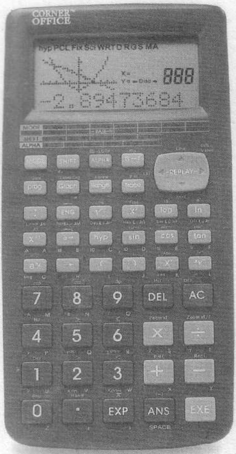
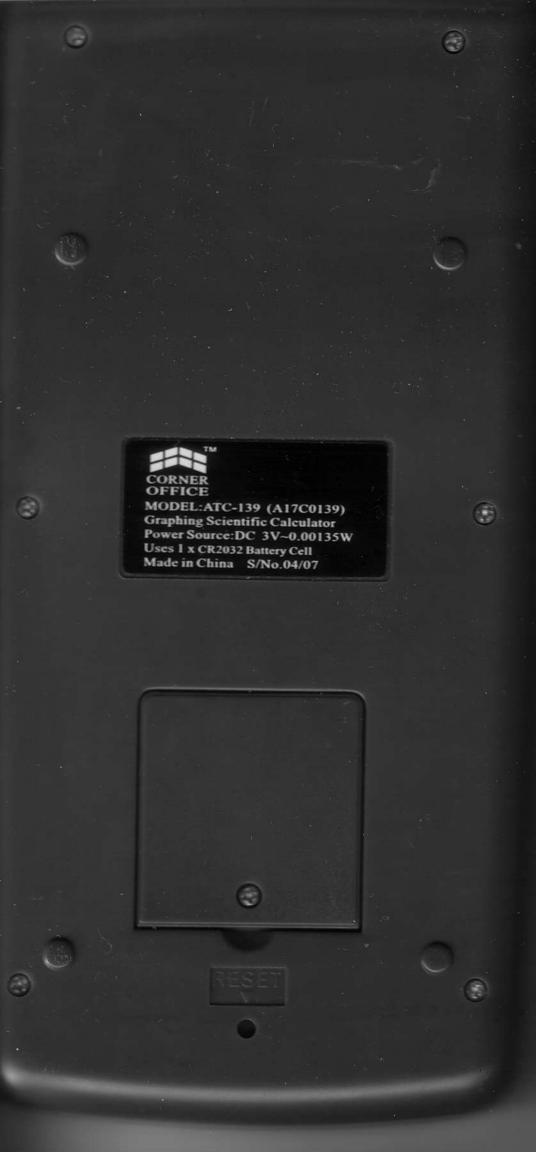
CORNER OFFICE™

Graphing Scientific Calculator

Operating Instructions



-FOR PERSONAL USE ONLY-


CORNER
OFFICE
MODEL:ATC-139 (A17C0139)
Graphing Scientific Calculator
Power Source:DC 3V~0.00135W
Uses 1 x CR2032 Battery Cell
Made in China S/No.04/07
RESET

CONTENTS

(Add three to get the PDF file page).

(A) INTRODUCTION

This unit is a advanced programmable calculator. Besides versatile scientific functions, a wide variety of useful graphs can be produced. Manual calculations can be easily performed following written formulas. A replay function is provided that allows confirmation or correction when key operation errors occur.

Programs can also be inputted by following true algebraic logic, so repeat and/or complex calculations are simplified. Be sure to carefully read this manual and keep it handy for future reference.

Important – Reset your calculator before using it the first time!
Always backup data!

HANDLING THE BATTERY POWER SUPPLY

Battery Safety:

Incorrect battery handling may cause it to burst or leak, and possibly damage equipment.
Make sure the new battery is the correct type.
Make sure the "+" faces up.
Keep battery out of the reach of children.
Do not dispose battery in fire.

Replacing battery :

- Make sure power is off before replacing the battery
- Make sure to store programs or data before performing this operation
- After replacing the battery, be sure to switch the calculator on and then perform the reset operation.

1. Remove the battery compartment door.
2. Remove the used battery.
3. Install the new battery and ensure the "+" side facing up.
4. Replace battery compartment door.
5. Turn the power on.

Auto power off function

The unit switches off approximately 6 minutes after the last key operation. Once this occurs, power can be restored by pressing the AC key. Numeric values in the memories, programs, or calculations made are unaffected when power is switched off.

Reset operation:

Strong external electrostatic charges can cause the calculator to malfunction. Should this happen, perform the following procedures to reset the calculator:

Warning: The following procedures clear all data from the memory of the calculator. To avoid the loss of important data, be sure to always keep a written backup copy.

Switch the power on

Press the RESET button on the back of the calculator with a thin, pointed object to reset the calculator and clear memory contents.

Remarks: Never press the RESET button while internal operations are being performed. Doing so may cause irreversible damage to the memory of your calculator.

PRECAUTIONS:

The calculator is made of precision components. Never attempt to disassemble the unit.
The display will go blank and keys will not operate during calculations. When you are operating the keypad, be sure to watch the display to ensure all your key operations are being performed correctly.
Avoid strong vibrations or impacts during program execution which can cause execution to stop or damage the calculator's memory.
Avoid dropping your calculator
Avoid using volatile liquids such as thinner or benzene to clean the unit. Wipe it with a soft, dry cloth or with a cloth that has been dipped in a solution of water and mild detergent.

Do not store the calculator or leave it in areas exposed to high temperature or humidity, or large amounts of dust.
Never leave a dead battery in the battery compartment. It can leak and damage the unit.

The manufacturer and its suppliers will not be liable to you or any other person for any damages, expenses, lost profits, lost savings or any other damages arising out of loss of data and / or formulas arising out of malfunctions, repairs or battery replacement. The user should prepare physical records of data to protect against such data loss.

● GENERAL GUIDE

IMPORTANT — the keys of the scientific calculator perform more than one function. The following explains all of the operations of each key. Read this section carefully before using your calculator for the first time.

● KEY MARKINGS

The keys of this unit perform a number of different functions. The key illustrated below, for example, is used to perform 4 different functions: X^{-1}, X!, A, /A.

Note the followings, concerning the key illustrated above.

Mode	Function
Direct Input	x^{-1}
SHIFT	x !
ALPHA	A
BASE-N, HEX	/A

The keypad is color-coded to help you quickly determine the key sequence you have to perform for each function. The following table shows how to interpret the various key markings.

Keypad Marking	Meaning
Orange	Press SHIFT and then key
Pink	Press ALPHA and then key
Green	Press key in MODE

In addition to the above, there are a number of key sequences indicated on the panel beneath the display (such as [ALPHA] [2Σx]). These key sequences can be used in the SD or LR mode only.

● TO READ THE DISPLAY

● Display indicators

The following indicators appear on the display to show you the current status of the calculator at a glance

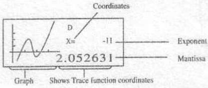
S: SHIFT key pressed.
M: MODE key pressed.
A: ALPHA key pressed
Sei: Number of significant digits specified
Fix: Number of decimal places specified
hyp: hyp key pressed
D: Degrees specified as the unit of angular measurement
R: Radians specified as the unit of angular measurement
G: Grads specified as the unit of angular measurement
WRT: Program write mode (MODE 2) specified
PCL: Program clear mode (MODE 3) specified
X=, Y=: Indicates current x- and y- coordinate location of trace function pointer
← → : Indicates display consists of more than 12 characters. ← indicates extra characters run off left side of display, → indicates characters run off right side.
Disp: Indicates displayed value is intermediate result

- **About the display layout**

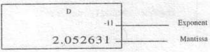
The display consists of a graphing area, as well as an area for indicators and characters. You can monitor the status of the calculator and programs by viewing the display.

