Congratulations on your purchase of this programmable scientific calculator.
The extra-thin, pocket book-sized electronic calculator with liquid crystal display is capable of performing ordinary step-by-step and programmed calculations. Equipped with 22 independent memories, it will compute values of 51 kinds of functions. The sequence of operation is basically the same as ordinary mathematical expressions (true algebraic logic) with parentheses nestable up to ten levels. For programmed operation, it can store a program or programs of up to a total of 256 steps.
The FA-1 adapter, provided optionally, allows to connect a cassette tape recorder to the calculator and store programs and contents of the memories in cassette tapes for later reloading.

This manual consists of Part 1, step-by-step calculation, and Part 2, programmed calculation. It is suggested that you get the thorough knowledge of operation of the calculator ...e.

INDEX

The calculator has undergone severe quality control and inspection.
It is advised that you take note of the following for its longer service life.

Precautions

- The calculator is composed of delicate electronic parts. Never disassemble it. Never expose it to shocks and drastic temperature changes. Do not leave or store it in hot, damp or dusty places. When the ambient temperature is very low, the display may be slow to work or remain dark. This kind of trouble will be removed as soon as it is put in normal temperature.
- Do not connect any device other than the FA-1 to the adapter jack.
- When the calculator is working, the display shows "–" sign and the keyboard is insensitive during this time. Make it a rule to look at the display before depressing a key.
- Replace the electrolytic batteries once a year even if the calculator is not used. If dead batteries are left in the calculator, leakage will occur, leading to trouble. Never leave dead batteries in the calculator.
- To clean the calculator, wipe it with a piece of dry, soft cloth soaked with neutral detergent and squeezed tight. Do not use solvents such as thinner, benzine, etc.
- For servicing contact your retailer or nearby dealer.

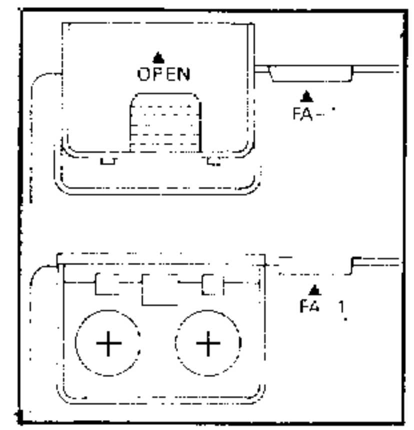
Battery maintenance

The calculator uses two silver oxide batteries (type: G-13).
As the batteries discharge, figures of the display grow dim. Batteries should then be renewed. Even when the calculator works properly, replace the batteries once a year.

- How to replace the batteries

(1) Turn off the power switch. Remove the battery compartment lid on the bottom.
(2) Remove the batteries. (Let the battery compartment face downward and tap the body lightly.)
(3) Clean new batteries with dry cloth. (Dirty surfaces of the batteries will result in contact failure.) Install them with the positive electrodes facing upward.
(4) Pressing the compartment lid on the batteries, slide it to close the compartment.

Note: After replacing batteries confirm that contents of memories and/or programs remain or not.



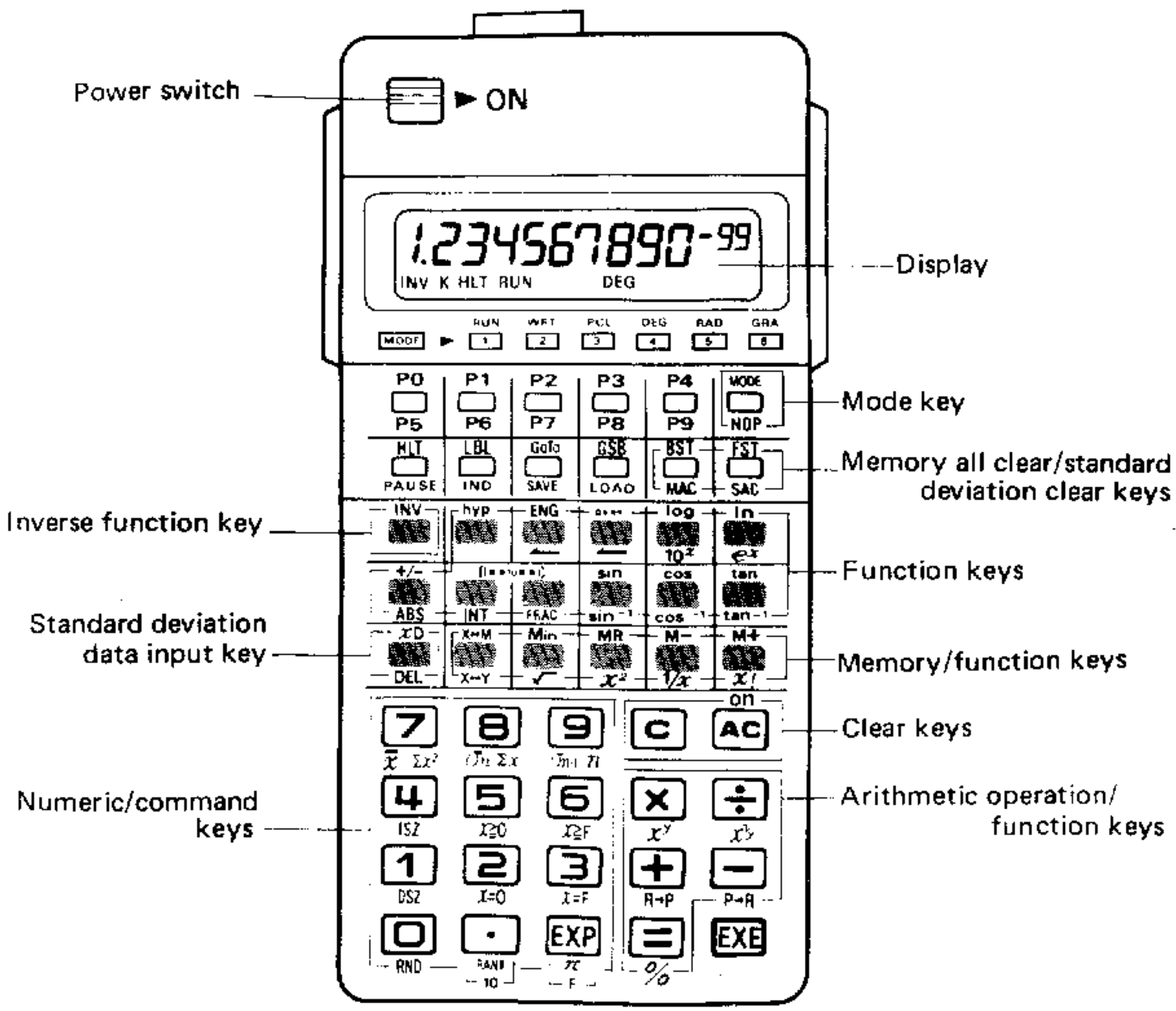
FA-1 adapter

Programs and data stored in the registers can be moved into cassette tape and reloaded later back into the calculator.
For this purpose, prepare the FA-1 adapter (option) and a cassette tape recorder which is provided with MIC (microphone) and EAR (earphone) or MONITOR jacks like an ordinary radio-cassette tape recorder.
With this system consisting of the calculator, FA-1, and a cassette tape recorder, you can listen to music synthesized and recorded by the calculator.
Refer to the instruction manual of the FA-1 adapter for details.

PART-1
Step-by-step calculation

Part 1 describes step-by-step calculation exclusively. (Refer to Part 2 for programmed calculation.)

1-1. Controls and components required for step-by-step calculation

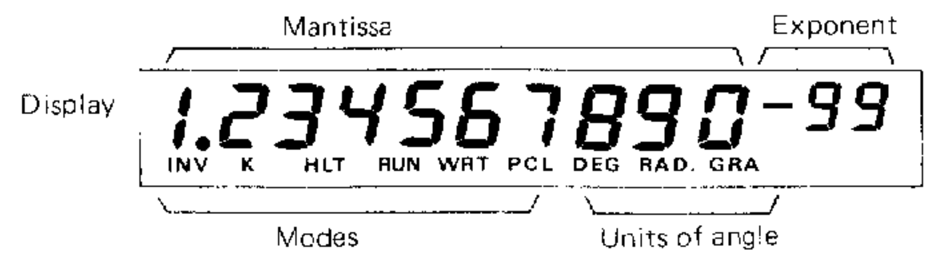


- **Register structure**

Register	Use
X-register (display)	Used for arithmetic operations and functional calculation
Y-register (arithmetic operations)	
L_1 - register	Used for a series of calculations with priority given implicitly (multiplication and division to addition and subtraction) or explicitly parentheses
L_2 - register	
⋮	
L_9 - register	
L_{10} - register	
M_0 - register	22 independent memories (M0 thru M9, M·0 thru M·9, MF and M·F) (M7, M8 and M9 are also used for calculation of standard deviation).
M_1 - register	
⋮	
M_7 - register (Σx^2)	
M_8 - register (Σx)	
M_9 - register (n)	
M·0 - register	
M·1 - register	
⋮	
M·8 - register	
M·9 - register	
MF - register	
M·F - register	

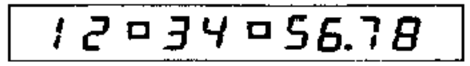
Power switch

Slide the knob to the right to turn on.
Contents of the M-registers are preserved even when the power switch is turned off.



The display shows input data, intermediate results, and resutls of operation. The mantissa section displays up to 10 digits (9 for negative numbers) with trailing zeros suppressed. The exponent section displays up to ±99. This remains dark when a number is displayed in the fixed point representation.
Sexagesimal numbers will be displayed as follows.

Display of a sexagesimal numer



(12°34'56.78" is meant)

The lowest position of the mantissa section may appear as "E" (error, see page 9) and "–" (calculator is working). A unit of angle, DEG, RAD, or GRA, which is currently effective will be displayed as necessary. The mode section will display INV when the [INV] key is depressed, K while calculation involving constants is in progress, and RUN while execution is in progress. For the modes of HLT, WRT, and PCL, refer to Part 2.

[MODE] : Mode key

Use the [MODE] key together with a numeric key of [1] to [5] to select a specific mode of operation or angle. (Symbolized by [MODE] in this manual.)

[MODE] [1] The display reads RUN. The calculator is ready for step-by-step and programmed operation.

[MODE] [2] Reads WRT. Ready for inputting and checking programs.

[MODE] [3] Reads PCL. Ready for erasing programs.

* Refer to Part 2 for the WRT and PCL modes.

[MODE] [4] Reads DEG. Angles will be given in degrees.

[MODE] [5] Reads RAD. Angles will be given in radians.

[MODE] [6] Reads GRA. Angles will be given in gradients.

(**Note:** 90 degrees = $\pi/2$ radians = 100 gradients)

* The RUN and DEG modes will be selected automatically after the power switch is turned on.
Another unit of angle can be selected only in the RUN mode.

[INV] : Inverse function key

To designate an inverse function, depress [INV] and the function key (marked by a brown symbol on the keyboard).
Depress [INV] and the display reads INV. Depress [INV] again to release the INV mode (the INV sign disappears).

[0] through [9], [0], [.]: Numeric and decimal point keys

- Input numbers by depressing these keys just in the order in which you write them normally (starting from the most significant digit).

* Numbers of up to 10 digits (9 digits for negative numbers and fractions) will be effective and additional digits will be simply disregarded.

- [INV], [RND] : To round a number
[INV] [RND] [4], for example, rounds the number currently displayed up if the fifth digit is greater than four and off otherwise into a four-digit number.
- [INV] [RAN#]: To generate a random number
A random number of 0.000 to 0.999 will be generated.
- [10] : To designate a register of M·0 to M·9 and M·F when depressed following a memory key ([X↔M], [Min], [MR], [M-] or [M+])
Ex. Recall the contents of M·6 [MR] [10] [6]

- [INV][x̄] : To calculate mean value ($\bar{x}$)
- [MR][Σx²] : To calculate square sum (Σx^2)
- [INV][σn] : To calculate standard deviation (σ_n)
- [MR][Σx] : To calculate sum (Σx)
- [INV][σn-1] : To calculate standard deviation (σ_{n-1})
- [MR][n] : To obtain the number of data (n)

Used to obtain standard deviation. Refer to "Calculation of standard deviation" for details.

[EXP/π] : Exponent/pi entry key

- To input a floating-point number, separate the mantissa and exponent with [EXP].
Ex. Enter 2.56×10^{34} 2 [.] 56 [EXP] 34

* The exponent is up to ±99. If an exponent which is outside the range is input, only the last two digits are effective.

- Depress [π] to input the value of π.
If it is depressed following [INV], the value of π will be displayed.
- [EXP/F] : Depress it following a memory key ([X↔M], [Min], [MR], [M-] or [M+]) to designate the MF register.
Ex. Add 25 to the contents of MF register 25 [M+] [EXP/F]

[+], [–], [×], [÷], [=] : Arithmetic operation/equal keys

- Use these keys to perform arithmetic operations. The sequence of operations is just the same as ordinary mathematical expressions. Depress [=] to obtain the final result. If one of these keys is depressed twice in succession, the preceding number will be temporarily memorized as a constant. (See page 12)
- [INV][xʸ] : Command for raising to power
Use to raise x (positive number) to the y (arbitrary number)-th power.
- [INV][x^(1/y)] : Command for extracting root
Use to obtain the y-th root of x.
- [INV][R→P] : Command of coordinate transformation
Use to transform rectangular coordinates to polar ones.
- [INV][P→R] : Command of coordinate transformation
Use to transform polar coordinates to rectangular ones.
- [INV][%] : Percent command
Use to obtain regular percentages, and-ons, discounts, ratios and increase/decrease values.

[+/–, ABS] : Sign change/absolute value key

- Use this key to reverse the sign of the number currently displayed (from positive to negative or vice versa). It reverses the sign of the exponent when pressed following [EXP].
- Use the sequence of [INV] and [ABS] to obtain the absolute value of the number currently displayed.

[(, INT] : Parenthesis/integer key

- Opening parenthesis nestable up to ten levels.
- Use the sequence of [INV] and [INT] to round off the decimal part of the number currently displayed.

