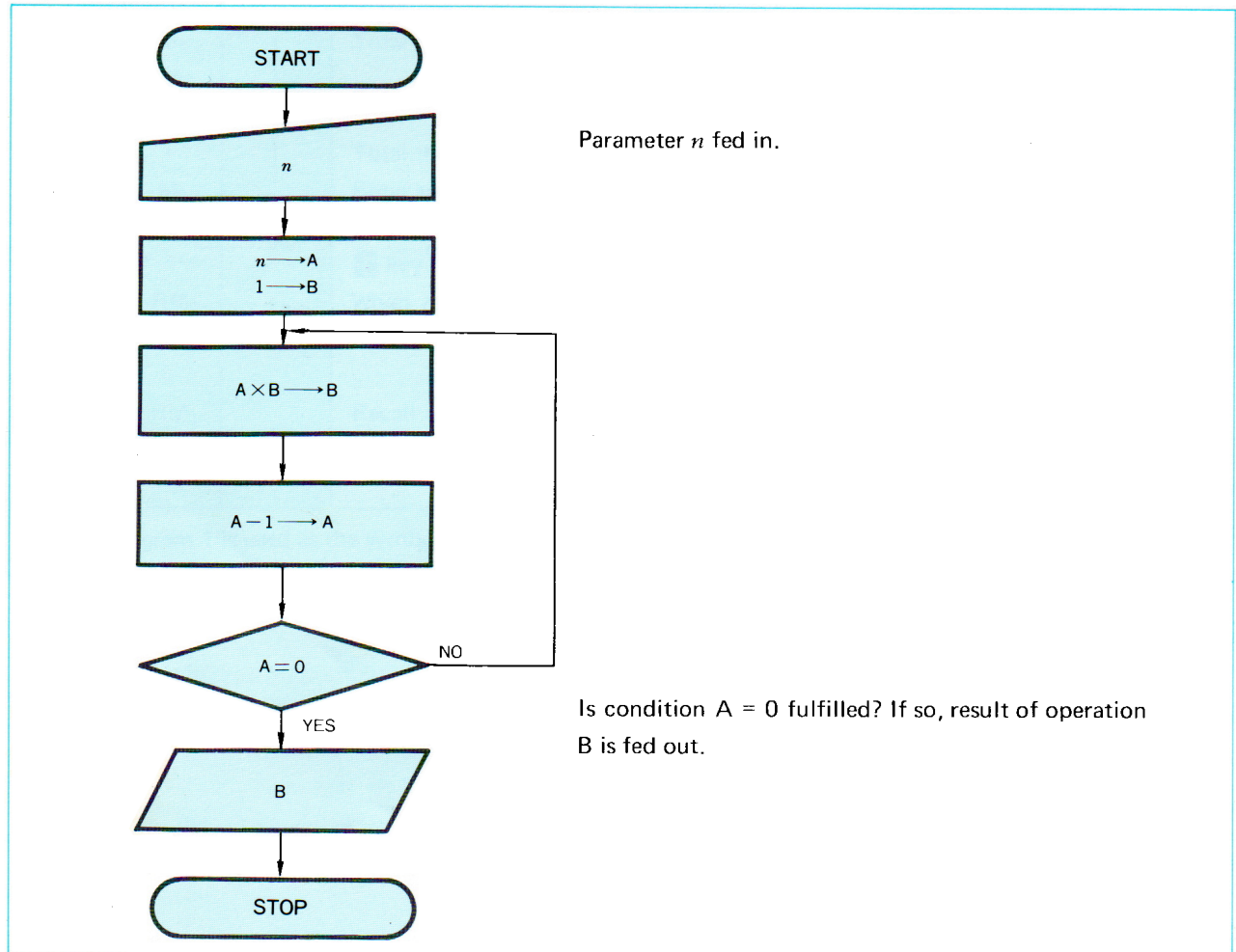
$$n! = n (n - 1) (n - 2) \dots 2 \cdot 1$$

Express the product of natural numbers from 1 to n .

(Programming)

1) The procedure involves the input of parameter n and determination of the result. It is necessary to establish memories for interim operation results and storage of $(n - 1)$. Operation is terminated when $(n - 1) = 0$ and the result of the operation is displayed.

2) Flow chart



Expressing the above flow in program instruction yields the following : E SM₃ 1 SM₄ FJ 1 RM₃ x RM₄

$\pm$ SM₄ 1 SC M₃ RM₃ SC MJ 1 RM₄

As n is a natural number, ROUND instruction is not used.