Example:

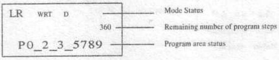
Graph Display



Calculation Display



Mode Status Display



- **Exponential display**

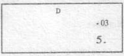
During normal calculation, this unit is capable of displaying up to 10 digits. Values that exceed this limlt, however, are displayed in exponential format. You can choose between 2 different types of exponential display formats.

NORM 1 mode: $10^{-2}(0.01) > |x|$, $|x| \geq 10^{10}$

NORM 2 mode: $10^{-9}(0.000000001) > |x|$, $|x| \geq 10^{10}$

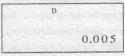
Selection of these modes can be carried out by pressing [MODE] [9] [EXE], when no specification has been made for the number of decimal places or significant digits. The present status is not displayed, so it is necessary to perform the following procedure to specify either display format:

1 [÷] 200 [EXE]



NORM 1 mode

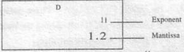
1 [÷] 200 [EXE]



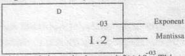
NORM 2 mode

NOTE: All of the examples in this manual show calculation results using the NORM 1 mode.

How to interpret exponential format



1.2^{11} indicates that the result is equivalent to 1.2×10^{11}. This means that you should move the decimal point in 1.2 eleven places to the right. Since the exponent is positive, this results in the value 120,000,000,000.

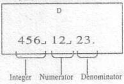


1.2^{-03} indicates that the result is equivalent to 1.2×10^{-03}. This means that you should move the decimal point in 1.2 three places to the left, since the exponent is negative. This results in the value 0.0012.

- **Special display formats**

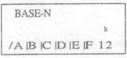
Special display formats are used for the representation of fraction, hexadecimal and sexagesimal values.

Fraction value display



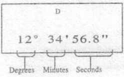
Display of 456 $\frac{12}{23}$

Hexadecimal value display



Display of ABCDEF12 (-1412567278)

Sexagesimal value display



Display of 12° 34' 56.8"

KEY OPERATIONS

- **Special operation keys**

[SHIFT] Shift key

Press when using the function commands and functions marked in orange on the keypad. An "s" will appear on the display to indicate that [SHIFT] has been pressed. Pressing [SHIFT] again will cause the "s" to disappear from the display.

[MODE] Mode key

Use the [MODE] key in combination with [.], [1] through [9], [+], [−], [×] and [÷] to specify the calculation mode and the unit of angular measurement.

[MODE] [1] — For manual calculations and program execution (RUN mode).

[MODE] [2] — WRT displayed. For writing or checking programs.

[MODE] [3] — PCL displayed. For clearing programs.

[MODE] [4] — D displayed. if [EXE] is pressed, unit of angular measurement is specified as degrees.

[MODE] [5] — R displayed. if [EXE] is pressed, unit of angular measurement is specified as radians.

[MODE] [6] — G displayed. if [EXE] is pressed, unit of angular measurement is specified as grads.

[MODE] [7] — Fix displayed. Entering a value from 0 to 9 followed by [EXE] will specify the number of decimal places according to the value entered. **Ex.** [MODE] [7] 3 [EXE] → **Three decimal places.**

[MODE] [8] — Sci displayed. Entering a value from 0 to 9 followed by [EXE] will specify the number of significant digits from 1 to 10. **Ex.** [MODE] [8] **5** [EXE] → **5 significant digits**
[MODE] [8] **0** [EXE] → **10 significant digits**

[MODE] [9] — Pressing [1] [EXE] will cancel the specified number of decimal places or pressing [2] [EXE] will cancel the specified number of significant digits

*** If you have not specified the number of decimal places or the number of significant digits, you can press [MODE] [9] [1] OR [2] [EXE] and the change the range of the exponential display (NORM 1 / NORM 2)

*** With the exception of the BASE-N mode, modes [7] ~ [9] can be used in combination with the manual calculation modes.

*** The mode last selected is retained in memory when the unit's power is switched off.

[MODE] [.] — Defm displayed. Entering a value followed by [EXE] will specify the number of memories available.

Ex. [MODE] [.] **10** [EXE] → **Number of memories available increased by 10.**

[MODE] [+] — Specifies COMP mode for arithmetic calculation or function calculation (program execution possible)

[MODE] [−] — For binary, octal or hexadecimal calculations / conversions (BASE-N mode)

[MODE] [×] — For standard deviation calculations (SD mode).

[MODE] [÷] — For regression calculations (LR mode).

The x^y and $^x\sqrt{\ }$ functions are not available in the LR mode. To use these functions, first perform the statistical operations and then press [MODE] [+] to enter the COMP mode.

[SHIFT] [MODE] [4] — Pressed after a numeric value representing degrees (°) is input.

[SHIFT] [MODE] [5] — Pressed after a numeric value representing radians (r) is input.

[SHIFT] [MODE] [6] — Pressed after a numeric value representing grads (g) is input.

A-LOCK

ALPHA **Alphabet key**

Press to input alphabetic characters or special characters. Pressing ALPHA displays "A" and allows the input of only one character. After that, the unit returns to the status it was in before the ALPHA key was originally pressed. Pressing SHIFT followed by ALPHA displays "S""A" which will lock the unit in this mode and allow consecutive input of alphabetic characters until ALPHA is pressed again.

Goto

Prog Program / Goto key

Press Prog, enter a value from 0 to 9 and then press EXE to execute a program.

Ex. Prog 1 EXE →**Execution of Program 1 begins**

Pressing SHIFT followed by Goto will cause Goto to appear on the display. This is a jump command used in programs.

Line / Lbl / INS / -REPLAY- / Value / x↔y

Cursor / Replay keys

The ◁ key moves the cursor to the left, ▷ moves the cursor to the right. In the Plot function, the △ key moves the pointer up, and ▽ moves the pointer down. Holding any of the keys down will cause the cursor to continuously move in the respective direction.

Once a formula or numeric value is input and EXE is pressed, the key ◁ and ▷ key become replay keys. In this case, pressing ▷ displays the formula or numeric value from the beginning, while pressing ◁ displays it from the end. This allows the formula to be executed again by changing the values.

Pressing SHIFT followed by INS displays the insert cursor ([ ]). Entering a value while the insert cursor is displayed inserts the value in the position immediately preceding the insert cursor location.

Pressing SHIFT followed by Lbl enters the "Lbl" (Label) command.

Pressing SHIFT followed by Line makes it possible to produce line graphs of regression lines

After you draw a graph, press SHIFT Value to display a value that shows the x-coordinate for the current location of the pointer on the graph. You can switch between display of the x-coordinate and the y-coordinate by pressing SHIFT x↔y.

Mcl

DEL **Delete key**

Press to delete the character at the current position of the cursor. When the character is deleted, everything to the right of the cursor position will shift one space to the left.

Pressing SHIFT Mcl EXE will clear the memory contents.

OFF ON

AC **All clear / Power ON / Power OFF key**

Press to clear all input characters or formulas. You can also use this key to clear the error message from the display.

Press ON to switch the power on (even if power was switched off by the Auto Power Off function).

Press SHIFT OFF to switch the power off. Note that mode setting and memory contents are protected even when power is turned off.