[), FRAC] : Parenthesis/fraction key

- Closing parenthesis nestable up to ten levels.
- Use the sequence of [INV] and [FRAC] to extract the decimal part of the number currently displayed.

[X↔M, X↔Y] : Register data exchange key

- Use this key along with a numeric key or [EXP/F] to exchange the contents of the X-register (number displayed) and an M-register.
Ex. To exchange the contents of X- and M5 register [X↔M] [5]
- Use the sequence of [INV] and [X↔Y] to exchange the contents of the X- and Y-register (number in the working register).

[Min/√] : Memory in/square root key
- Use this key along with a numeric key or "F" to put the displayed number in an M-register. Old data held in the register will be automatically erased.
- Use the sequence of [INV] and [√] to obtain the square root of the number displayed.

[MR/x²] : Memory recall/square key
- Use this key along with a numeric key or "F" to display the contents of an M-register. This command does not change the contents of the register.
- Use the sequence of [INV] and [x²] to obtain the square of the number displayed.

[M−/1/x] : Memory minus/reciprocal key
- Use this key along with a numeric key or "F" to subtract the number displayed from the contents of an M-register. (This key does not have the function of [=].)
- Use the sequence of [INV] and [1/x] to obtain the reciprocal of the number displayed.

[M+/x!] : Memory plus/factorial key
- Use this key along with a numeric key or "F" to add the number displayed to the contents of an M-register. (This key does not have the function of [=].)
- Use the sequence of [INV] and [x!] to obtain the factorial of the number currently displayed.

[xD/DEL] : Standard deviation data input/delete key
- Data input key for standard deviation.
- Use the sequence of [INV] and [DEL] to delete a data which has been input during the calculation of standard deviation.

[sin/sin⁻¹] [cos/cos⁻¹] [tan/tan⁻¹] : Trigonometric funcion/inverse trigonometric function keys
- Use one of these keys to obtain the value of the corresponding trigonometric function with the number currently displayed taken as the argument.
- Use the sequence of [INV] and one of these keys to obtain the value of the corresponding inverse trigonometric function.

[hyp] : Hyperbolic function key
- Use the sequence of [hyp] and [sin], [cos] or [tan] to obtain the value of the corresponding hyperbolic function with the number currently displayed taken as the argument.
- Use the sequence of [INV], [hyp] and [sin], or [hyp], [INV] and [sin], for example, to obtain the value of $\sinh^{-1}$ with the number displayed taken as the argument.

[ENG] : Engineering key
Use this key to have a floating-point representation of the number displayed as illustrated below.

12.3456	12.3456
1st [ENG]	1.23456 01
2nd [ENG]	12.3456 00
3rd [ENG]	12345.6 −03
4th [ENG]	12345600 −06
5th [ENG]	12345600 −06 (Unchanged)

12.3456	12.3456
1st [INV][ENG]	1.23456 01
2nd [INV][ENG]	0.0123456 03
3rd [INV][ENG]	0.000012345 06
4th [INV][ENG]	0. 00000012 09
5th [INV][ENG]	0. 00000012 09 (Unchanged)
[ENG]	0.000012345 06
[ENG]	0.0123456 03

As illustrated above, when [ENG] is depressed first, the displayed number will be converted to its basic floating point representation (that is, normalized to 10). Subsequent depression of [ENG] or the sequence of [INV] and [ENG] causes the exponent to decrease or increase in units of 3.

[°'"] : Sexagesimal key (decimal-to-sexagesimal conversion key)
- Use this key as the unit key of sexagesimal data such as angle and time.
  Ex. 78°45'12" 78 [°'"] 45 [°'"] 12 [°'"]
- Use the sequence of [INV] and [°'"] to convert the decimal representation of angle or time to the sexagesimal representation.

[log/10ˣ] : Common logarithm/power-of-10 key
- Use this key to obtain the common logarithm of the number displayed.
- Use the sequence of [INV] and [10ˣ] to obtain the value of 10 raised in power by the number displayed (i.e. inverse common logarithm).

[ln/eˣ] : Natural logarithm/power-of- *e* key
- Use this key to obtain the natural logarithm of the number displayed.
- Use the sequence of [INV] and [eˣ] to raise "*e* (= 2.7182818...)" to the power determined by the displayed number (i.e. inverse natural loagrithm).

[ON/AC] : All clear key
- Use this key to clear all registers (including the X-, Y- and parenthesis registers) but the M-registers.
- Depress this key to release the "error status" when "*E.*" is displayed. (See page 9)
- Depress this key to resume power which is automatically turned off by the auto power-off function. (See page 9)

[C] : Clear key
Depress this key to erase a data which has just been input erroneously.

[BST/MAC] : Step back/memory all clear key

- Refer to Part 2 for [BST].
- Use the sequence of [INV] and [MAC] to clear all M-registers.

[FST/SAC] : Step forward/standard deviation calculation all clear key

- Refer to Part 2 for [FST].
- Use the sequence of [INV] and [SAC] to clear registers M7, M8, and M9 before beginning calculation of standard deviation.

* **Refer to Part 2 for other keys which are not described yet. They are not needed for step-by-step calculation.**

1-2. Operational remarks

Turn on the power switch before starting operation. Put it in the RUN mode, if not so, by depressing [MODE] and [1].
The unit of angle displayed has nothing to do with calculation unless it involves angular arguments.

■ Correction

- You can erase input data only before a command key is depressed. Depress [C] to erase data and input the correct one.
- Similarly you can correct intermediate results (e.g. the value of a function or operation with nested parentheses). Depress [C] to erase the erroneous result and follow the related steps of calculation again.
- You can correct [+], [−], [×], [÷], [INV], [x^y] and [INV], [$x^{1/y}$], if depressed erroneously. Depress the correct key immediately after the wrong one. Note that the priority of operation remains uncorrected.

■ Overflow or error

If error is detected during operation, "*E.*" or "*C.*" will be displayed in the lower part of the display and the calculator will come to a halt.
The following cases will be detected as error.

1. A result or intermediate result (of arithmetic, functional, or standard deviation operation) or the contents of an M-register has exceeded the range of ±9.999999999 x 10^{99}. (The M-register holds the data which is stored before overflow occurs.)
2. An argument of a function has exceeded its predetermined range. (See page 46)
3. Improper operation is attempted in the calculation of standard deviation. (Ex. You attempt to calculate $\bar{x}$ or σ_n when n=0.)

* "*E.*" will be displayed in these cases. Depress [AC] to release the error status and go back to the first step.

4. The number of levels of nested operation (with parentheses, multiplication, division, x^y and $x^{1/y}$) exceeds 10. (Ex. 2 [+] 3 [×] . . . is attempted after [(] is depressed ten times.)

* "*C.*" will be displayed in this case. If you depress [C] at this time, the intermediate result obtained just before the error occurs will be displayed and you can continue operation with it. Otherwise you can depress [AC] to go back to the first step.

* Underflow (a value smaller than ±1 x 10^{-99}) is not regarded as error but as 0.

■ Auto power-off function

If you leave the calculator unoperated for about 14 minutes (except during programmed calculation), the calculator will be automatically turned off. This prevents wasting power of the batteries. To turn it on, depress [AC] or re-operate the power ON-OFF switch.
(Contents of the M-registers and programs will be preserved even after the auto power-off function works.)

1-3. Basic operations

■ Addition, subtraction, multiplication and division

- The sequence of operation is just the same as ordinary mathematical expressions.

EXAMPLE	OPERATION	READ-OUT
23 + 4.5 − 53 = −25.5	23 [+] 4 • 5 [−] 53 [=]	−25.5
56 ×(−12)÷(−2.5)= 268.8	56 [×] 12 [+/−] [÷] 2 • 5 [+/−] [=]	268.8

* To input a negative number, depress the number and then [+/−].

EXAMPLE	OPERATION	READ-OUT
12369 × 7532 × 74103 = 6.903680612 × 10^{12} (= 6903680612000)	12369 [×] 7532 [×] 74103 [=]	6.903680612 12
1.23 ÷ 90 ÷ 45.6 = 2.997076023 × 10^{-4} (= 0.0002997076023)	1 • 23 [÷] 90 [÷] 45 • 6 [=]	2.997076023−04

* The result will be displayed in the floating-point representation if it is large than 10^{10} or smaller than 10^{-2}

EXAMPLE	OPERATION	READ-OUT
(4.5 × 10^{75})×(−2.3 × 10^{-78}) = −0.01035	4 • 5 [EXP] 75 [×] 2 • 3 [+/−] [EXP] 78 [+/−] [=]	−0.01035
= −1.035 × 10^{-2}	* To convert to floating-point number [ENG]	−1.035−02

* The result is displayed in the fixed-point representation if it is between 10^{-2} and 10^{10}. Depress [ENG] to convert it to the floating-point number. (See page 8.)

EXAMPLE	OPERATION	READ-OUT
(2 × 10^5)÷(−7)= −28571.4285	2 [EXP] 5 [÷] 7 [+/−] [=]	−28571.4285[714]
(2 × 10^5)÷(−7)+ 28571 = −0.4285714	(Continue) [+] 28571 [=]	−0.4285714

* Arithmetic operations will be made over 12-digit mantissa which are held in the registers.

- Priority is given to multiplication and division over addition and subtraction in mixed operations.

EXAMPLE	OPERATION	READ-OUT
3 + 5 × 6 = 33	3 [+] 5 [×] 6 [=]	33.
7 × 8 − 4 × 5 = 36	7 [×] 8 [−] 4 [×] 5 [=]	36.
1 + 2 − 3 × 4 ÷ 5 + 6 = 6.6	1 [+] 2 [−] 3 [×] 4 [÷] 5 [+] 6 [=]	6.6

■ Parenthesis calculations

- Parentheses can be nested up to ten levels.
- The display will be cleared to zero when an opening parenthesis is input. With a closing parenthesis input, the intermediate result of that level will be displayed.

EXAMPLE	OPERATION	READ-OUT
$100-\underline{(2+3)\times 4}=80$	100[−][(]2[+]3[)]	5.
	(Continue) [×]4[=]	80.

* The first opening parenthsis and the closing parenthesis preceding [=] may be omitted even if they are nested.

EXAMPLE	OPERATION	READ-OUT
$(2+3)\times(4+5)=45$	2[+]3[)][×][(]4[+]5[=]	45.
$10-\{2+7\times(3+6)\}=-55$	10[−][(]2[+]7[×][(]3[+]6[=]	−55.
$\{(2+3)\times 4-(5+6)\times 3\}\times 2=-26$	2[+]3[)][×]4[−] [(]5[+]6[)][×]3[)][×]2[=]	−26.
$\frac{2\times 3+4}{5}=(2\times 3+4)\div 5=2$	2[×]3[+]4[)][÷]5[=]	2.

* In the above example, [=] may be used instead of [)].

EXAMPLE	OPERATION	READ-OUT
$\frac{2}{3}(\frac{8}{10}-\frac{1}{2})=0.2$	2[÷]3[×][(]8[÷]10[−]1[÷]2[)][=]	0.2

* The fractional expression should be converted to the primary expression using parentheses.

EXAMPLE	OPERATION	READ-OUT
$\frac{5\times 6+6\times 8}{15\times 4+12\times 3}=0.8125$ $=(5\times 6+6\times 8)\div(15\times 4+12\times 3)$	[(]5[×]6[+]6[×]8[)][÷] [(]15[×]4[+]12[×]3[)][=]	0.8125
$(1.2\times 10^{19})-\{(2.5\times 10^{20})\times\frac{3}{100}\}=4.5\times 10^{18}$	1•2[EXP]19[−][(]2•5[EXP]20[×]3[÷]100[)][=]	4.5 18
$\frac{6}{4\times 5}=0.3$	4[×]5[÷]6[INV][x↔y][=]	0.3

* The above sequence of operation is equivalent to 6[÷][(]4[×]5[)][=] and 6[÷]4[÷]5[=].

■ Constant calculations

- Depress an arithmetic operation key twice (or even times) in succession. And the number which is currently displayed will be reserved as a constant, indicated by "K" in the display. It need not be reinput in the succeeding calculations.

EXAMPLE	OPERATION	READ-OUT	
$12\underline{+23}=35$	23[+][+]12[=]	35.	
$45\underline{+23}=68$	45[=]	68.	
$(-78)\underline{+23}=-55$	78[+/−][=]	−55.	
$7\underline{-5.6}=1.4$	5•6[−][−]7[=]	1.4	
$2.9\underline{-5.6}=-2.7$	2•9[=]	−2.7	
$(8.5\times 10^{3})\underline{-5.6}=8494.4$	8•5[EXP]3[=]	8494.4	
$2.3\underline{\times 12}=27.6$	12[×][×]2•3[=]	27.6	
$(-4.56)\underline{\times 12}=-54.72$	4•56[+/−][=]	−54.72	
$\frac{5}{8}\underline{\times 12}=7.5$	[(]5[÷]8[)][=]	7.5	
$78\underline{\div 9.6}=8.125$	9•6[÷][÷]78[=]	8.125	
$(1.2\times 10^{15})\underline{\div 9.6}=1.25\times 10^{14}$	1•2[EXP]15[=]	1.25 14	
$45\underline{\div 9.6}=4.6875$	45[=]	4.6875	
$\underline{3\times 6}\times 9=162$	3[×]6[×][×]9[=]	162.	
$\underline{3\times 6}\times 8=144$	8[=]	144.	
$\underline{3\times 6}\times(5+6)=198$	[(]5[+]6[)][=]	198.	
$\{(1.\dot{1}^{3})^{2}\}^{2}=3.138428376$	1•1[×][×][=][=]	1.331	(3rd power)
	[×][=][×][=]	3.138428376	
$17+17+17+17=68$	17[+][+][=][=][=]	68.	
$8+8+8+11+11+11=57$	8[+][+][=][=][+][+]11[INV][x↔y][=][=][=]	57.	
$50-3.6-3.6-3.6-3.6=35.6$	50[−][−]3•6[INV][x↔y][=][=][=][=] (or 3•6[−][−]50[=][=][=][=])	35.6	

* [÷][÷] and [−][−] are equivalent to [INV][x↔y].