Execution of program calculation

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 a flowing fluid)

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

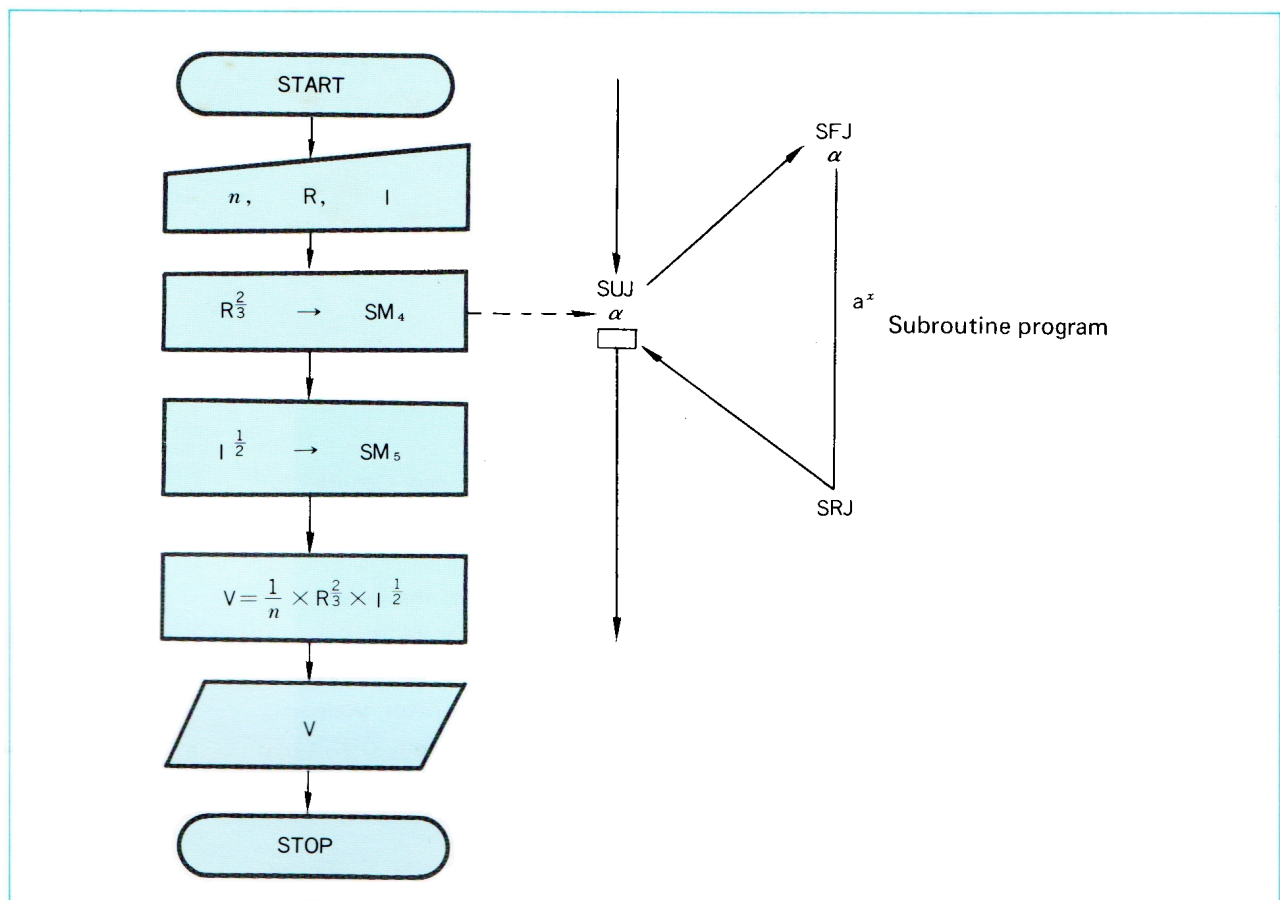
$$\text{Equation : } V = \frac{1}{n} R^{\frac{2}{3}} I^{\frac{1}{2}}$$

V : Rate of flow

(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) SM₃ E(R) SM₄ E(I) SM₅ RM₄ SM₁₄ 2 ÷ 3 ± SM₁₃ SUJ (α) X RM₅ ÷ RM₃ 5/4 1 ± 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)

Procedure	Key Operation	Display	Remarks
1		0	Manual calculation area → Program calculation area
2	 	30	Input of n
3		30	
4	   	.747	Input of R
5		.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 calculation.

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



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