EXE **Execute key**

Press to obtain the result of a calculation or to draw a graph. Pressed after data input for a programmed calculation or to advance to the next execution after a calculation result is obtained.

(-)

Ans **Answer / Minus key**

SPACE

Pressing Ans followed by EXE will recall the last calculation result. It can be recalled by Ans EXE even after it has been cleared using the AC key or by switching the power off. When used during program execution, the last result calculated can be recalled.

Press following SHIFT key to entering a numeric value to make that value negative. Ex: **-123→SHIFT (-) 123**

Press following ALPHA key to input a space.

- **Numeric / Decimal point / Exponent input keys**

Rnd ≒ Ran# π
[0] ~ [9] [.] [EXP]
z (r) ° [ ]

When entering numeric values, enter the number in order. Press the [.] key to enter the decimal point in the desired position.

Ex. To input 1.23×10^{-6} → press 1.23 [EXP] [SHIFT] [(-)] 6.

[SHIFT] key combinations for the various modes are as follows:

COMP mode ([MODE] [+])	BASE-N mode ([MODE] [-])	SD mode ([MODE] [×])	LR mode ([MODE] [×])
[⇒] [=] [≠] [Isz] [≤] [≥] [Dsz] [<] [>] [Rnd] [Ran#] [π]	[⇒] [=] [≠] [Isz] [≤] [≥] [Dsz] [<] [>] Pol(, Rec(, Rnd, Ran# and π cannot be used in this mode	[⇒] [=] [≠] [Isz] [≤] [≥] [x̄] [xσn] [xσn-1] [Rnd] [Ran#] [π] Standard deviation functions can be used	[A] [B] [r] [ȳ] [yσn] [yσn-1] [x̄] [xσn] [xσn-1] [Rnd] [Ran#] [π] Paired variable statistic functions can be used

- **Calculation keys**

Pol(Rec(Zoom xf Zoom x1/f
[+] [−] [×] [÷] **Arithmetic operation keys**
X Y S T

For addition, subtraction, multiplication, and division, enter the calculation as it reads. [SHIFT] key combinations for the various modes are as follows:

COMP mode

[Zoom xf] [Zoom x1/f] —— Following [SHIFT], this key causes the graph currently shown on the display to be enlarged or reduced in accordance with the factor setting.

COMP mode or SD mode

[Pol(] [Rec(] —— Coordinate transformation

LR mode

[x̂] [ŷ] —— Estimated value calculation of x̂ and ŷ

[Pol(] [Rec(] —— Coordinate transformation

- **Graph keys**

Used to produce a variety of graphs. These keys cannot be used in the **BASE-N** mode.

Zoom Org
[Graph] **Graph / Original zoom key**

Press before entering a formula to be used for a graph ("Graph Y=" appears on the display).

Press to return an enlarged or reduced graph to its original size.

When pressed following the [ALPHA] key, the results of each section of the programmed calculations or consecutive calculations are sequentially displayed with each press of [EXE].

Factor
[Range] **Range / Factor key**

Used to confirm or set the range and size of graphs

Press following [SHIFT] to magnify or reduce the upper and lower ranges of graphs.

Press following [ALPHA] in order to assign the same value to more than one memory.

Ex. To store the value 456 to memories A through F: 456 [→] [SHIFT] [ALPHA] [A] [~] [F] [EXE]

Plot
[Trace] **Trace / Plot key**
?

Used to trace over an existing graph and display the x or y coordinate value.

Press following [SHIFT] to plot a point on the graph screen.

To indicate data input within a programmed calculation or repeat calculation, press [ALPHA] and then [?]

Cls
[G↔T] **Graph-Text / Clear screen key**

Switches between the graph display and text display

[SHIFT] [Cls] [EXE] clears the graph display ("done" is displayed)

- **Function keys**

Press for functional calculation. Various uses are available in combination with the [SHIFT] key, or depending on the mode being used.

[:] **Multi-statement key**

xnor k **Graph-Text / Clear screen key**

Press to separate formulas or commands in programmed calculations or consecutive calculations.

The result of such combination is known as a multi-statement.

Press following [SHIFT] in the BASE-N mode to enter the logical operation for negation of logical sums (xnor).

[ENG] **Engineering / Negation key**

Not m

Press to convert a calculation result to an exponential display whose exponent is a multiple of three.

(Kilo 10^3 = k, mega 10^6 = M, . giga 10^9 = G, milli 10^{-3} = m, micro 10^{-6} = μ, nano 10^{-9} = n, pico 10^{-12} = p)

When obtaining logical negation for a value in the BASE-N mode, press prior to entering the value.

Press following the [SHIFT] key in the BASE-N mode to obtain the exclusive logical sum.

Int

[√] **Square root / Integer key**

Dec μ

Press prior to entering a numeric value and press [EXE] to obtain the square root of that value

When pressed following the [SHIFT] key, the integer portion of a value can be obtained.

Press followed by [EXE] in the BASE-N mode to specify the decimal calculation mode.

When pressed following the [SHIFT] key in the BASE-N mode, the subsequently entered value is specified as a decimal value.

Frac

[x^2] **Square / Fraction key**

Hex n

Press after a numeric value is entered and press [EXE] to obtain the square of that value.

Press following [SHIFT] key prior to inputting number in order to obtain fraction part of that number.

Press followed by [EXE] in the BASE-N mode to specify the hexadecimal calculation mode.

When pressed following the [SHIFT] key in the BASE-N mode, the subsequently entered value is specified as a hexadecimal value.

10^x

[log] **common logarithm / Antilogarithm key**

Bin p

Press prior to entering a value and press [EXE] to obtain the common logarithm of that value.

When pressed following the [SHIFT] key, the subsequently entered value becomes an exponent of 10.

Press followed by [EXE] in the BASE-N mode to specify the binary calculation mode.

When pressed following the [SHIFT] key in the BASE-N mode, the subsequently entered value is specified as a binary value.

e^x

[In] **Natural logarithm / Exponential key**

Oct f

Press prior to entering a value and press [EXE] to obtain the common logarithm of that value.

When pressed following the [SHIFT] key, the subsequently entered value becomes an exponent of e

Press followed by [EXE] in the BASE-N mode to specify the octal calculation mode.

When pressed following the [SHIFT] key in the BASE-N mode, the subsequently entered value is specified as an octal value.

x!

[x^{-1}] **Reciprocal / Factorial key**

/A A

Press after entering a value and press [EXE] to obtain the reciprocal of that value.

When pressed following the [SHIFT] key, the factorial of a previously entered value can be obtained.

Press in the BASE-N mode to enter A (10_{10}) of a hexadecimal value.

[° ' "] **Degree / minute / second key** (**decimal ↔ sexagesimal key**)

1B B

Press to enter sexagesimal value (degree / minute / second or hour / minute / second).

Ex: 78° 45' 12" → 78[° ' "] **45**[° ' "] **12**[° ' "]

When pressed following the [SHIFT] key, a decimal based value can be displayed in degrees/minutes/seconds (hours/minutes/seconds).

Press in the BASE-N mode to enter B (11_{10}) of a hexadecimal value.

hyp Hyperbolic key

Pressing hyp and then sin, cos, or tan prior to entering a value produces the respective hyperbolic function (sinh, cosh, tanh) for the value.

Pressing SHIFT, then hyp and then sin, cos or tan prior to entering a value produces the respective inverse hyperbolic function ($\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$) for the value.