EXAMPLE	OPERATION	READ-OUT	
$\frac{56}{4\times(2+3)}=2.8$	4[×][(]2[+]3[)][÷][÷]	20.	(Divider)
	56[=]	2.8	

■ Memory calculations

- The calculator is equipped with 22 registers, M0, M1 through M9, M·0, M·1 through M·9, MF and M·F. These registers are available for you with the aid of [x↔M], [Min], [MR], [M−], [M+], [EXP/F] and numeric keys.
- Contents of the registers are preserved even after the power switch is turned off. Depress [Min] and [MAC] to clear all the M-registers.

EXAMPLE	OPERATION	READ-OUT
23 + 9 = 32	23 [+] 9 [=] [Min] [1]	32.
53 − 6 = 47	53 [−] 6 [=] [M+] [1]	47.
−) 45 × 2 = 90	45 [×] 2 [=] [M−] [1]	90.
99 ÷ 3 = 33	99 [÷] 3 [=] [M+] [1]	33.
22	[MR] [1]	22.

* Depress [Min] to put a data into an M-register which automatically clears the previously stored data. (The register need not be cleared in advance.)
[M+] and [M−] do not have the function of [=].

EXAMPLE	OPERATION	READ-OUT
7 + 7 + 7 + (2 × 3) + (2 × 3) + (2 × 3) − (2 × 3) = 33	7 [Min] [EXP/F] [M+] [EXP/F] [M+] [EXP/F] 2 [×] 3 [=] [M+] [EXP/F] [M+] [EXP/F] [M+] [EXP/F] [M−] [EXP/F] [MR] [EXP/F]	33.
7 + 8 + 9 = 24	7 [Min] [1] [+] 8 [Min] [2] [+] 9 [Min] [3] [=] [Min] [4]	24.
4 + 5 + 6 = 15	4 [M+] [1] [+] 5 [M+] 2 [+] 6 [M+] [3] [=] [M+] [4]	15.
3 + 6 + 9 = 18	3 [M+] [1] [+] 6 [M+] [2] [+] 9 [M+] [3] [=] [M+] [4]	18.
14 19 24 57	[MR] [1]	14.
	[MR] [2]	19.
	[MR] [3]	24.
	[MR] [4]	57.
7 × 4 × 12.3 = 344.4	7 [×] 4 [×] 1 2 • 3 [Min] [1] [=]	344.4
− 12.3 × (8 + 5) = − 159.9	[MR] [1] [+/−] [×] [(] 8 [+] 5 [)] [=]	− 159.9
(12.3 + 6) × 9 = 164.7	[(] [MR] [1] [+] 6 [)] [×] 9 [=]	164.7
12 × (2.3 + 3.4) − 5 = 63.4	12 [×] [(] 2 • 3 [+] 3 • 4 [)] [Min] [1] [−] 5 [=]	63.4
30 × (2.3 + 3.4 + 4.5) − 15 × 4.5 = 238.5	4 • 5 [M+] [1] [x↔M] [1] [×] 30 [−] 1 5 [×] [MR] [1] [=]	238.5

(At [x↔M]:) To exchange 4.5 displayed with the contents of the M1 register.

Note: When a memory key is used, data in the registers consist of 10-digit mantissa.

EXAMPLE	OPERATION	READ-OUT
$(2 \times 10^5) \div (-7) + 28571 = -0.42857$	2 [EXP] 5 [÷] 7 [+/−] [=] [Min] [1]	− 28571.4285 [7]
(Same example as in page 10)	(Continue) [+] 28571 [=]	− 0.42857
	[MR] [1] [+] 28571 [=]	− 0.42857

1-4. Function calculations

- To obtain the value of a function, input an argument and then depress the function key.
- Calculation of functions can be mixed with ordinary arithmetic operations nested by parentheses.
- Refer to the specifications, given in page 46, for the accuracy and arguments of functions.

■ Trigonometric functions (sin, cos, tan) and inverse trigonometric functions ($\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$)

EXAMPLE	OPERATION	READ-OUT
14° 25′ 36″ = 14.42666666°	14 [°′″] 25 [°′″] 36 [°′″]	14.42666666
12.3456° = 12° 20′ 44.16″	1 2 • 3456 [INV] [°′″]	12ᵒ20ᵒ44.16
sin 63° 52′ 41″ = 0.897859011	[MODE] [4] 63 [°′″] 52 [°′″] 41 [°′″] [sin] "DEG"	0.897859011
$\cos\left(\frac{\pi}{3}\text{rad}\right) = 0.5$	[MODE] [5] "RAD" [π] [÷] 3 [=] [cos]	0.5
tan (−35gra) = −0.61280078	[MODE] [6] "GRA" 35 [+/−] [tan]	−0.61280078
2·sin 45° x cos 65° = 0.597672477	"DEG" 2 [×] 45 [sin] [×] 65 [cos] [=]	0.597672477
$\sin^{-1}$ 0.5 = 30° (Solve equation sin$x$ = 0.5)	"DEG" [•] [5] [INV] [sin]	30.
$\cos^{-1}\frac{\sqrt{2}}{2} = 0.785398163$rad	"RAD" 2 [INV] [√] [÷] 2 [=] [INV] [cos]	0.785398163
$= \frac{\pi}{4}$rad	[÷] [π] [=]	0.25
$\tan^{-1}0.741 = 36.53844576°$	"DEG" [•] 741 [INV] [tan]	36.53844576
= 36° 32′ 18.4″	[INV] [°′″]	36ᵒ32ᵒ18.4
$2.5 \times (\sin^{-1}0.8 - \cos^{-1}0.9) = 68° 13′ 13.53″$	"DEG" 2 • 5 [×] [(] [•] 8 [INV] [sin] [−] [•] 9 [INV] [cos] [)] [=] [INV] [°′″]	68ᵒ13ᵒ13.53

■ Logarithmic functions (log, ln) and exponential functions ($10^x, e^x, x^y, x^{1/y}$)

EXAMPLE	OPERATION	READ-OUT
log 1.23 (= $\log_{10}1.23$) = 0.089905111	1 • 23 [log]	0.089905111
ln 90 (= $\log_e 90$) = 4.49980967	90 [ln]	4.49980967
log 456 ÷ ln 456 = 0.434294481 ($\log x / \ln x$ = fixed constant)	456 [Min] [1] [log] [÷] [MR] [1] [ln] [=]	0.434294481
$10^{1.23}$ = 16.98243652 (To obtain the anti-logarithm of log 1.23)	1 • 23 [INV] [10^x]	16.98243652
$e^{4.5}$ = 90.0171313 (To obtain the anti-logarithm of ln 4.5)	4 • 5 [INV] [e^x]	90.0171313
$10^4 \cdot e^{-4} + 1.2 \cdot 10^{2.3}$ = 422.5878666	1 [EXP] 4 [×] 4 [+/-] [INV] [e^x] [+] 1 • 2 [×] 2 • 3 [INV] [10^x] [=]	422.5878666
$5.6^{2.3}$ = 52.58143837	5 • 6 [INV] [x^y] 2 • 3 [=]	52.58143837
$123^{\frac{1}{7}}$ (= $\sqrt[7]{123}$) = 1.988647795	123 [INV] [$x^{1/y}$] 7 [=]	1.988647795

* x^y and $x^{1/y}$ can be registered as a constant.

EXAMPLE	OPERATION	READ-OUT
$4^{2.5}$ = 32	2 • 5 [INV] [x^y] [INV] [x^y] 4 [=]	32.
$0.16^{2.5}$ = 0.01024	[•] 16 [=]	0.01024.
$9^{2.5}$ = 243	9 [=]	243.
$(78-23)^{-12}$ = $1.305111829 \times 10^{-21}$	[(] 78 [−] 23 [)] [INV] [x^y] 12 [+/-] [=]	1.305111829^{-21}

* x^y and $x^{1/y}$ will be calculated prior to multiplication and division.

EXAMPLE	OPERATION	READ-OUT
$2 + 3 \times 64^{\frac{1}{3}} - 4 = 10$	2 [+] 3 [×] 64 [INV] [$x^{1/y}$] 3 [−] 4 [=]	10.
$2^2 + 3^3 + 4^4 = 287$	2 [INV] [x^y] 2 [+] 3 [INV] [x^y] 3 [+] 4 [INV] [x^y] 4 [=]	287.
$10^{5.1} + 9^{5.1} + e^{5.1}$ = 199615.7293	5 • 1 [INV] [x^y] [INV] [x^y] 10 [=] [Min] [1] 9 [=] [M+] [1] 1 [INV] [e^x] [=] [M+] [1] [MR] [1]	199615.7293

(An equivalent operation is 5 [•] 1 [INV] [10^x] [+] 9 [INV] [x^y] 5 [•] 1 [+] 5 [•] 1 [INV] [e^x] [=]

EXAMPLE	OPERATION	READ-OUT
$2 \times 3.4^{(5+6.7)} = 3306232$	2 [×] 3 • 4 [INV] [x^y] [(] 5 [+] 6 • 7 [)] [=]	3306232.
log sin 40° + log cos 35° = −0.27856798	[MODE] [4] "DEG" 40 [sin] [log] [+] 35 [cos] [log] [=]	−0.27856798
The anti logarithm ··· 0.526540784	(Continue) [INV] [10^x]	0.526540784

■ Hyperbolic functions (sinh, cosh, tanh) and inverse hyperbolic functions ($\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$)

EXAMPLE	OPERATION	READ-OUT
sinh 3.6 = 18.28545536	3 • 6 [hyp] [sin]	18.28545536
tanh 2.5 = 0.986614298	2 • 5 [hyp] [tan]	0.986614298
cosh 1.5 + sinh 1.5 = 0.22313016 $= e^{-1.5}$	1 • 5 [Min] [+/-] [hyp] [cos] [−] [MR] [+/-] [hyp] [sin] [=]	2.352409615 0.22313016
Proof of $\cosh x \pm \sinh x = e^{\pm x}$	(Continue) [ln]	−1.5
$\sinh^{-1} 30$ = 4.094622224	30 [INV] [hyp] [$\sin^{-1}$]	4.094622224
$\cosh^{-1}\left(\frac{20}{15}\right)$ = 0.795365461	20 [÷] 15 [)] [INV] [hyp] [$\cos^{-1}$]	0.795365461
Solve tanh 4x = 0.88 $x = \frac{\tanh^{-1} 0.88}{4}$ = 0.343941914	[•] 88 [INV] [hyp] [$\tan^{-1}$] [÷] 4 [=]	0.343941914
$\sinh^{-1} 2 \times \cosh^{-1} 1.5$ = 1.389388923	2 [INV] [hyp] [$\sin^{-1}$] [×] 1 • 5 [INV] [hyp] [$\cos^{-1}$] [=]	1.389388923

■ Miscellaneous functions ($\sqrt{\ }$, x^2, $1/x$, $x!$, RAN#, RND, ABS, INT, FRAC)

EXAMPLE	OPERATION	READ-OUT
$\sqrt{2} + \sqrt{5}$ = 3.650281539	2 [INV] [$\sqrt{\ }$] [+] 5 [INV] [$\sqrt{\ }$] [=]	3.650281539
$2^2 + 3^2 + 4^2 + 5^2 = 54$	2 [INV] [x^2] [+] 3 [INV] [x^2] [+] 4 [INV] [x^2] [+] 5 [INV] [x^2] [=]	54.
$\frac{1}{\frac{1}{3} - \frac{1}{4}} = 12$	3 [INV] [$1/x$] [−] 4 [INV] [$1/x$] [=] [INV] [$1/x$]	12.
8! (= 1×2×3×···×7×8) = 40320	8 [INV] [$x!$]	40320.
Generate a random number between 0.000 and 0.999.	[INV] [RAN#]	(Example) 0.570
Round the result of 12.3 x 4.56 to three significant digits. 12.3×4.56 ≒ 5.61×10[1]	12 • 3 [×] 4 • 56 [=] [INV] [RND] [3]	5.61 01

* The result will be in the floating point representation.

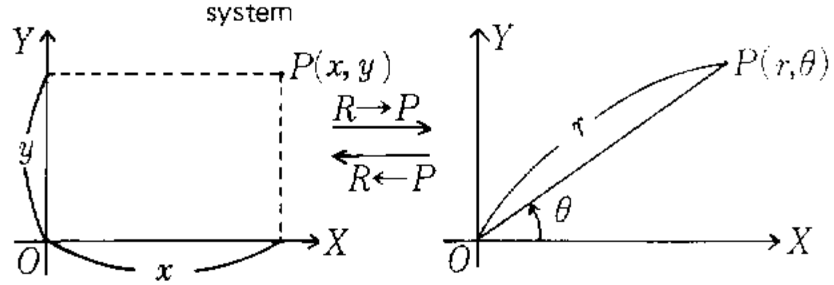
EXAMPLE	OPERATION	READ-OUT
$\log \left\vert \frac{3}{4} \right\vert$ = 0.124938736	3 [÷] 4 [=] [log] [INV] [ABS]	0.124938736
Obtain the integer part of 7800/96. ······81	7800 [÷] 96 [=] [INV] [INT]	81.
Obtain the fraction part of 7800/96. ···0.25	7800 [÷] 96 [=] [INV] [FRAC]	0.25

EXAMPLE	OPERATION	READ-OUT
$\sqrt{13^2-5^2}+\sqrt{3^2+4^2}=17$	[(] 13 [INV] [x^2] [−] 5 [INV] [x^2] [)] [INV] [√] [+]	
	[(] 3 [INV] [x^2] [+] 4 [INV] [x^2] [)] [INV] [√] [=]	17.
$\sqrt{1-\sin^2 40^\circ}=0.766044443$	[MODE] [4] "DEG" 1 [−] 40 [sin] [INV] [x^2] [=] [INV] [√]	0.766044443
$=\cos 40^\circ$	(Continue) [INV] [$\cos^{-1}$]	40.
(Proof of $\cos\theta=\sqrt{1-\sin^2\theta}$)		
$1/2!+1/4!+1/6!+1/8!$	2 [INV] [$x!$] [INV] [$1/x$] [+] 4 [INV] [$x!$] [INV] [$1/x$] [+]	
$=0.543080357$	6 [INV] [$x!$] [INV] [$1/x$] [+] 8 [INV] [$x!$] [INV] [$1/x$] [=]	0.543080357
${}_{10}P_4=\frac{10!}{(10-4)!}=5040$	10 [INV] [$x!$] [÷] [(] 10 [−] [4] [)] [INV] [$x!$] [=]	5040.
${}_{12}C_5=\frac{12!}{5!(12-5)!}=792$	12 [INV] [$x!$] [÷] [(] 5 [INV] [$x!$] [×]	
	[(] 12 [−] 5 [)] [INV] [$x!$] [)] [=]	792.