Press hyp in the BASE-N mode to enter C (12_{10}) of a hexadecimal value.

sin cos tan **Trigonometric function / Inverse trigonometric function keys**

Press one of these keys prior to entering a value to obtain the respective trigonometric function for the value

Press SHIFT and then one of these keys prior to entering a value to obtain the respective inverse trigonometric function for the value.

Press in the BASE-N mode to enter D, E, F (13_{10}, 14_{10}, 15_{10}) of a hexadecimal value.

ab/c **Fraction / Negative key**

Use this key for input of simple fractions and mixed fractions.

Ex. To input 23/45: 23 ab/c 45 To input 2-3/4: 2 ab/c 3 ab/c 4

For improper fractions, press this key following SHIFT (indicated by SHIFT a d/c in this manual)

Press in the **BASE-N** mode prior to entering a value to obtain the negative of that value. The negative number is the two's complement of the value entered.

→ **Assignment key**

Press prior to entering a memory to assign the result of a calculation to the memory

Ex To assign the result of 12+45 to memory A: 12+45 → ALPHA A EXE

Press this key following SHIFT to clear all data from the statistical memories.

( ) **Parenthesis keys**

Press the open parenthesis key and the closed parenthesis key at the position required in a formula.

When pressed following the SHIFT key, a comma or semicolon can be inserted to separate the arguments in coordinate transformation or consecutive calculations.

x^y **Power / Absolute value key**

Enter x (any number), press this key, and then enter y (any number) to compute x to the power of y. In the SD or LR mode, this function is only available after pressing the SHIFT key.

Press following the SHIFT key to obtain the absolute value of a subsequently entered numeric value.

Press in the BASE-N mode to obtain a logical product ("and").

Press in the SD or LR mode to delete input data.

$\sqrt[x]{\ }$ Root / Cube root key

Enter x, press this key, and then enter y to calculate the xth root of y. In the SD or LR mode, this function is only available after pressing the SHIFT key.

Press following the SHIFT key to obtain the cube root of a subsequently entered numeric value.

Press in the BASE-N mode to obtain a logical sum ("or")

Used as a data input key in the SD or LR mode.

- **Contrast**

If the display becomes dim and difficult to read, even if you increase contrast, it probably means that battery power is getting low. In such a case, replace battery as soon as possible. After replacing battery, perform the RESET operation.

● GENERAL CALCULATION INFORMATION

1.4.1 Calculation priority sequence

This calculator employs true algebraic logic to calculate the parts of a formula in the following order:

1) Coordinate transformation: Pol (x, y), Rec(r, θ)
2) Type A functions: With these functions, the value is entered and then the function key is pressed.
   x^2, x^{-1}, $x!$, $^{\circ}$, ^{r}, ^{g}, ° ' "
3) Power / Root: x^y, $\sqrt[x]{\ }$
4) Fractions: a*b/c*
5) Abbreviated multiplication format in front of π, memory or parenthesis 2π, 4R, etc.
6) Type B functions: With these functions, the function key is pressed and then the value is entered.
   $\sqrt{\ }$, $\sqrt[3]{\ }$, log, ln, e^x, 10^x, sin, cos, tan, $\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$, sinh, cosh, tanh, $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$, (-), Abs, Int, Frac, parenthesis, (following in BASE-N calculations only) d, h, b, o, Neg, Not
7) Abbreviated multiplication format in front of Type B functions: $2\sqrt{3}$, A log2, etc.
8) ×, ÷
9) +, −
10) and (BASE-N calculations only)
11) or, xor, xnor (BASE-N calculations only)
12) Relational operators <, >, =, ≠, ≤, ≥

When functions with the same priority are used in series, execution is performed from right to left.

$e^x \ln \sqrt{120} = e^x [\ln(\sqrt{120})]$

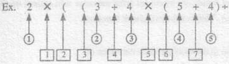
Otherwise, execution is from left to right.

Compound functions are executed from right to left.

Anything contained within parentheses receives highest priority.

● Number of stacks

This calculator uses a memory known as a "stack" for temporary storage of low priority numeric values and commands (functions, etc.). The numeric value stack has 10 levels, while the command stack has 24. If a formula exceeds the stack space available, a stack error (Stk ERROR) message appears in the display.



Numeric stack value

①	2
②	3
③	4
④	5
⑤	4
⋮	

Command stack

1	×
2	(
3	(
4	+
5	×
6	(
7	+
⋮	

*** Calculations are performed in sequence, with the highest priority operation first. Once a calculation is executed, it is cleared from stack.

● Calculation modes

This unit features modes for manual calculations, storing programs, and modes for general as well as statistical calculations.

Operation modes

There are a total of three operation modes.

1) RUN mode: Graph production as well as manual calculations and program executions.
2) WRT mode: Program storage and editing.
3) PCL mode: Deletion of stored programs.

Calculation modes

There are a total of four calculation modes which are employed according to the type of calculation.

1) COMP mode: General calculations, including functional calculations.
2) BASE-N mode: binary, octal, decimal, hexadecimal conversion and calculations, as well as logical operations. Function calculations and graph drawing cannot be performed.
3) SD mode: Standard deviation calculation (single-variable statistics).
4) LR mode: Regression calculation (paired-variable statistics)

With so many modes available, calculations should always be performed after confirming which mode is active.

*** When the power of the unit is switched off (including AUTO Power Off), the current system mode is cancelled, and the unit will be set to the RUN mode when switched on again. However, the calculation mode, number of decimal place setting ([MODE] [7] n), number of significant digits ([MODE] [8] n), and angle unit (Deg, Rad, Gra) will be retained in memory.

*** To return to standard operation (initialized state) press [MODE] [+] (COMP mode) – [MODE] [1] (RUN mode) – [MODE] [9] (Norm mode).

- **Number of input / output digits and calculation digits**

The allowable input / output range (number of digits) of this unit is 10 digits for a mantissa and 2 digits for an exponent. Calculations, however, are internally performed with a range of 12 digits for mantissa and 2 digits for an exponent.

Example: $3 \times 10^5 \div 7 =$

3 [EXP] 5 [÷] 7 [EXE] → 42857.14286

3 [EXP] 5 [÷] 7 [-] 42857 [EXE] → 0.1428571

*** Calculation results greater than 10^{10} (10 billion) or less than 10^{-2} (0.01) are displayed in exponential form.

Example: **123456789 x 9638**

123456789 [x] 9638 [EXE] → 1.189876532 ^{12}

Once a calculation is completed, the mantissa is rounded off to 10 digits and displayed. The displayed mantissa can be used for the next calculation.

Example: $3 \times 10^5 \div 7 =$

3 [EXP] 5 [÷] 7 [EXE] → 42857.14286

[-] 42857 [EXE] → 0.14286

*** Values are stored in memory with 12 digits for the mantissa and 2 digits for the exponent.

- **Overflow and errors**

If the calculation range of the unit is exceeded, or incorrect inputs are made, an error message will appear on the display window and subsequent operation will be suspended. This is the error check function.

The following operations will result in errors:

1) The answer, whether intermediate or final, or any value in memory exceeds the value of $\pm 9.999999999 \times 10^{99}$.
2) An attempt is made to perform functional calculations that exceed the input range.
3) Improper operation during statistical calculations. **(Ex. Attempting to obtain $\bar{x}$ or xσ n without data input)**
4) The capacity of the numeric value stack or the command stack is exceeded.
   (Ex. Entering 23 successive [(]'s followed by 2 [+] 3 [x] 4 [EXE])
5) Even though memory has not been expanded, a memory name such as Z [2] is used.
6) Input errors are made. (Ex. 5 [x] [x] 3 [EXE])
7) When improper arguments are used in commands or functions that require arguments.
   (e.g. Input of an argument outside of the range of 0~9 for Sci or Fix.)

The following error messages will be displayed for the operations noted above:

1~3 Ma ERROR
4 Stk ERROR
5 Mem ERROR
6 Syn ERROR
7 Arg ERROR

Besides these, there is a "Ne ERROR" (nesting error) and a "Go ERROR". These errors mainly occur when using programs.

- **Number of input characters**

This unit features a 127-step area for calculation execution.

One function comprises one step. Each press of numeric or [+][-][x] and [÷] keys comprise one step. Though such operations as [SHIFT] [x!] require two keys operations, they actually comprise only one function and, therefore, only one step. These steps can be confirmed using the cursor. With each press of the ◁ or ▷ key the cursor is moved one step.

Input characters are limited to 127 steps. Usually the cursor is represented by a blinking "_", but once the 121st step is reached the cursor changes to a blinking "■". If the "■" appears during a calculation, the calculation should be divided at some point and performed in two parts.

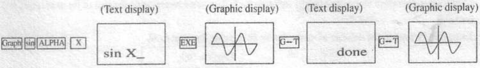
*** When numeric values or calculation commands are input, they appear on the display window from the left. Calculational results, however, are displayed from the right.

• Graphic and text displays

This unit has a graph display for production of graphs, as well as a text display for production of formulas and commands. These two types of display contents are stored independently.

Switching between graph and text displays is performed using the [G↔T] key. Each press of [G↔T] switches from the current type of display to the other.

Example:



Operations to clear the display depend upon the type of display being shown:

Graphs: [SHIFT] [Cls] [EXE] Text: [AC]

Pressing the [AC] key causes a cleared text display to appear if pressed during a graph display.

• Corrections

To make corrections in a formula that is being input, use the ◁ and ▷ keys to move to the position of the error and then press the correct keys.

*** If after making corrections, input of the formula is complete, the answer can be obtained by pressing [EXE]. If, however, more is to be added to the formula, advance the cursor using the ▷ key to the end of the formula for input.

- If an unnecessary character has been included in a formula, use the ◁ and ▷ keys to move to the position of the error and press the [DEL] key. Each press of [DEL] will delete one command (one step)
- If a character has been omitted from a formula, use the ◁ and ▷ keys to move to the position where the character should have been input, and press [SHIFT] followed by the [INS] key. Press [SHIFT] [INS] and insertions can be subsequently performed as desired.

*** When [SHIFT] [INS] are pressed, the letter at the insertion position is surrounded by "[ ]" and blinks. The insert function is activated until you press ◁, ▷, or [AC] or until you press [SHIFT] [INS] again.

• Memory

This unit contains 26 standard memories. Memory names are composed of the 26 letters of the alphabet. Numeric values with 12 digits for a mantissa and 2 digits for an exponent can be stored.

Example: To store 123.45 in memory A:

123.45 [→] [ALPHA] [A] [EXE]

123.45→A_
123.45

Values are assigned to a memory using the [→] key followed by the memory name.

Example: To store the sum of memory A+78.9 in memory B:

[ALPHA] [A] [+] 78.9 [→] [ALPHA] [B] [EXE]

A+78.9→B_
202.35

Example: To add 74.12 to memory B:

[ALPHA] [B] [+] 74.12 [→] [ALPHA] [B] [EXE]

B+74.12→B_
276.47

- To check the contents of a memory, press the name of the memory to be checked followed by [EXE].

  [ALPHA] [A] [EXE] 123.45

- To clear the contents of a memory (make them 0), proceed as follows:

  Example: To clear the contents of memory A only:

  0 [→] [ALPHA] [A] [EXE] 0.

  Example: To clear the contents of all the memory

  [SHIFT] [Mcl] [EXE]

Mcl_
0.

- To store the same numeric value to multiple memories, press [ALPHA] followed by [~].

  Example: to store a value of 10 in memories A through J:

  10 [→] [SHIFT] [ALPHA] [A] [~] [J] [EXE]

10→A~J
10.

- **Memory expansion**

Though there are 26 standard memories, they can be expanded by changing program storage steps to memory. Memory expansion is performed by converting 8 steps to one memory.

Number of memories	26	27	28	•••	36	•••	74	•••	76
Number of steps	400	392	384	•••	320	•••	16	•••	0

Memory is expanded in units of one. A maximum of 50 memories can be added for a maximum total of 76(26+50). Expansion is performed by pressing [MODE], followed by [.], a value representing the size of the expansion, and then [EXE].

Example: To expand the number of memories by 30 to bring the total to 56.

[MODE] [.] 30 → Defm 30_

[EXE] → M-56 S-160

(M-56: Number of memories; S-160: Current number of remaining steps)

The number of memories and number of remaining steps are displayed. The number of remaining steps indicates the current unused area, and will differ according to the size of the program stored. To check the current number of memories press [MODE], followed by [.] and then [EXE].

[MODE] [.] [EXE] → M-56 S-160

To initialize the number of memories (to return the number to 26), enter a zero for the value in the memory expansion sequence outlined above.

[MODE] [.] 0 [EXE] → M-26 S-400

*** Though a maximum of 50 memories can be added, if a program has already been stored and the number of remaining steps is less than the desired expansion, an error will be generated. The size of the memory expansion must be equal to or less than the number of steps remaining.

*** The expansion procedure ([MODE] [.] expansion value) can also be stored as program.

- Using expanded memories

Expanded memories are used in the same manner as standard memories, and are referred to as Z[1], Z[2], etc. The letter Z followed by a value in brackets indicating the sequential position of the memory is used as the memory name (Brackets are formed by [ALPHA] [.] for "[" and [ALPHA] [EXP] for "]"). After the number memories have been expanded by 5, memories Z[1] through Z[5] are available.

2. Manual Calculations

Basic Calculations

- Arithmetic operations

Arithmetic operations are performed by pressing the keys in the same sequence as in the formula.

For negative values, press [SHIFT] [(-)] before entering the value.

Example	Key Operation	Display	Remark
23+4.5-53=	23 [+] 4.5 [-] 53 [EXE]	-25.5	
56x(-12)÷(-2.5)=	56 [x] [SHIFT] [(-)] 12 [÷] [SHIFT] [(-)] 2.5 [EXE]	268.8	
12369x7532x74103=	12369 [x] 7532 [x] 74103 [EXE]	6.903680613 [12]	Results greater than . 10^{10} (10billion) or less than 10^{-2} (0.01) are displayed in exponential form.
$(4.5x10^{75})x(-2.3x10^{-79})$=	4.5 [EXP] 75 [x] [SHIFT] [(-)] 2.3 [EXP] [SHIFT] [(-)] 79 [EXE]	-1.035 [-03]	NORM1
$(2+3)x10^2$=	[(] 2 [+] 3 [)] [x] 1 [EXP] 2 [EXE]	500.	The correct result cannot be derived by entering [(] 2 [+] 3 [)] [EXP] 2. Be sure to enter [x] 1 between the [)] and [EXP] in this example.
$(1x10^5)$÷7=	1 [EXP] 5 [÷] 7 [EXE]	14285.71429	Internal calculations are calculated with 12 digits for a mantissa, and the result is displayed rounded off to 10 digits. Internally, however, the mantissa of the result is 12 digits.
$(1x10^5)$÷7-14285=	1 [EXP] 5 [÷] 7 [-] 14285 [EXE]	0.7142857	

For mixed arithmetic operations, multiplication and division are given priority over addition and subtraction.