■ Coordinate transformation (rectangular to polar and vice versa)

Rectangular coordinate system

Polar coordinate system



Argument θ of polar coordinates meets the following inequalities whatever unit of angle is used.
$-180^\circ < \theta \leqq 180^\circ$

EXAMPLE	OPERATION	READ-OUT	
Transform (14, 20.7) to polar coordinates with the argument in degrees.	[MODE] [4] "DEG" 14 [INV] [R→P] 20·7 [=]	24.98979791	(r)
	(Continue) [INV] [X↔Y] [INV] [←D·M·S]	55°55°42.2 (In degrees, minutes and seconds)	
Transform (7.5, −10) to polar coordinates with the argument in radians.	[MODE] [5] "RAD" 7·5 [INV] [R→P] 10 [+/−] [=]	12.5	(r)
	(Continue) [INV] [X↔Y]	−0.92729521	(rad)
Transform (25, 56°) to rectangular coordinates.	"DEG" 25 [INV] [R→P] 56 [=]	13.97982258	(x)
	(Continue) [INV] [X↔Y]	20.72593931	(y)
Transform (4.5, 2/3π radians) to rectangular coordinates.	"RAD" 4·5 [INV] [P→R] [(] 2 [÷] 3 [×] [π] [)] [=]	−2.25	(x)
	(Continue) [INV] [X↔Y]	3.897114317	(y)

■ Percentage

EXAMPLE	OPERATION	READ-OUT
17% of 1500 $1500\times\frac{17}{100}=255$	1500 [×] 17 [INV] [%]	255.
15% add-on of 620 $620+620\times\frac{15}{100}=713$	620 [×] 15 [INV] [%] [+]	713.
4% discount of 7.53 $7.53-7.53\times\frac{4}{100}=7.2288$	7·53 [×] 4 [INV] [%] [−]	7.2288
Percentage of 7.8 against 9.6 $\frac{7.8}{9.6}\times100=81.25(\%)$	7·8 [÷] 9·6 [INV] [%]	81.25
A specimen of 500 grams is added with 300 grams. How many percent is the total weight to the initial one? $\frac{300+500}{500}\times100=160(\%)$	300 [+] 500 [INV] [%]	160.
A temperature rose from 40° C to 46° C. How many percent did it rise? $\frac{46-40}{40}\times100=15(\%)$	46 [−] 40 [INV] [%]	15.
A solution of 500 cc spilled to 400 cc. How many percent did it spill? $\frac{400-500}{500}\times100=-20(\%)$	400 [−] 500 [INV] [%]	−20.

- Percentage calculation involving constants

EXAMPLE	OPERATION	READ-OUT
1) What is 15% of 1500? 225	1500 [×] [×] 15 [INV] [%]	225.
What is 23% of 1500? 345	(Continue) 23 [INV] [%]	345.
What is 25% of 1500? 375	(Continue) 25 [INV] [%]	375.
2) What is 26% of 2200? 572	26 [×] [×] 2200 [INV] [%]	572.
What is 26% of 3300? 858	3300 [INV] [%]	858.
What is 26% of 3500? 910	3500 [INV] [%]	910.
3) How many percent is 30 to 192? . . 15.625%	192 [÷] [÷] 30 [INV] [%]	15.625
How many percent is 12 to 192? 6.25%	12 [INV] [%]	6.25
How many percent is 156 to 192? 81.25%	156 [INV] [%]	81.25
4) 600 grams was added to 1200 grams. What percent is the total to the initial weight? 150%	1200 [+] [+] 600 [INV] [%]	150.

EXAMPLE	OPERATION	READ-OUT
510 grams was added to 1200 grams. What percent is the total to the initial weight? 142.5%	510[INV][%]	142.5
840 grams was added to 1200 grams. What percent is the total to the initial weight? 170%	840[INV][%]	170.
5) How many percent down is 138 to 150? 8%	150[−][−]138[INV][%]	−8.
How many percent down is 129 to 150? . . . 14%	129[INV][%]	−14.
How many percent up is 168 to 150? . . . 12%	168[INV][%]	12.
How many percent up is 156 to 150? . . . 4%	156[INV][%]	4.

1-5. Calculation of standard deviation

- Before starting calculation of standard deviation, clear registers M7, M8, and M9 by [INV][SAC].
- During calculation of standard deviation, other operations (including programmed ones) can be performed unless registers M7, M8, and M9 are used.

■ Data input operation and calculation formulas

- To input a data, depress [x_D] following its value. To input the same value, depress [x_D] as many times as the number of the data.
- When the data involves frequency, input the value, depress [×], input the frequency, and depress [x_D].
  The value of frequency may not be an integer.
- Formulas of standard deviation

$$\sigma_n = \sqrt{\frac{\sum_{i=1}^{n}(xi - \bar{x})^2}{n}} = \sqrt{\frac{\Sigma x^2 - (\Sigma x)^2/n}{n}}$$

To obtain the standard deviation of a finite population using all data of the population

$$\sigma_{n-1} = \sqrt{\frac{\sum_{i=1}^{n}(xi - \bar{x})^2}{n-1}} = \sqrt{\frac{\Sigma x^2 - (\Sigma x)^2/n}{n-1}}$$

To estimate the standard deviation of a population using sampled data of the population

- Mean

$$\bar{x} = \frac{\sum_{i=1}^{n} xi}{n} = \frac{\Sigma x}{n}$$

EXAMPLE	OPERATION	READ-OUT
Data: 55, 54, 51, 55, 53, 53, 54, 52	[INV][SAC]55[x_D]54[x_D]51[x_D]55[x_D]	
	53[x_D][x_D]54[x_D]52[x_D]	52.
	(Standard deviation σ_n) [INV][σ_n]	1.316956719
	(Standard deviation σ_{n-1}) [INV][σ_{n-1}]	1.407885953
	(Mean $\bar{x}$) [INV][$\bar{x}$]	53.375
* The order of depressing the keys to obtain the results is not fixed.	(Number of data n) [MR][n]	8.
	(Sum Σx) [MR][Σx]	427.
	(Square sum Σx^2) [MR][Σx^2]	22805.
What is the unbiased variance, and the difference between each data and mean in the above example?	(Continue) [INV][σ_{n-1}] [INV][x^2]	1.982142857 (Unbiased variance)
	[INV][$\bar{x}$][−][−]55[=]	1.625 $(55-\bar{x})$
	54[=]	0.625 $(54-\bar{x})$
	51[=]	−2.375 $(51-\bar{x})$

What are $\bar{x}$ and σ_{n-1} of the data given below?

No.	Value	Frequency
1	110	10
2	130	31
3	150	24
4	170	2
5	190	3

OPERATION	READ-OUT
[INV][SAC]110[×]10[x_D]	110.
130[×]31[x_D]	130.
150[×]24[x_D]	150.
170[x_D][x_D]	170.
190[x_D][x_D][x_D]	190.
[MR][n]	70.
[INV][$\bar{x}$]	137.7142857
[INV][σ_{n-1}]	18.42898069

* **To delete or correct the wrong data (I)**
Correct operation: 51 [x_D]
(1) Wrong operation: 50 [x_D]
Depress [INV][DEL] and input the correct data.
(2) 49 [x_D] input in some steps before
Depress in the order of 49, [INV], and [DEL] and input the correct data.
(3) Wrong operation: 51[×]
Depress 1 and [x_D] subsequently or [AC] and input the correct data.

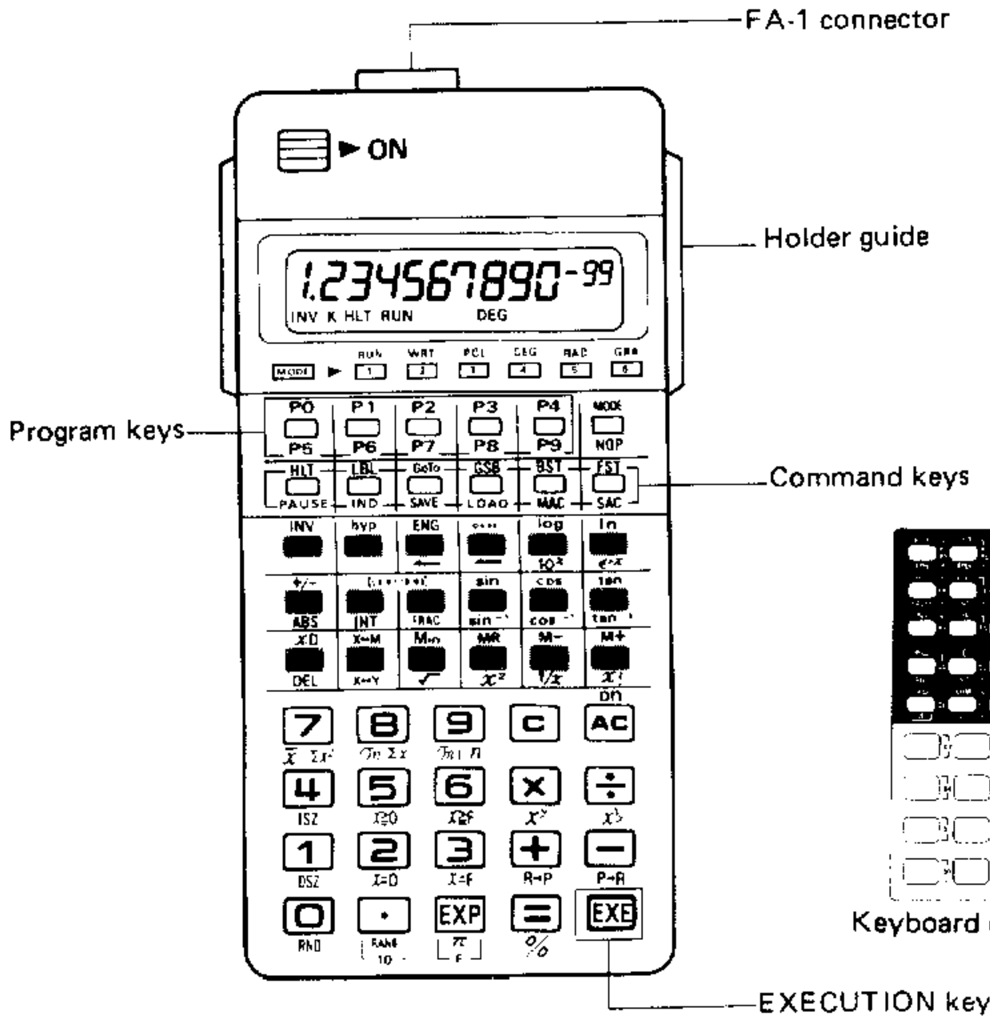
* **To delete or correct the wrong data (II)**
Correct operation: 130 [×] 31 [x_D]
(1) Wrong operation: 120[×]
Depress [AC] and input the correct data.
(2) Wrong operation: 120 [×] 31
Depress [AC] and input the correct data.
(3) Wrong operation: 120 [×] 30[x_D]
Following 120 [×] 30[INV][DEL], input the correct data.
(4) Wrong operation in some steps before: 120 [×] 30 [x_D]
Following 120 [×] 30 [INV] [DEL], input the correct data.

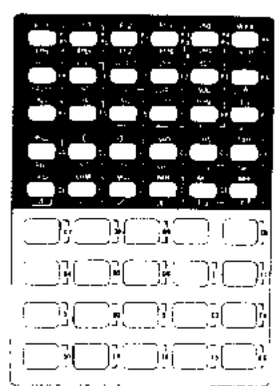
PART-2 Programmed calculation

Part 2 describes programmed operation of the calculator, placing stress to programming. The efficiency of programmed calculation depends on programs. All programs are based on certain algorithms. It is advised that you get a good understanding of programming fundamentals.

2-1. Controls and components required for programmed calculation

* Description of keys and explanation given in Part 1 are not repeated here.





Keyboard overlay sheet

P0 … P4 / P5 … P9 **: PROGRAM NUMBER keys**

Depress one, alone or following [INV], to select one of programs P0 to P9.
You cannot start programmed operation without depressing a program number key.

MODE/NOP **: MODE/NO OPERATION key**

[MODE][1]: RUN (RUN mode)
This is the mode of programmed and step-by-step operations.

[MODE][2]: WRT (WRITE mode)
Put the calculator in this mode to "write" programs, "check" them, and make "addition", "deletion" and "correction".

[MODE][3]: PCL (PROGRAM CLEAR mode)
This is the mode to delete all or specific programs stored in the calculator.

[MODE][4]: DEG
[MODE][5]: RAD
[MODE][6]: GRA
A specific unit of angle will be selected.

[INV][NOP]:
This is a dummy command which simply consumes a certain span of time on execution. The calculator will do no meaningful operation.

HLT/PAUSE **: HALT/PAUSE key**

[HLT]: Halt
During WRT mode
Use this command to stop a running program for inputting data or displaying results. The display reads "HLT"
During RUN mode
The program which is currently running stops after the current instruction has been executed. This is convenient to intercept execution of a program which takes a long time.

[INV][PAUSE]: Temporary Halt
During WRT mode
Use this command to bring a running program to a temporary halt. The program will restart automatically in about a second.

LBL/IND **: LABEL/INDIRECT key**

[LBL]: Destination of jump
During WRT mode
Use this command to write the destination of an unconditional jump.

[INV][IND]: Indirect addressing
During WRT mode
This indicates indirect addressing of a memory location or the destination of an unconditional jump.
During RUN mode
This indicates indirect addressing of a memory location or the destination of a "manual (unprogrammed) jump"

GOTO/SAVE **: GOTO/SAVE key**

[GOTO]: Unconditional jump
During WRT mode
Write an unconditional jump command with this key.

During RUN mode
Depress this key to let control jump occasionally when execution of a program is at a halt (HLT displayed).

INV SAVE: Data transfer from calculator to cassette tape (only when the FA-1 is connected)
During WRT mode
Use to load a command so that the contents of an M-register will be transferred to a cassette tape through the FA-1.
During RUN mode
Use this command to transfer the contents of an M-register.
During PCL mode
Programs stored in the calculator will be moved to a cassette tape.

GSB/LOAD : SUBROUTINE CALL/LOAD key

GSB: Subroutine call
During WRT mode
Use to "write" a command to call a subroutine.
During RUN mode
Use during program debugging (see page 31).

INV LOAD: Data transfer from cassette tape to calculator (only when the FA-1 is connected)
During WRT mode
Use to "write" a command so that data will be loaded from a cassette tape to an M-register through the FA-1.
During RUN mode
Data will be loaded from a cassette tape into an M-register.
During PCL mode
Programs stored in a cassette tape will be loaded into the calculator.

BST/MAC : STEP BACK/MEMORY ALL CLEAR key

BST: Step back
During WRT mode
Use this command to trace the program which is being written or checked backward step by step. Keep the key depressed for more than about a second to back fast.
During RUN mode
During debugging, the command which has just been executed will be displayed by its code as long as this key is depressed.

INV MAC: Memory all clear
During WRT mode
Use to "write" a command so that all M-registers will be cleared.
During RUN mode
All M-registers will be cleared.
During PCL mode
All programs stored will be cleared.

FST/SAC : STEP FORWARD/STANDARD DEVIATION ALL CLEAR key

FST: Step forward
During WRT mode
Use this command to execute a program step by step during checking. Keep the key depressed for more than about a second to advance fast.
During RUN mode
Use to execute a program step by step during debugging.

INV SAC: Clear M7, M8 and M9 registers
During WRT mode
Use to load a command so that the M7, M8 and M9 registers used for calculation of standard deviation will be cleared.

During RUN mode
The M7, M8 and M9 registers will be cleared.

EXE : EXECUTE key

During WRT mode
Use to "write" a command so as to initiate data transfer between the calculator and cassette tape via the FA-1.
During RUN mode
During programmed operation, depress this key to restart execution of a program which is at a halt (with "HLT" displayed).

C: CLEAR key

During WRT mode
Depress this key to delete the currently displayed command of a program stored in the calculator.
During RUN mode
The displayed data will be erased for correction.

AC: ALL CLEAR key

During WRT mode
Use to "write" a command so that all registers but the M-registers will be cleared.
During RUN mode
All registers but the M-registers will be cleared. The mode of programmed operation will be released if this key is depressed in the middle of programmed operation (with "–" displayed).
During PCL mode
A designated program alone will be cleared.

1/DSZ, 4/ISZ, 2/x=0, 5/x≥0, 3/x=F, 6/x≥F : Numeric/condition test keys

These keys will be used following INV in the WRT mode.

INV DSZ: DECREMENT AND SKIP ON ZERO
The contents of the M0 register is decremented and, if the result is zero, the next command is skipped; otherwise the next command is executed.

INV ISZ: INCREMENT AND SKIP ON ZERO
This is the same as INV DSZ but that the contents of the M0 register is incremented.

INV x=0
The next command is executed if the contents of the X-register (display register) is zero; otherwise the next command is skipped.

INV x≥0
The next command is executed if the contents of the X-register is positive or zero; otherwise the next command is skipped.

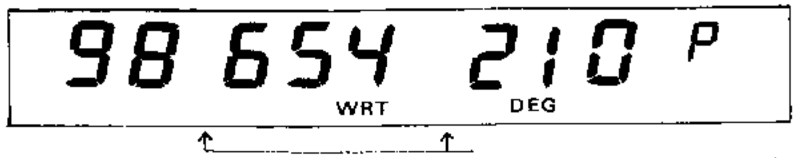
INV x=F
The next command is executed if the contents of the X- and MF registers are identical; otherwise the next command is skipped.

INV x≥F
The next command is executed if the contents of the X-register are greater than or equal to the contents of the MF register; otherwise the next command is skipped.

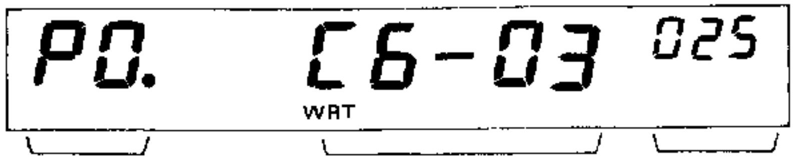
■ Program size and command code

The calculator is capable of storing 10 programs (P0 to P9) as long as the total number of steps is not greater than 128. The program number, command code, and the number of steps are displayed while a program is being written, checked or debugged.

During MODE 2



Program number stored.



Program number | Command code (Min 3) | Number of steps

- The command code displayed consists of a symbol (*P, F, C., or E*) and numbers (0, 1 to 9) as indicated on "the keyboard overlay sheet".
- A step of execution performs a function. There are three kinds of commands: depression of a single key makes a step of command (1-key command), and successive depression of two and three keys makes a step of command (2- and 3-key commands).
  One-key commands: number, +/−, +, −, ×, ÷, =, [(,)], sin, log, HLT, P0, etc.
  Two-key commands: X↔M1, M+0, GoTo3, LBL5, hyp sin, $\sin^{-1}$, ABS, $\sqrt{\ }$, $x{=}0$, x^y, P7, etc.
  Three-key commands: INV hyp $\sin^{-1}$, RND4, GSB P6, etc.
  Program number designation uses a step but is not counted in the number of steps displayed.

2-2. Basic programming

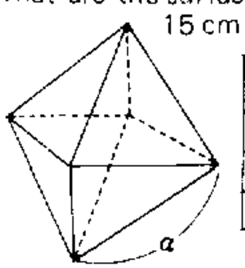
- The following procedure will be followed to carry out programmed calculation.

> **(1) Analysis**
> We analyze the problem and find an algorithm for solution.
> **(2) Programming**
> We convert the algorithm into a program.
> **(3) Loading the program**
> We load the program into the calculator.
> **(4) Execution**
> We execute the program.

Let us see the folowing example.

Example:

What are the surface area and volume of regular octahedrons which have ridges of 10, 7, and 15 cm respectively.



Ridge (*a*)	Surface area (*S*)	Volume (*V*)
10 cm	( ) cm^2	( ) cm^3
7	( )	( )
15	( )	( )

(1) Analysis

Let *S* represent the surface area, *V* the volume, and *a* the length of a ridge of a regular octahedron. Then *S* and *V* are expressed as follows.

$$S = 2\sqrt{3}\,a^2, \qquad V = \frac{\sqrt{2}}{3}a^3$$

(2) Programming

To carry out the above calculation manually, operate the calculator in the following way.

2 [×] 3 [INV] [√] [×] Value of *a* [INV] [x²] [=] → *S*
2 [INV] [√] [÷] 3 [×] Value of *a* [INV] [x^y] 3 [=] → *V*

You may put the value of *a* in an M-register before use, and this is more convenient. In this case, operate in the following way.

Value of *a* [Min] [1] 2 [×] 3 [INV] [√] [×] [MR] [1] [INV] [x²] [=] → *S*
2 [INV] [√] [÷] 3 [×] [MR] [1] [INV] [x^y] 3 → *V*

The above sequence of operation can be programmed just as it is.
Once execution of the program begins, it does not stop unless the HLT command is executed. To input data and display results, execution must be stopped temporarily using the command.
The following sequence of operation stores the above program in P0.

P0 HLT, Min 1, 2, ×, 3, INV $\sqrt{\ }$, ×, MR 1, INV x^2, =, HLT,
2, INV $\sqrt{\ }$, ÷, 3, ×, MR 1, INV x^y, 3, =, HLT,

(In this manual, each step is separated by ",".)

(3) Loading the program

> To load the program to the calculator:
> (1) Put the calculator in the WRT mode (MODE 2).
> If the program number which is to be used is already in use, it must be changed or deleted. (See "How to delete programs" (page 33) and "How to change program number" (page 33).)
> (2) Depress keys in the sequence of the program. (If you have depressed a wrong key, depress [C] and then the right one.)

OPERATION	READ-OUT	
MODE 2	9 6 WRT 43210 P	(P5, P7, and P8 are already in use.)
[P0]	P0. 115	(Number of steps available for use)
[HLT]	P0. FP 001	
[Min]	P0. E6 002	(2-key command)
[1]	P0. C6-01 002	(2-key command)
[2]	P0. 02 003	
[×]	P0. E1 004	

OPERATION	READ-OUT	
[3]	P0. 03 005	
[INV]	P0. FF 006	(2-key command)
[√]	P0. FF-C6 006	
[×]	P0. E1 007	
[MR] [1]	P0. C7-01 008	(2-key command)
[INV] [x²]	P0. FF-C7 009	(2-key command)
[=]	P0. E5 010	
[HLT]	P0. FP 011	
[2]	P0. 02 012	
[INV] [√]	P0. FF-06 013	
[÷]	P0. E2 014	
[3]	P0. 03 015	
[×]	P0. E1 016	
[MR] [1]	P0. C7-01 017	(2-key command)
[INV] [x²]	P0. FF-E1 018	(2-key command)
[3]	P0. 03 019	
[=]	P0. E5 020	
[HLT]	P0. FP 021	

The program has been loaded into the calculator.

* As you depress keys during the WRT mode, corresponding commands are "written in" (put into the memory of) the calculator as part of a program. The command which has been written is displayed together with the step number. The program need not end with HLT.

(4) Execution

To execute the program loaded into the calculator:
(1) Put the calculator in the RUN mode ([MODE] [1]).
(2) Depress the relevant program number key ([P0] to [P4] , [INV] [P5] to [INV] [P9]).
(3) When the display reads "HLT", input data (or read the result) and depress [EXE] .
(4) To execute the program again with varied data, depress the program number key.
(5) To stop programmed operation, operate [MODE] [1] ("HLT" displayed is cleared).

In the above example, surface area S and volume V of a regular octahedron having ridges whose length is a are unknown.

Ridge (a)	Surface area (S)	Volume (V)
10 cm	(346.4101615) cm²	(471.4045207) cm³
7 cm	(169.7409791)	(161.6917506)
15 cm	(779.4228634)	(1590.990257)

OPERATION		READ-OUT	
	[MODE] [·]	RUN 0.	(Unnecessary if "RUN" is already displayed)
(Program number assignment)	[P0]	HLT RUN 0.	
	10 [EXE]	346.4101615	(S for a = 10)
(Again)	[EXE]	471.4045207	(V for a = 10)
(To execute again)	[P0]	471.4045207	
	(a) 7 [EXE]	169.7409791	(S for a = 7)
	[EXE]	161.6917506	(V for a = 7)
	[P0]	161.6917506	
	(a) 15 [EXE]	779.4228634	(S for a = 15)
	[EXE]	1590.990257	(V for a = 15)
(To terminate programmed operation)	[MODE] [1]	0.	

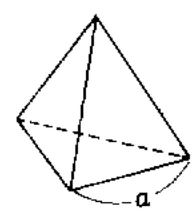
"HLT" is cleared.

* In programmed operation, execution proceeds and a result is displayed each time after [EXE] is depressed following input data or after a result is read.

2-3. Checking and correcting program

Example:

Let us calculate the surface area and volume of regular tetrahedrons having ridges of 10, 7.5, and 20 cm respectively.



Ridge (a)	Surface area (S)	Vlume (V)
10 cm	(173.2050807) cm²	(117.8511302) cm³
7.5	(97.42785792)	(49.71844555)
20	(692.820323)	(942.8090416)

(1) Analysis

Let S represent the surface area, V the volume, and a the length of the ridge of a regular tetrahedron. Then S and V are expressed as follows.

$$S = \sqrt{3}\,a^2, \qquad V = \frac{\sqrt{2}}{12}a^3$$

(2) Programming

We put values of a in memory and select P1 as the program number. Then the program is as follows.

P1 HLT, Min 1, 3, INV $\sqrt{\ }$, X, MR 1, INV x^2, =, HLT, 2, INV $\sqrt{\ }$, ÷, 1, 2, X, MR 1, INV x^y, 3, =, HLT,   20 steps

We can load this program directly into the calculator. Here we modify the "regular octahedron" program given in page 26 to learn how to check program and make correction or modification.

(3) Checking and modifying program

Checking program means confirming the contents of a program, written in program memory, on the display.
For this purpose, put the calculator in the WRT mode ([MODE][2]), assign the program number, and operate [FST] or [BST]. Then the number of steps and the contents of the program will be displayed. To make addition and deletion, follow basically the same procedure.

1. Change the program number from P0 to P1.
2. Delete "2" and "X".
3. Replace "3" in "÷, 3, x" with "1,2".

This is the modification we make to obtain the desired program.

OPERATION		READ-OUT	
	[MODE][2]	9 6 WRT 4321 P	
	[P0]	P0. WRT 93	(Program P0 has been called. The number of steps still available for use is displayed)
Changing program number	[C]	P . 93	
	[P1]	P1. 93	
(Program check)	[FST]	P1. FP 001	(1st step [HLT])
(Program check continues.)	[FST]	P1. C6-01 002	(2nd steps [Min][1])
	[FST]	P1. 02 003	(3rd step: number 2)
(Deletion)	[C]	P1. 003	
	[FST]	P1. E1 003	This command is deleted. (3rd step: [×])
(Deletion)	[C]	P1. 003	
	[FST]	P1. 03 003	(3rd step: numeral 3)
	[FST]	P1. FF-C6 004	(4th step: [INV][√])
	[FST]	P1. E1 005	(5th step [×])

Depress [FST] successively until the step of "2, INV $\sqrt{\ }$, ÷," is reached. (You may keep [FST] depressed. If you go beyond the desired step, depress [BST] .)