Example	Key Operation	Display	Remark
3+5x6=	3 [+] 5 [x] 6 [EXE]	33.	
7x8-4x5=	7 [x] 8 [-] 4 [x] 5 [EXE]	36.	
1+2-3x4÷5+6=	1 [+] 2 [-] 3 [x] 4 [÷] 5 [+] 6 [EXE]	6.6	

- **Parenthesis calculations**

Example	Key Operation	Display	Remark
100-(2+3)x4=	100 [-] [(] 2 [+] 3 [)] [x] 4 [EXE]	80.	
2+3x(4+5)=	2 [+] 3 [x] [(] 4 [+] 5 [EXE]	29.	Closed parentheses occurring immediately before operation of the key may be omitted, no matter how many are required.
(7-2)x(8x5)=	[(] 7 [-] 2 [)] x [(] 8 [+] 5 [EXE]	65.	A multiplication sign (x) occurring immediately before an open parenthesis can be omitted.
10-{2+7x(3+6)}=	10 [-] [(] 2 [+] 7 [x] [(] 3 [+] 6 [)] [)] [EXE]	-55.	*** Henceforth, abbreviated style will not be used in this manual.
(2x3+4)÷5=	[(] 2 [x] 3 [+] 4 [)] [÷] 5 [EXE]	2.	
(5x6+6x8)÷(15x4+12x3) =	[(] 5 [x] 6 [+] 6 [x] 8 [)] [÷] [(] 15 [x] 4 [÷] 12 [x] 3 [)] [EXE]	0.8125	
$(1.2x10^{19})$-$\{(2.5x10^{20})x3÷100\}$=	1.2 [EXP] 19 [-] [(] 2.5 [EXP] 20 [x] 3 [÷] 100 [)] [EXE]	18 4.5	
6÷(4x5)=	6 [÷] [(] 4 [x] 5 [)] [EXE]	0.3	This case is the same as 6 [÷] 4 [÷] 5 [EXE]

- **Memory calculations**

Memory contents are not erased when power is off. They are cleared by pressing [SHIFT] followed by [Mcl] and then [EXE].

Example	Key Operation	Display	Remark
	9.874 [→] [ALPHA] [A] [EXE]	9.874	The [→] key is used to input numeric values in memory. (Clearing a memory before input is not required, because the previous value in the memory will be replaced with the new value.)
9.874x7=	[ALPHA] [A] [x] 7 [EXE]	69.118	
9.874x12=	[ALPHA] [A] [x] 12 [EXE]	118.488	
9.874x26=	[ALPHA] [A] [x] 26 [EXE]	256.724	
9.874x29=	[ALPHA] [A] [x] 29 [EXE]	286.346	
23+9=	23 [+] 9 [→] [ALPHA] [B] [EXE]	32.	
53-6=	53 [-] 6 [EXE]	47.	
	[ALPHA] [B] [+] [Ans] [→] [ALPHA] [B] [EXE]	79.	
45x2=	45 [x] 2 [EXE]	90.	
	[ALPHA] [B] [-] [Ans] [→] [ALPHA] [B] [EXE]	-11.	
99+3=	99 [+] 3 [EXE]	33.	
	[ALPHA] [B] [+] [Ans] [→] [ALPHA] [B] [EXE]	22.	
12x(2.3+3.4)-5=	2.3 [+] 3.4 [→] [ALPHA] [G] [EXE]	5.7	
	12 [x] [ALPHA] [G] [-] 5 [EXE]	63.4	
30x(2.3+3.4+4.5) -15x4.5=	2.3 [+] 3.4 [→] [ALPHA] [G] [EXE]	5.7	Multiplication signs (x) immediately before memory names can be omitted.
	4.5 [→] [ALPHA] [H] [EXE]	4.5	
	30 [x] [(] [ALPHA] [G] [+] [ALPHA] [H] [)] [-] 15 [ALPHA] [H] [EXE]	238.5	

- **Specifying the number of decimal places, the number of significant digits, and the exponent display**

To specify the number of decimal places, press [MODE] followed by [7], a value indicating the number of places (0-9), and then [EXE].

To specify the number of significant digits, press [MODE] followed by [8], a value indicating the number of significant digits (0-9 to set from 1 to 10 digits), and then [EXE].

Press the [ENG] key or [SHIFT] followed by [ENG] will cause the exponent display for the number being displayed to change in multiples of 3.

The specified number of decimal places or number of significant digits will not be cancelled until another value or [MODE] [9] is specified using the sequence: [MODE], [9], [EXE]. Specified values are not cancelled even if power is switched off or another mode (besides [MODE] [9])is specified.

Even if the number of decimal places and number of significant digits are specified, internal calculations are performed in 12 digits for a mantissa, and the displayed value is stored in 10 digits. To convert these values to the specified number of decimal places and significant places and significant digits, press [SHIFT] followed by [Rnd] and then [EXE].

You cannot specify the display format (Fix, Sci) while the calculator is in the BASE-N mode. Such specifications can only be made if you first exit the BASE-N mode.

Example	Key Operation	Display	Remark
100÷6=	100 [÷] 6 [EXE]	16.66666667	
	[MODE] [7] 4 [EXE]	Fix 16.6667	Four decimal places specified
	[MODE] [9] [EXE]	16.66666667	Specification cancelled
	[MODE] [8] 5 [EXE]	Sci 01 1.6667	Five significant digits specified
	[MODE] [9] [EXE]	16.66666667	Specification cancelled
			*** Values are displayed rounded off to the place specified.
	[MODE] [7] 3 [EXE]	16.667	Three decimal places specified.
200÷7x14	200 [÷] 7 [EXE]	Fix 28.571	Continues calculation with 10 digits display.
	[x]	Fix 8.57142857 x _	
	14 [EXE]	Fix 400.000	
	200 [÷] 7 [EXE]	Fix 28.571	If the same calculation is performed with the specified number of digits.
	[SHIFT] [Rnd] [EXE]	Fix 28.571	Value stored internally cut off at specified decimal place.
	[x]	Fix 28.571x_	
	14 [EXE]	Fix 399.994	
	[MODE] [9] [EXE]	399.994	Specification cancelled
123m x 456 = ?m	123 [x] 456 [EXE] [ENG]	03 56.088	
= ?km	[SHIFT] [ENG]	06 0.056088	
78g x 0.96 = ?g	73 [x] 0.96 [EXE] [ENG]	00 74.88	
= ?kg	[SHIFT] [ENG]	03 0.07488	

● Special Functions

● Answer function

The answer function stores the result of the most recent calculation. Once a numeric value or numeric expression is entered and [EXE] is pressed, the result is stored by this function.

To recall the stored value, press the [Ans] key. When [Ans] is pressed, "Ans" appears on the display along with the answer function value. The value can be used in subsequent calculations.

NOTE: Since the "Ans" function works just like any other memory, it will be referred to as "Ans memory" throughout this manual.

Example: 123+456=579   789-579=210

Key Operation	Display
[AC] 123 [+] 456 [EXE]	579.
789 - [Ans]	789-Ans_
[EXE]	210.