	[FST]	P1. E2 012	(12th step: [÷])
	[FST]	P1. 03 013	(13th step: numeral 3)
(Deletion)	[C]	P1. 013	
(Addition)	[1]	P1. 01 013	(13th step: numeral 1)
(Addition)	[2]	P1. 02 014	(14th step: numeral 2)
	[FST]	P1. E1 015	(15th step: [×])
	[FST]	P1. 07-01 016	(16th step: [MR][1])
	[FST]	P1. FF-E1 017	(17th step: [INV][x^y])
	[FST]	P1. 03 018	(18th step: numeral 3)
	[FST]	P1. E5 019	(19th step: [=])
	[FST]	P1. FP 020	(20th step: [HLT])
(To advance further)	[FST]	P1. 094	(The number of steps available for use is displayed.)
	[FST]	P1. FP 001	(1st step: [HLT])
	[FST]	P1. C6-01 002	(2nd step: [Min][1])
(To back up)	[BST]	P1. FP 001	(1st step: [HLT])
	[BST]	P1. 094	(Program check proceeds from the last step.)
	[BST]	P1. FP 020	(20th step: [HLT])
	[BST]	P1. E5 019	(19th step: [=])
	[MODE][1]	RUN 0.	(No command is written in this step.)

Now modification of the program is complete.

How to check program

(1) Put the calculator in the WRT mode ([MODE][2]). (Program numbers, which are already in use, are not displayed.)
(2) Depress the relevant program number key ([P0] to [P4], [INV][P5] to [INV][P9]).
(3) Depress [FST] or [BST] and check the command code and step number displayed. (If you keep [FST] or [BST] depressed, you will advance or trace back over program steps continuously. When the end or beginning of the program is reached, fast forwarding or reversing will be repeated again.)

How to make addition, deletion and correction

(1) Display a relevant step by the procedure of program check.
(2) To delete the step, depress [C].
(3) To correct it, depress [C] to erase and then input the correct command.
(4) To add a step, display the step after which insertion is to be made and input a new step or steps. (Addition becomes impossible when there is no more program area available.)
 * Addition and deletion may be done in any order. When addition and deletion are made, step numbers are automatically readjusted.
 (You may use [INV][NOP] to leave step numbers unchanged after deletion.)

(4) Execution

OPERATION		READ-OUT	
	[MODE] [1]	0.	(Unnecessary when "RUN" is already displayed)
(Program number assignment)	[P1]	0.	
	10[EXE]	173.2050807	(S for a = 10)
	[EXE]	117.8511302	(V for a = 10)

Repeat these three steps with other values of a.

Debugging

The program you have made and loaded into the calculator may have some bugs (errors). Follow the procedure below to remove such bugs (i.e. to "debug" the program).

How to debug program:

(1) Put the calculator in the RUN mode ([MODE][1]).
(2) Depress [GSB] and the relevant program number key.
(3) As you depress [FST], the program is executed step by step. (If the display reads "HLT", input test data and depress [FST].)
(4) If you keep [BST] depressed, the command which has just been executed and its step number are displayed.
(5) Repeat steps (3) and (4) and check results and commands.
 * You can enter steps (3) and (4) even in the middle of programmed operation (when execution is at a halt) and resume programmed operation by depressing [EXE] during debugging.

Example:

Let us debug the "regular tetrahedron" program which is mentioned before.

OPERATION		READ-OUT	
	[MODE] [1]	RUN 0.	(Unnecessary if the RUN mode is entered)
(Debugging starts)	[GSB] [P1]	P 1. RUN	
	[FST]	P 1. HLT	(1st step: HLT)
(Data) 2		2.	(Input test data)

OPERATION		READ-OUT	
	[FST]	2.	(2nd step Min 1)
	[FST]	3.	(3rd step numeral 3)
	[FST]	1.732050807	
(Check command) (Keep depressed)	[BST]	P 1. EE-06 004	(4th step $\sqrt{\ }$) (Step No. 4: [INV][√])
	[FST]	1.732050807	(5th step: X)
	[FST]	2.	(6th step MR 1)
	[FST]	4.	(7th step x^2)
	[FST]	6.928203230	(8th step:=)
	[FST]	6.928203230 HLT	(9th step HLT) S for a= 2
	[FST]	2.	(10th step numeral 2)
	[FST]	1.414213562	(11th step: $\sqrt{\ }$)
	[FST]	1.414213562	(12th step: ÷)
(Check command) (Keep depressed)	[BST]	P 1. E2 012	(Step No. 12 : [÷])
	[FST]	1.	(13th step: numeral 1)
	[FST]	12.	(14th step: numeral 2)
	[FST]	0.11785113	(15th step : X)
(Check command) (Keep depressed)	[BST]	P 1. E1 015	(Step No. 15 : [×])
	[FST]	2.	(16th step: MR 1)
(Check command) (Keep depressed)	[BST]	P 1. C7-01 016	(Step No. 16: [MR][1])
	[FST]	2.	(17th step: x^y)
(Check command) (Keep depressed)	[BST]	P 1. F6-E1 017	(Step No. 17: [INV][x^y])
(Execution proceeds normally)	[EXE]	0.942809041 HLT	V for a=2

How to delete programs

- To delete all programs:

Put the calculator in the PCL mode ([MODE][3]). Operate [INV][MAC] . (The contents of the Mregisters remain unerased. To clear them, operate [INV] [MAC] in the RUN mode.)

- To delete a specific program:

Put the calculator in the PCL mode. Depress the relevant program number key [P0] to [P4] , (or [P5] to [P9] following [INV]) and [AC] . (If a program number is used, the display indicates the program number by depressing the program number key in the PCL mode. But it is not displayed if the program number is not used (i.e. no program uses the program number).)

How to change program number

Each program has a certain program number ([P0] to [P9]). More than one program cannot use the same program number. You may want to assign a specific program number to a specific program. If the number is used, you can reassign the program to a new number so that the desired number can be used for your program. This is done in the following way.

(1) Put the calculator in the WRT mode ([MODE][2]) and depress the relevant program number key.
(2) Depress [C] .
(3) Depress another program number key to which reassignment is made. (If this program number is used by a program, this operation is ineffective.)
(4) Put the calculator in the RUN mode ([MODE][1]) or the WRT mode ([MODE][2]).

Example 1: To change P0 to P9. [MODE] [2][P0][C][INV][P9][MODE] [1]

Example 2: To change P5 to P6. [MODE] [2][INV][P5][C][INV][P6][MODE] [1]

Remarks for programming and operation

■ Programming rules

- The program consists of commands arranged just in the same order as with step-by-step calculation. (The sequence of commands is the same as of ordinary mathematical expressions.)
- All built-in functions can be used in programs.
- There is no restriction on the length of mathematical expressions.
- Any number of constants can be used in a program (each has a mantissa of up to 10 digits and an exponent of up to 2 digits). In this case, each of digits, decimal point, +/-, and EXP is counted as one step.
- Calculation involving constants can be programmed.

■ To write a program (WRT mode)

- Just after the WRT mode is entered, you can depress only either of [P0] to [P4] , [INV][P5] to [INV][P9] , and [MODE] [1] to [MODE] [3] .
- The number of steps displayed during the WRT mode plus one is the actual number of steps written in the program memory because the step of [Pn] (or [INV] [Pn]) is not counted which actually makes a step.
- When the program area becomes full, no more commands cannot be written (but those which are already written remain there).
- While a program is written, validity check of function arguments and nested parentheses, etc. are not carried out (error may be detected during execution).
- Depress [P0] to [P4] , [INV][P5] to [INV][P9] , or [MODE][1] to [MODE][3] to terminate loading a program. Then you can load another program or enter a different mode.
- The program need not end with HLT. (If HLT is not programmed, the last result remains displayed after the last command is executed.)

■ Execution (RUN mode)

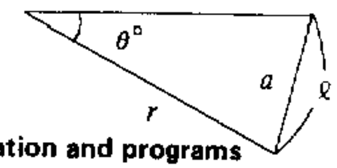
- Nothing will occur when a program number key to which no program is assigned is depressed.
- When execution of a program is at a halt, you can carry out other calculation manually. You may use the result of calculation as an input data or use the intermediate result of programmed operation currently displayed for some other calculation. If the intermediate result is used in subsequent programmed operation, the data must be restored before programmed operation is resumed.

- If you depress [AC] when execution is at a halt, the currently displayed data and command will be cleared. If you depress [AC] during execution ("–" displayed), execution of the program will terminate. [AC] allows you to terminate operation when it takes a long time or it is running in a loop.
- If you depress [HLT] during programmed operation ("–" displayed), execution comes to a halt after the current command is executed. This is the same when [HLT] is depressed in the middle of PAUSE. [HLT] allows you to debug a program in the middle of execution.
- Errors ("E." displayed) possibly detected during programmed operation are summarized below.
  1. Overflow of a result of operation or data put in memory.
  2. Overflow of an argument or value of a function.
  3. The destination of a jump ([GOTO]) is not found.
  4. A sburoutine which is called ([GSB]) is not found.
  5. Illegal nesting of parentheses or overflow in an L-register (In this case "C." is displayed.)

  In all cases (but 5), the display indicates "E." and the pertinent step number and execution comes to a halt.
  To localize the command whose execution resulted in error, release the error status by depressing [AC] (or [C] for the case of 5) and then depress [RST]

2-4. Programming examples

Let us study some variations of program to solve a problem for their merits and demerits.
Problem: What are the length, ℓ, of arc and the length, a, of chord of a sector whose radius is r and central angle is θ?



$\ell = \frac{r \pi \theta}{180}$ $a = 2r\sin\frac{\theta}{2}$

- **Operation and programs**

	OPERATION STEP		PROGRAM	
A	1	[P0]	P0 MODE 4, HLT,	
	2	r [EXE]	Min 1, X, HLT,	
	3	θ [EXE] → ℓ	Min 2, X, π, ÷, 1, 8, 0, =, HLT,	(P0)
	4	[EXE] → a	MR 2, ÷, 2, =, sin, X, 2, X, MR 1, =,	24 steps plus 1 step
B	1	r [P0]	P0 MODE 4, Min 1, X, HLT,	
	2	θ [EXE] → ℓ	Min 2, X, π, ÷, 1, 8, 0, =, HLT,	(P0)
	3	[EXE] → a	MR 2, ÷, 2, =, sin, X, 2, X, MR 1, =,	23 steps plus 1 setp
C	1	r [Min][1]		
	2	θ [Min][2]		
	3	[P0] → ℓ	P0 MODE 4, MR 1, X, π, X, MR 2, ÷, 1, 8, 0, =, HLT,	(P0)
	4	[EXE] → a	MR 2, ÷, 2, =, sin, X, 2, X, MR 1, =,	22 step plus 1 step
D	1	r [P0]	P0 Min 1, ... 1 step	
	2	θ [P1]	P1 Min 2, MODE 4, ... 2 steps	
	3	[P2] → ℓ	P2 MR 1, X, π, X, MR 2, ÷, 1, 8, 0, =, . 10 steps	(P0, P1, P2, P3)
	4	[P3] → a	P3 MR 2, ÷, 2, =, sin, X, 2, X, MR 1, =, 10 steps	plus 4 steps

- **Advantages and disadvantages**

A: Standard program
- The sequence of operation is simple. After the program number is assigned, [EXE] is operated simply or after input data.
- It is possible to store many programs (up to 10 programs).
- If requires a small number of steps.
- Since the sequence of operation is fixed, it is impossible to change part of data or to get part of results.

B. A variation
- This is basically the same as A but the sequence of operation is not so simple.

C: Data input operation is not easy.
- The number of steps is the samllest of all.
- It is possible to store many programs.
- Data must be put in memory before execution. This may induce error.

D: User's function type
- Data may be input in any order. Results can be get in any order. The program number keys are connected with certain functions.
- This is convenient since data can be changed partly.
- This requires more steps than the others.
- This may cause the shortage of program numbers if the number of input data and unknowns are large.
- It is inconvenient to store more than one program.

The most preferable program will depend on a particular case.
Programs of type A are mainly described in this manual and library. Some programs are of the other types.

Jump

There are four kinds of jump commands.
1. Unconditional jump to a designated register: GOTO, LBL
2. Conditional jump which skips the next command depending on the content of the X-register:
   $x = 0$, $x \geqq 0$, $x = F$, $x \geqq F$
3. Conditional jump which skips the next command depending on the content of an M-register: DSZ, ISZ
4. Jump to a subroutine and return from it: GSB.

These commands can be used independently and in combination.

■ Use of unconditional jump

- GOTO N causes control to jump to LBL N of the program unconditionally.
- N is a digit of 0 or 1 to 9.
- GOTO N and LBL N of up to 10 pairs can be used anywhere in a program.
- More than one GOTO N may be used in a program but the destination of jump (LBL N) must be uniquely defined.
- If the destination of a jump is undefined, execution of the program results in error.
- You can input the jump command GOTO N, manually (depress [GOTO][N]([0] or [1] to [9]). If the destination, LBL N, is undefined, nothing will be performed.

■ Use of conditional jump

Conditional jump 1

- The content of the X-register (displayed) is compared with zero or the content of the MF register and, if a specific condition is met, the next step is executed; otherwise the next step is skipped.
- There are four commands of this type.
  - $x = 0$: Test is made if the content of the X-register is zero.
  - $x \geqq 0$: Test is made if the content of the X-register is zero or positive.
  - $x = F$: Test is made if the contents of the X- and MF registers are equal.
  - $x \geqq F$: Test is made if the content of the X-register is equal to or greater than that of the MF register.

- **Example 1:** To find the largest number out of registers M1 through M3.

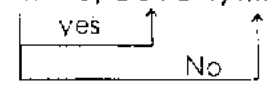
MR 1, Min F, MR 2, INV $x \geqq F$, Min F, MR 3, INV $x \geqq F$, Min F, MR F, ...

- **Example 2:** To find the smallest number out of registers M1 through M3.

MR1, +/−, MinF, MR 2, +/−, INV $x \geqq F$, Min F, MR 3, +/−, INV $x \geqq F$, Min F, MR F, +/−, ...

- **Example 3:** To jump to LBL 1 if the content of the M4 register is zero.

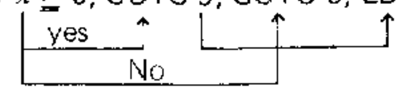
MR 4, INV $x = 0$, GOTO 1, ...



(LBL 1 may be located in this program either in the first place or at the end.)

- **Example 4:** To jump to LBL 5 if the content of the M5 register is negative.