Numeric values with 12 digits for a mantissa and 2 digits for an exponent can be stored in the Ans memory. The Ans memory is not cleared even if the power is turned off. Each time [EXE] is pressed, the value in the Ans memory is replaced with the value produced by the new calculation. When execution of a calculation results in an error, however, the Ans memory retains its current value.

When a value is stored to another memory using the [EXE] key, that value is not stored in the Ans memory.

Example: Perform calculation 78+56=134, then store the value 123 to memory A:

Key Operation	Display
78 [+] 56 [EXE]	134.
[Ans] [EXE] ... Checking the content of Ans	134.
123 [→] [ALPHA] [A] [EXE]	134.
[Ans] [EXE]	134.

The Ans memory can be used in the same manner as the other memories, thus making it possible to use it in calculation formulas. In multiplication operations, the [x] immediately before Ans can be omitted.

Example: 15 x 3 = 45 78 x 45 - 23 = 3487

[AC] 15 [x] 3 [EXE]	45.
78 [Ans] [-] 23 [EXE]	3487.

- **Continuous calculation function**

Even if calculations are concluded with the [EXE] key, the result obtained can be used for further calculations. Such calculations are performed with 10 digits mantissa of the displayed value.

Example 1: To calculate ÷3.14 after 3x4=12

[AC] 3 [x] 4 [EXE]	12.
(Continuing) [÷] 3.14	12.÷3.14
[EXE]	3.821656051.

Example 2: To calculate 1÷3x3

[AC] 3 [÷] 3 [x] 3 [EXE]	1.
1 [÷] 3 [EXE]	0.333333333
(Continuing) [x] 3 [EXE]	0.999999999

This function can be used with memory and Type A functions (x^2, x^{-1}, x!, ° ' ", °, r, g), +, −, x^y, and $^x\sqrt{\ }$

Example 1: To store the result of 12x45 in memory C:

12 [x] 45 [EXE]	540.
(Continuing) [→] [ALPHA] [C]	540.→C
[EXE]	540.

Example 2: To square the result of 78÷ 6=13

[AC] 78 [÷] 6 [EXE]	13.
(Continuing) [x^2]	$13.^2$_
[EXE]	169.

- **Replay function**

This function stores the latest formula executed. After execution is complete, pressing either the ◁ or ▷ key will display the formula.

Pressing ▷ will display the formula from the beginning, with the cursor located under the first character. Pressing ◁ will display the formula from the end, with the cursor located at the space following the last character. After this, use ▷ and ◁ to move the cursor to check the formula. You can edit numeric values or commands for subsequent execution.

NOTE: As with the number of input steps, the Replay function can accept input of up to 127 steps.

- **Error position display function**

When an ERROR message appears, press ▷ or ◁ to display the calculation with the cursor located at the step that caused the error. You can also clear an error by pressing [AC] and then reenter the value and formulas from the beginning.

- **Multi-statement function**

- The multi-statement function available in program calculations can also be used in manual calculations.
- With the multi-statement function, multiple statements are linked together with a colon ([:])separating them.
- Pressing the [EXE] key after a multi-statement is entered causes the entire chain of statements to be executed from left to right.
- Using "◢" ([ALPHA] [◢]) in place of a colon displays the calculation result up to the point that "◢" is encountered

Example: 6.9x123=848.7 123÷3.2=38.4375

123 [→] [ALPHA] [A] [:] 6.9 [x] [ALPHA] [A] [ALPHA] [◢] [ALPHA] [A] [÷] 3.2	6.9×A◢A÷3.2
[EXE]	Disp 848.7 Appears on display when "◢" is used.
[EXE]	38.4375

*** The final result of a multi-statement is always displayed, whether a "◢" symbol is input at the end of the last statement in the chain.

*** consecutive calculations contained in multi-statements cannot be performed

123x456 : x5 (Invalid)

Functional calculations

Angular measurement units

The unit of angular measurement (degrees, radians, grads) is set by pressing [MODE] followed by a value from 4 through 6 and then [EXE].

The numeric value from 4 through 6 specifies degrees, radians and grads respectively.

Once a unit of angular measurement is set, it remains in effect until a new unit is set. Settings are not cleared when power is off.

You cannot specify the unit of angular measurement (degrees, radians, grads) while the calculator is in the BASE-N mode. Such specifications can only be made if you first exit the BASE-N mode.

Example	Key Operation	Display	Remark
4.25 rad = ? degrees	[MODE] [4] [EXE] 4.25 [SHIFT] [MODE] [5] [EXE]	243.5070629	
1.23 grad = ? radians	[MODE] [5] [EXE] 1.23 [SHIFT] [MODE] [6] [EXE]	0.019320794	
7.89 degrees = ? grads	[MODE] [6] [EXE] 7.89 [SHIFT] [MODE] [4] [EXE]	8.766666667	
47.3° +82.5 rad= ? °	[MODE] [4] [EXE] 47.3 [+] 82.5 [SHIFT] [MODE] [5] [EXE]	4774.20181	
12.4° +8.3 rad- 1.8grad = ? °	12.4 [+] 8.3 [SHIFT] [MODE] [5] [-] 1.8 [SHIFT] [MODE] [6] [EXE]	486.33497	
24° 6' 31"+85.34 rad = ? rad	[MODE] [5] [EXE] 24 [° ' "] 6 [° ' "] 31 [° ' "] [SHIFT] [MODE] [4] [+] 85.34 [EXE]	85.76077464	
36.9° +41.2 rad = ? grad	[MODE] [6] [EXE] 36.9 [SHIFT] [MODE] [4] [+] 41.2 [SHIFT] [MODE] [5] [EXE]	2663.873462	

Trigonometric functions and inverse trigonometric functions

Be sure to set the unit of angular measurement before performing trigonometric function and inverse trigonometric function calculations.

The operations noted below cannot be performed in the BASE-N mode.

Example	Key Operation	Display	Remark
sin63° 52'41"=	[MODE] [4] [EXE] [sin] 63 [° ' "] 52 [° ' "] 41 [° ' "] [EXE]	D 0.897859012	
cos(π /3 rad)=	[MODE] [5] [EXE] [cos] [(] [SHIFT] [π] [÷] 3 [)] [EXE]	R 0.5	
tan(-35grad)=	[MODE] [6] [EXE] [tan] [SHIFT] [(-)] 35 [EXE]	G -0.612800788	
2.sin45° x cos65° =	[MODE] [4] [EXE] 2 [x] [sin] 45 [x] [cos] 65 [EXE]	D 0.597672477	
cot30° =1/tan30° =	1 [÷] [tan] 30 [EXE]	1.732050808	
sec(π /3 rad)= 1/ cos(π /3 rad)=	[MODE] [5] [EXE] 1 [÷] [cos] [(] [SHIFT] [π] [÷] 3 [)] [EXE]	R 2	
cosec30° = 1/sin30° =	[MODE] [4] [EXE] 1 [÷] [sin] 30 [EXE]	D 2	
$\sin^{-1}$ 0.5=	[SHIFT] [$\sin^{-1}$] 0.5 [EXE]	30	

$\cos^{-1}\sqrt{2}/2$= ? rad = π /4 rad	[MODE] [5] [EXE] [SHIFT] [$\cos^{-1}$] [(] [√] 2 [÷] 2 [)] [EXE] [÷] [SHIFT] [π] [EXE]	R 0.785398163 0.249999999	If the total number of digits for degrees / minutes / seconds exceeds 10, the high-order values (degrees and minutes) are given display priority, and any lower-order values are not displayed. However, the entire value is stored as a decimal value.
$\tan^{-1}$0.741= ?° = ? ° ' "	[MODE] [4] [EXE] [SHIFT] [$\tan^{-1}$] 0.741 [EXE] [SHIFT] [° ' "]	D 36.53844576 36° 32'18.4"	
2.5x($\sin^{-1}$0.8-$\cos^{-1}$0.9) = ?° ' "	2.5 [x] [(] [SHIFT] [$\sin^{-1}$] 0.8 [-] [SHIFT] [$\cos^{-1}$] 0.9 [)] [EXE] [SHIFT] [° ' "]	D 68° 13'13.52"	

- **Logarithmic and exponential functions**

The operations noted below cannot be performed in the BASE-N mode.