MR 5, INV $x \geqq 0$, GOTO 9, GOTO 5, LBL 9, ...



(LBL 5 may be located in this program either in the first place or at the end.)

Program which uses conditional and unconditional jumps

Example: Let us obtain the roots of quadratic equation in different ways for real and imaginary roots.

No.	Equation	Coefficients			Roots
		a	b	c	
(1)	$8x^2 + 6x + 1 = 0.$	8	6	1	(−0.25, −0.5)
(2)	$2x^2 + 26x + 89 = 0$	2	26	89	(−6.5 ± 1.5 i)
(3)	$2x^2 - 28x + 98 = 0$	2	−28	98	(7)

- Solution

The roots of $ax^2 + bx + c = 0$ are

$$x = \frac{-b \pm \sqrt{D}}{2a}$$

where $D = b^2 - 4ac$

- Programming ([MODE][2])

P0	HLT, Min 1, HLT, Min 2, HLT, Min 3, 2, X, MR 1, =, Min 5, ...	Input a, b, and c and calculate $2a$.
	MR 2, INV x^2, −, 4, X, MR 1, X, MR 3, =, Min 4, INV $x \geqq 0$, GOTO 1, ...	Go to LBL 1 if $D \geqq 0$.
LBL2,	MR 2, +/−, ÷, MR 5, =, INV PAUSE, ... MR 4, +/−, INV $\sqrt{\ }$, ÷, MR 5, =, INV PAUSE, GOTO 2,....	Imaginary roots are displayed one by one.
LBL1,	MR 2, +/−, +, MR 4, INV $\sqrt{\ }$, =, ÷, MR 5, =, HLT, ... MR 2, +/−, −, MR 4, INV $\sqrt{\ }$, =, ÷, MR 5, =, HLT, ...	Real roots are displayed.

59 steps

- Execution ([MODE] [1])

(1) [P0] 8 [EXE]
6 [EXE]
1 [EXE] → −0.25
[EXE] → −0.5

(2) [P0] 2 [EXE]
26 [EXE]
89 [EXE] → −6.5 and 1.5 are displayed alternately in intervals of about a second.

(3) [P0] 2 [EXE]
28 [+/−] [EXE]
98 [EXE] → 7
[EXE] → 7

Conditional jump 2

- ISZ and DSZ are this type of commands. ISZ increments the content of the M0 register and DSZ decrements it before executing a test.
- If the content of the M0 register is not zero, the next step is executed; otherwise it is skipped.
- These commands are used for loop control.
- **Examples:** To run in a loop 7 times.

With ISZ ..... 7, +/−, Min 0, LBL 1, ... INV ISZ, GOTO 1, ...
7 times M0 ≠ 0 — M0 = 0

With DSZ..... 7, Min 0, LBL 1, ... INV DSZ, GOTO 1, ...
7 times M0 ≠ 0 — M0 = 0

■ Use of subroutine

A program may consists of a main program and subroutines. The main program composes the central framework. A subroutine is a self-contained part of a program which can be incorporated into the program as many times as necessary. It can be incorporated anywhere in a program and even in different main programs. In other words, a subroutine performs a certain self-contained part of job which may be requested many times in a main program. Use of subroutines makes programming easier and reduces the number of steps.

Program incorporating subroutines

Example: Let us make the common part of the "regular octahedron" and "regular tetrahedron" programs respectively given in pages 25 and 28 into a subroutine.

Regular octahedron	P0 HLT, Min 1, 2, X, GSB INV P9, 3, GSB INV P8,	7 steps
Regular tetrahedron	P1 HLT, Min 1, GSB INV P9, 1, 2, GSB INV P8,	6 steps
Subroutine	INV P9 3, INV $\sqrt{\ }$, X, MR 1, INV x^2, =, HLT, 2, INV $\sqrt{\ }$, ÷,	10 steps
Subroutine	INV P8 X, MR 1, INV x^y, 3, =, HLT,	6 steps
		Total 29 steps

The total number of steps of the two main programs are 43 steps (21 + 20 + 2 (P0, P1) = 43), and it is reduced to 33 steps (7 + 6 + 10 + 6 + 4 (P0, P1, INV P9, INV P8) = 33) by using the subroutines. Operation remains unchanged as described in pages 28 and 31 .
Note: If GSB Pn is placed in parentheses, do not use "equal" and a closing parenthesis which is not paired with an opening parenthesis in the subroutine.

- GSB Pn forces control to jump to program Pn and, upon completion of execution of program Pn, to return to the step which is next to GSB Pn.
- Pn is P1 to P4 or INV P5 to INV P9.
- GSB Pn may be located anywhere in a program.
- If program Pn which is referred to by GSB Pn is undefined, execution results in error.
- GOTO N and LBL N used in a subroutine are effective only in the subroutine. (The destination of jump GOTO in a main program cannot be located in a subroutine.)
- Subroutines can be nested up to four levels (that is, a subroutine or subroutines can be called in another subroutine as long as the level of nesting does not exceed four). Command GSB will be disregarded if the nesting level exceeds four.

■ Use of indirect addressing

IND is the command of indirect addressing for designating a register or a destination of jump.

Indirect addressing of M-register

- Use IND together with register commands (X↔M, Min, MR, M−, M+) to assign an M-register indirectly.
- INV IND, M+ n addresses a register indicated by the content of the Mn register to execute M+. (The same effect is produced by unprogrammed operation.)
- You may use any register command in place of M+.
  Example: INV IND X↔M 3 executes X↔M 5 when M3 contains 5.
- n is 0, 1 to 9, ·0, ·1 to ·9, F, or ·F.
- If the Mn register contains other than 0 to 19, the most and second most significant digits of the integer part are taken as the register number. If the content exceeds 20, the command is ineffective.
  Example 1: INV IND MR 8 performs no operation when M8 contains−256.
  Example 2: INV IND MR 8 executes MR 2 when M8 contains−2.56.

Indirect addressing for jump

- INV IND GOTO n forces control to jump toa LBL designated by the content of the Mn register.
  Example: INV IND GOTO 3 is equivalent to GOTO 5 (jump to LBL 5) when M3 contains 5.
- n is 0, or 1 to 9.
- If Mn contains a number other than 0 to 9, the most significant digit of the integer part is taken as the LBL number.
  Example: INV IND GOTO 5 causes jump to LBL 0 when M5 contains 0.1.
- If the destination is undefined, the jump command will be disregarded.

Indirect addressing for loop control

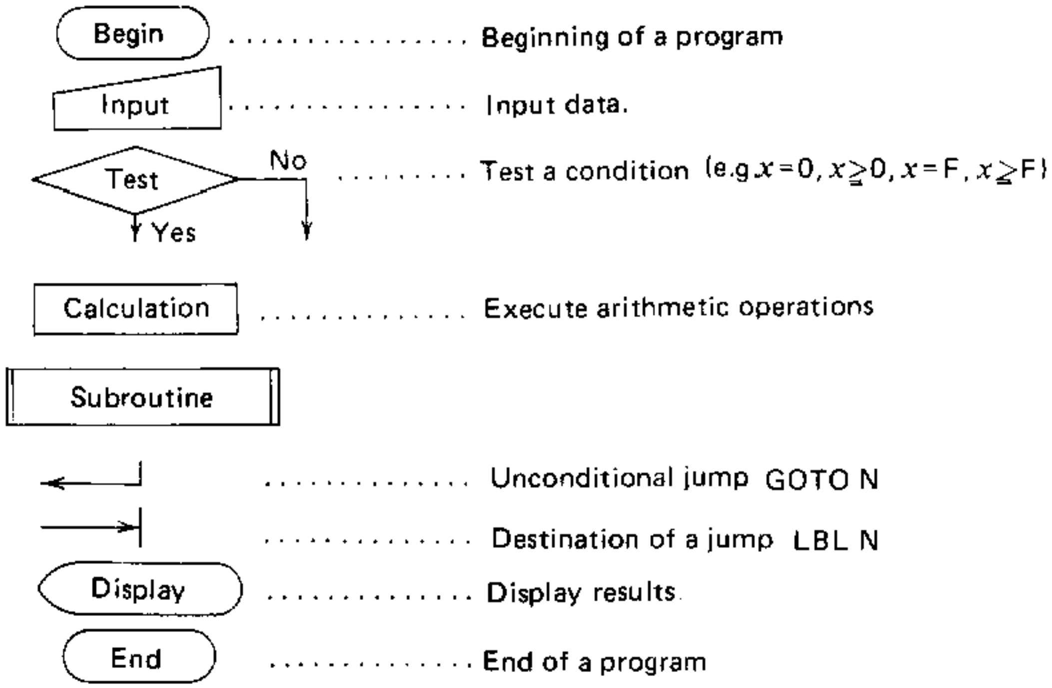
- INV IND INV ISZ or INV IND INV DSZ causes ISZ or DSZ to operate on the register designated by the content of the M0 register.
  Example 1: When M0 contains 5 and M5 100, INV IND INV DSZ decrements the content of M5 into 99.
  Example 2: When M0 contains 3 and M3 −1, INV IND INV ISZ increments the content of M3 to 0 and skips the next step.
- Refer to "Indirect addressing of M-register" for the case M0 contains a number other than 0 to 9.

Indirect addressing for subroutine call

- INV IND GSB 0 calls the subroutine which is identified by the content of the M0 register.
  Example: When M0 contains 7, INV IND GSB 0 is equivalent to GSB P7.
- When M0 contains a number other than 0 to 9, the most significant digit of the integer part is taken as the program number.
- If the program number designated is not used, the subroutine call command will be disregarded.

■ Draw flowchart

- It is suggested that you draw a flowchart which represents a sequence of events occurring when a calculation of interest is executed.
  The flowchart is usually drawn with symbols as follows.



■ Elementary programming

Example 1: To obtain the sum and difference of the largest and smallest numbers out of input data (x ≧ F)

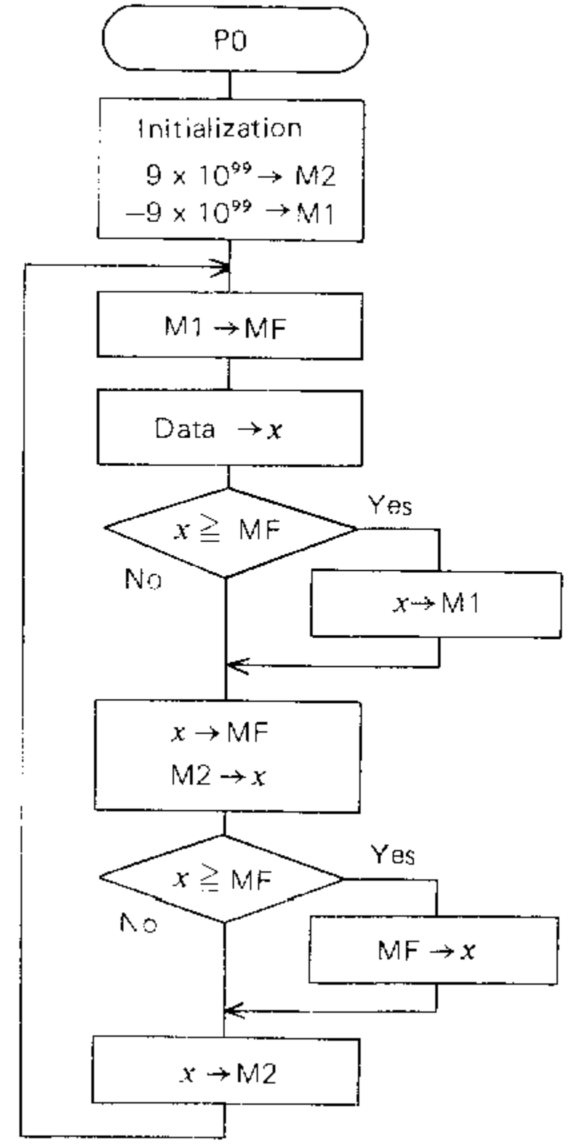
● Program

```
P0   9, EXP, 9, 9, Min 2, +/–, Min 1,
LBL 1, MR 1, Min F, AC, HLT,
     INV x ≧ F, Min 1, Min F,
     MR 2, INV x ≧ F, MR F, Min 2, GOTO 1,      20 steps
P1   MR 1, +, MR 2, =, HLT,
     MR 1, –, MR 2, =,                          9 steps
```

● Operation

	[P0]	
Data	[EXE]	
Data	[EXE]	
Repeat this.		
At the end	[P1]	Sum
	[EXE]	Difference

● Flowchart



* The flowchart of program P1 is omitted.

Example 2: To sum up data separately classified by code (1 to 9)   (IND, DSZ)

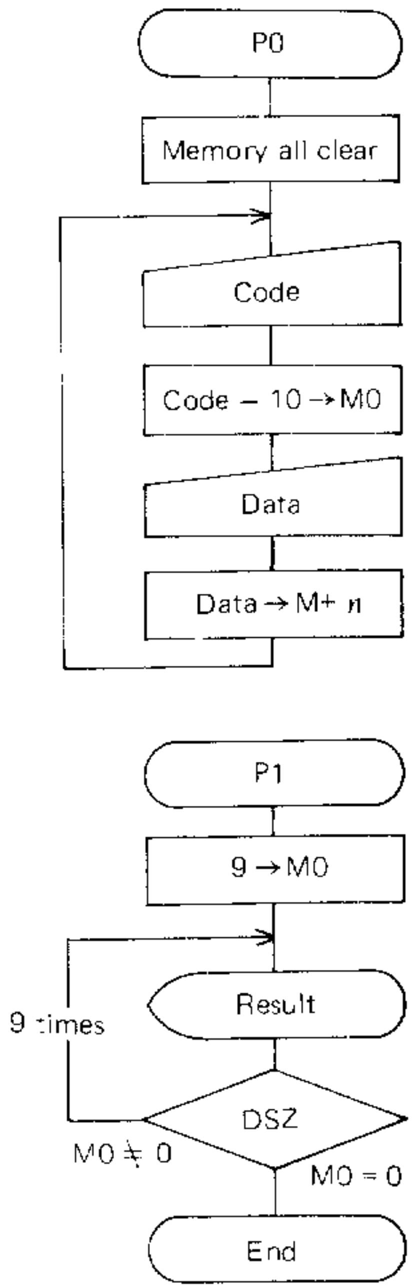
- **Program**

```
P0   INV MAC,
LBL 1,AC, HLT, –, 1, 0, =, Min 0, AC, HLT,
      INV IND, M+ 0, GOTO 1.                    14 steps
P1   9, Min 0.
LBL 1, INV IND, MR 0, HLT,
      IVN DSZ, GOTO 1,                          8 steps
```

- **Operation**

[P0]
Code [EXE]
Data [EXE]
Repeat the above steps.
At the end [P1] ········ Sum of data of code 1
[EXE] ········ Sum of data of code 2
⋮
[EXE] ········ Sum of data of code 9

- **Flowchart**



Example 3: To input data into nine M-registers sequentially and display the data

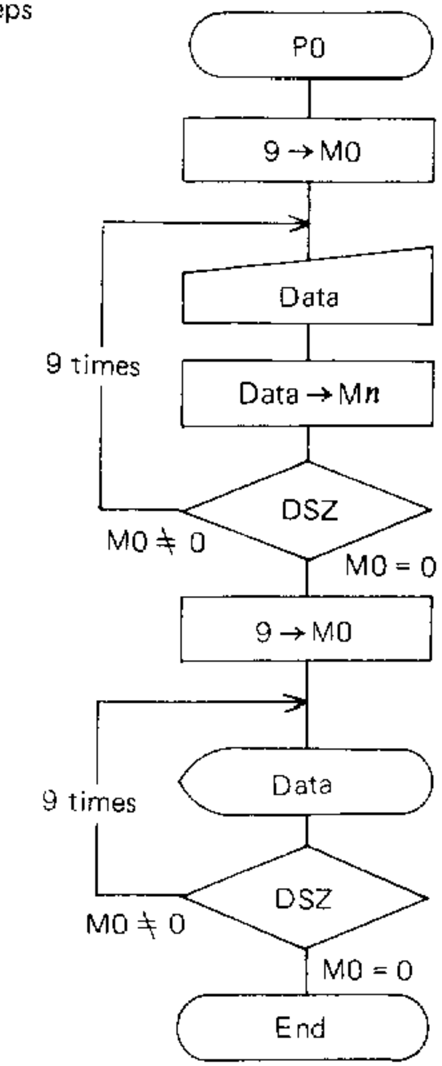
- **Program**

```
P0   9, Min 0,
LBL 1, AC, HLT, INV IND, Min 0,
      INV DSZ, GOTO 1,
      9, Min 0,
LBL 2, INV IND, MR 0, INV PAUSE,
      INV DSZ, GOTO 2,                          17 steps
```

- **Operation**

[P0]
Data 1 [EXE]
Data 2 [EXE]
⋮
Data 9 [EXE] ········ Data are displayed sequentially in intervals of a second.

- **Flowchart**



Example 4: To round data to n decimal places

- **Program 1**

INV P8 Min 9, INV INT, X ↔ M 9, INV FRAC, INV RND n, M− 9, MR 9, 7 steps

* This program will not work well if the first decimal place is zero.

- **Program 2**

INV P8 X, n, INV 10^x, Min 9, +, ·, 5, =, INV INT, ÷, MR 9, =, 12 steps

* This program accepts positive data only.

- **Program 3**

INV P8 ÷, INV ABS, Min 9, X, [(, n, INV 10^x, X, X ↔ M 9, +, ·, 5,)], INV INT, ÷, MR 9, =, 17 steps

* Both positive and negative data can be input.

- **Operation (for program 3 with n=3)**

12.3456 [INV] [P8] ⟶ 12.346
4.04444 [+/−] [INV] [P8] ⟶ −4.044

Example 5: Decimal-hexadecimal conversion

- **Program**

INV P9 Min 6, [(, 1, 6, Min 9, 2, INV 10^x, Min 8, GSB INV P8,)], 10 steps
P4 Min 6, [(, 2, INV 10^x, Min 9, 1, 6, Min 8, GSB INV P8,)], 10 steps
INV P8 0, Min 0, [(,
LBL 1, INV ISZ, [(, [(, MR 6, ÷, MR 9,)], Min 6, INV FRAC, X, MR 9,
)], INV IND, Min 0, MR 6, INV INT, Min 6, INV x=0, GOTO 2, GOTO 1,
LBL 2, INV IND, M− 0, [(, INV IND, MR 0, X, MR 8,)], INV DSZ, GOTO 2,
÷, MR 8,)], 38 steps

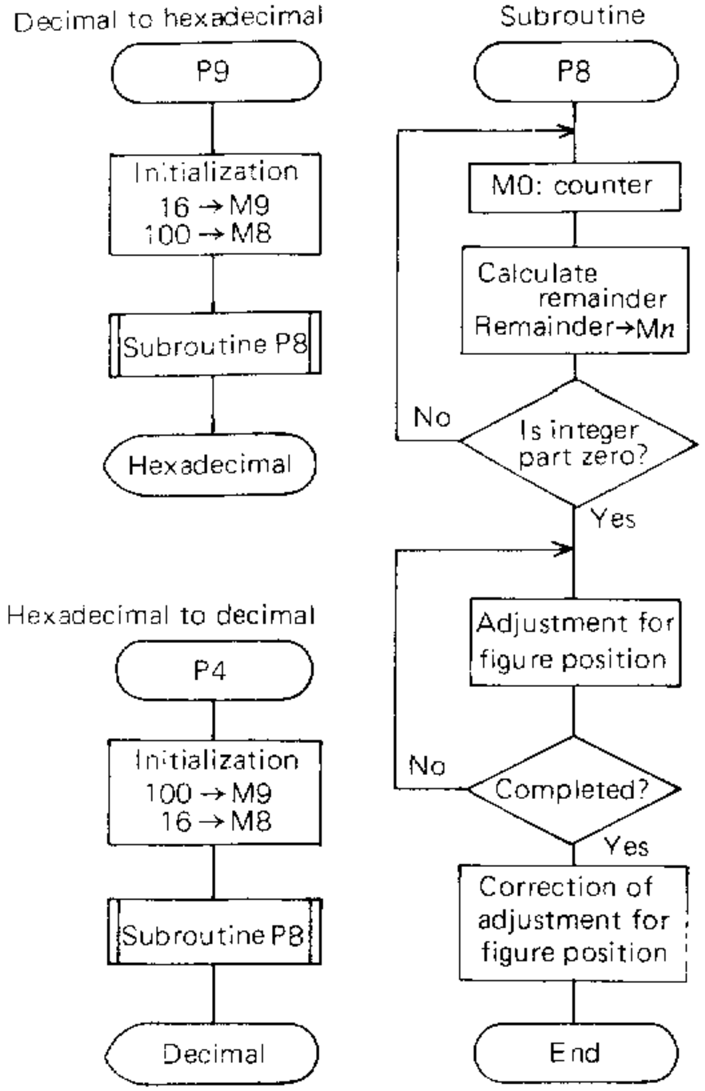
* The ranges of input and output data are as follows.
−655359 ≦ decimal number ≦ 1048575
−9FFFF ≦ hexadecimal number ≦ FFFFF
The [P4] key can be used as the decimal-hexadecimal conversion key for arithmetic operations like other function keys.

- **Operation**

Decimal 1234 is hexadecimal 4D2.
1234 [INV] [P9] → 4:13:02
Decimal −600000 is hexadecimal −927C0.
600000 [+/−] [INV] [P9] → −9:02:07:12:00
Hexadecimal A2B3 is decimal 41651.
10:02:11:03 [P4] → 41651
Hexadecimal 3FC5 + 77ED is hexadecimal B7B2.
3:15:12:05 [P4] [+] 7:07:14:13 [P4] [=] [INV] [P9] → 11:07:11:02

Decimal	Hexadecimal	Decimal	Hexadecimal
0	00	8	08
1	01	9	09
2	02	10	A
3	03	11	B
4	04	12	C
5	05	13	D
6	06	14	E
7	07	15	F

- **Flowchart**



Example:
Conversion of decimal 1500 to hexadecimal 5EC
1500 → 5EC

1 → M0 1500 ÷ 16 = 93
Remainder 12 → M1
2 → M0 93 ÷ 16 = 5
Remainder 13 → M2
3 → M0 5 ÷ 16 = 0
↓ to LBL 2
Remainder 5 → M3

5 x 100 2 → M0
(5 x 100 + 13) x 100 1 → M0
{(5 x 100 + 13) x 100 + 12} x 100
0 → M0
↓ End of repetition

$$\frac{\{(5\times 100+13)\times 100+12\}\times 100}{100}$$

Specifications

Basic features

Basic operations: arithmetic operations (addition, subtraction, multiplication and division with capability of judging priority of operation), negative numbers, exponents, parentheses of up to 10 levels and constants permitted.
Built-in functions: trigonometric and inverse trigonometric functions (with angle in degrees, radians or gradients), hyperbolic and inverse hyperbolic functions, logarithmic and exponential functions, inverse, factorial, square root, square, raising to power of high orders, extracting roots of high orders, decimal↔sexagesimal conversion, coordinate transformation (rectangular to polar and vice versa), absolute value, removing integer part, removing fraction part, percentages, random number, π
Statistical functions: standard deviation (2 kinds), mean, sum, square sum, number of data
Memory: 22 (nonvolatile) registers
Range of number: $\pm 1 \times 10^{-99}$ to $\pm 9.999999999 \times 10^{99}$ and 0, internal operations use 12-digit mantissas
Decimal point: full floating with underflow

Programmable features

Number of steps: up to 256 (nonvolatile, 1 step performs a function)
Jump: unconditional jump (GOTO) up to 10 pairs, conditional jump ($x=0, x\geqq 0, x=F, x\geqq F$), conditional jump for loop control (ISZ, DSZ)
subroutine call (GSB), up to 9 subroutines, up to 4 levels
Number of programs storable: up to 10 (P0 to P9)
Checking and editing functions: check, debug, deletion, addition, etc.
Indirect addressing: for M-register, destination of jump, calling subroutine
Miscellaneous functions: "manual jump" (GOTO), temporary suspension of execution (PAUSE), command code and step number displayed during check, FA-1 adapter (option) for cassette tape recorder I/O

Capacity:

	Input range	Output accuracy
Entry/basic calculations:	10 digit mantissa, or 10 digit mantissa plus 2 digit exponent up to $10^{\pm 99}$	
Scientific calculations:		
$\sin x$, $\cos x$, $\cos^{-1} x$	$\lvert x\rvert < 1440°$ (8π rad, 1600 gra)	±1 in the 10th digit
$\sin^{-1} x$, $\cos^{-1} x$	$\lvert x\rvert \leqq 1$	– " –
$\tan^{-1} x$	$\lvert x\rvert < 1 \times 10^{100}$	– " –
$\log x / \ln x$	$0 < x < 1 \times 10^{100}$	– " –
10^x	$\lvert x\rvert < 100$	– " –
e^x, $\sinh x$, $\cosh x$	$-227 \leqq x \leqq 230$	– " –
$\tanh x$	$-115 \leqq x \leqq 230$	– " –
$\sinh^{-1} x$	$\lvert x\rvert < 1 \times 10^{50}$	– " –
$\cosh^{-1} x$	$1 \leqq x < 1 \times 10^{50}$	– " –

$\tanh^{-1} x$	$\lvert x \rvert < 1$	— " —
$\sqrt{x}$, x^y,	$0 \leqq x < 1 \times 10^{100}$	— " —
$x^{1/y}$ ($\sqrt[y]{x}$)	$0 \leqq x < 1 \times 10^{100}, y \neq 0$	— " —
x^2	$\lvert x \rvert < 1 \times 10^{50}$	— " —
$1/x$	$\lvert x \rvert < 1 \times 10^{100}$, $x \neq 0$	— " —
$x!$	$0 \leqq x \leqq 69$ (x: natural number)	— " —
R → P	$\lvert x \rvert < 1 \times 10^{50}$, $\lvert y \rvert < 1 \times 10^{50}$	— " —
P → R	$\lvert r \rvert < 1 \times 10^{100}$	— " —
	$\lvert \theta \rvert < 1440°$ (8π rad, 1600 gra)	
Decimal to sexagesimal	Within ± 27777.77777	
π	10 digits	

- **Display:** 10-digit mantissa (including negative sign), 2-digit exponent, liquid crystal, possible sexagesimal representation, INV, K, HLT, RUN, WRT, PCL, DEG, RAD, and GRA modes displayed
- **Error check function:** Overflow (10^{100} or more) and fatal error for execution detected ("*E.*" displayed)
- **Power consumption:** 0.0008 W (calculator alone)
  0.001 W (with FA-1 adapter)
- **Power source:** 2 silver oxide batteries (Type: G-13, UCC357, 10L14, RW-22 or RW-42). The calculator gives approximately 1300 hours (1000 hours with FA-1 adapter) continuous operation on type G-13.
- **Auto power-off:** Power will be automatically cut off about 14 minutes after end of operation.
- **Usable temperature:** 0°C – 40°C (32°F – 104°F)
- **Dimensions:** 9.6mmH x 71mmW x 141.2mmD
  (3/8"H x 2-3/4"W x 5-1/2"D)
- **Weight:** 103g (3.6 oz) including batteries.

Felicitaciones por haber adquirido esta calculadora científica programable.
Esta calculadora super-fina, de tamaño de bolsillo con pantalla de cristal líquido es capaz de realizar cálculos comunes paso a paso o cálculos programados. Equipada con 22 memorias independientes, puede computar valores de 51 tipos de funciones. La secuencia de operación es básicamente la misma de las expresiones matemáticas comunes (lógica algebraica verdadera) con capacidad de paréntesis hasta diez niveles. Para la operación programada, puede almacenar un programa o varios, hasta un total de 256 pasos.
El adaptador FA-1, provisto como opción, permite conectar un grabador a cintas de cassette a la calculadora y almacenar programas y contenidos de memorias en las cintas de cassettes para recargar más tarde.

Este manual contiene una 1ª parte, de cálculos paso a paso, y la 2ª parte, de cálculos programados. Le aconsejamos que estudie cuidadosamente el funcionamiento de la calculadora antes de usarla.

INDICE