Example	Key Operation	Display	Remark
Log 1.23 ($\log_{10}1.23$)=	[log] 1.23 [EXE]	0.089905111	
In90($\log_e 90$)=	[ln] 90 [EXE]	4.49980967	
Log456÷In456=	[log] 456 [÷] [ln] 456 [EXE]	0.434294481	log/In ratio = constant M
$4^x=64$ x.log 4 =log 64 x=log 64 / log 4 =	[log] 64 [÷] [log] 4 [EXE]	3.	
$10^{1.23}$ =	[SHIFT] [10^x] 1.23 [EXE]	16.98243652	
$e^{4.5}$ =	[SHIFT] [e^x] 4.5 [EXE]	90.0171313	
$10^4 \times e^{-4}+1.2\times10^{2.3}$ =	[SHIFT] [10^x] 4 [x] [SHFIT] [e^x] [SHIFT] [(-)] 4 [+] 1.2 [x] [SHIFT] [10^x] 2.3 [EXE]	D 422.5878667	
$5.6^{2.3}$ =	5.6 [x^y] 2.3 [EXE]	52.58143837	
$\sqrt[7]{123}$ (=$123^{1/7}$) =	7 [$\sqrt[x]{\ }$] 123 [EXE]	1.988647795	
$(78-23)^{-12}$ =	[(] 78 [-] 23 [)] [x^y] [SHIFT] [(-)] 12 [EXE]	-21 1.305111829	
2+3x$\sqrt[3]{64}$-4=	2 [+] 3 [x] 3 [$\sqrt[x]{\ }$] 64 [-] 4 [EXE]	10.	x^y and $\sqrt[x]{\ }$ given calculation priority over x and +
$2\times3.4^{(5+6.7)}$ =	2 [x] 3.4 [x^y] [(] 5 [+] 6.7 [)] [EXE]	3306232.	

- **Hyperbolic functions and inverse hyperbolic functions**

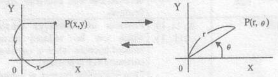
The operations noted below cannot be performed in the BASE-N mode.

Example	Key Operation	Display	Remark
sinh 3.6 =	[hyp] [sin] 3.6 [EXE]	18.28545536	
cosh 1.23 =	[hyp] [cos] 1.23 [EXE]	1.856761057	
tanh 2.5 =	[hyp] [tan] 2.5 [EXE]	0.986614298	
cosh1.5 –sinh1.5 = ? $=e^{-1.5}$	[hyp] [cos] 1.5 [-] [hyp] [sin] 1.5 [EXE] (continuing) [ln] [Ans] [EXE]	0.22313016 -1.5	coshx ± sinhx $=e^{\pm x}$
$\sinh^{-1}30$=	[hyp] [SHIFT] [$\sin^{-1}$] 30 [EXE]	4.094622224	
$\cosh^{-1}(20/15)$=	[hyp] [SHIFT] [$\cos^{-1}$] [(] 20 [÷] 15 [)] [EXE]	0.795365461	
x=($\tanh^{-1}0.88$)/4=	[hyp] [SHIFT] [$\tan^{-1}$] 0.88 [÷] 4 [EXE] [)]	0.343941914	–
$\sinh^{-1}2\times\cosh^{-1}1.5$=	[hyp] [SHIFT] [$\sin^{-1}$] 2 [x] [hyp] [SHIFT] [$\cos^{-1}$] 1.5 [EXE]	1.389388923	
$\sinh^{-1}(2/3)+\tanh^{-1}(4/5)$ =	[hyp] [SHIFT] [$\sin^{-1}$] [(] 2 [÷] 3 [)] [+] [hyp] [SHIFT] [$\tan^{-1}$] [(] 4 [÷] 5 [)] [EXE]	1.723757406	

- **Coordinate transformation**

Your calculator lets you convert between rectangular coordinates and polar coordinates.

Rectangular coordinates Polar coordinates



Calculation results are stored in variable memory I and variable memory J. Contents of variable memory I are displayed first. To display contents of memory J, press [ALPHA] [J] [EXE].

	I	J
Pol	r	θ
Rec	x	y

With polar coordinates, θ can be calculated within a range of $-180^\circ < \theta \leqq 180^\circ$. The calculation range is the same for radians and grads.

The operations noted below cannot be performed in the BASE-N mode.

Example	Key Operation	Display	Remark
x=14, y=20.7 r= ?° θ= ?°	[MODE] [4] [EXE] [SHIFT] [Pol(] 14 [SHIFT] [,] 20.7 [)] [EXE] (continuing) [ALPHA] [J] [EXE] [SHIFT] [° ' "]	D 24.98979792 55.92839019 55° 55'42.2"	r θ
x=7.5 y=-10 r= ?rad θ= ?rad	[MODE] [5] [EXE] [SHIFT] [Pol(] 7.5 [SHIFT] [,] [SHIFT] [(-)] 10 [)] [EXE] (continuing) [ALPHA] [J] [EXE]	r 12.5 -0.927295218	r θ
r=25 θ=56° x= ? y= ?	[MODE] [4] [EXE] [SHIFT] [Rec(] 25 [SHIFT] [,] 56 [)] [EXE] (continuing) [ALPHA] [J] [EXE]	D 13.97982259 20.72593931	x y
r=4.5 θ=2/3π x= ? y= ?	[MODE] [5] [EXE] [SHIFT] [Rec(] 4.5 [SHIFT] [,] [(] 2 [÷] 3 [x] [SHIFT] [π] [)] [)] [EXE] (continuing) [ALPHA] [J] [EXE]	r -2.25 3.897114317	x y

- **Other functions(√, X^2, X^{-1}, X!, $^3\sqrt{}$, Ran#, Abs, Int, Frac)**

The operations noted below cannot be performed in the BASE-N mode.

Example	Key Operation	Display	Remark
$\sqrt{2}+\sqrt{5}=$	[√] 2 [+] [ ] 5 [EXE]	3.65028154	
$2^2+3^2+4+^25+^2=$	2 [x²] [+] 3 [x²] [+] 4 [x²] [+] 5 [x²] [EXE]	54.	
1/(1/3-1/4) =	[(] 3 [x⁻¹] [-] 4 [x⁻¹] [)] [x⁻¹] [EXE]	12.	
8!(=1x2x3x···x8)=	8 [SHIFT] [x!] [EXE]	40320.	
$\sqrt[3]{36x36x49}=$	[SHIFT] [³√] [(] 36 [x] 42 [x] 49 [)] [EXE]	42.	
Random number generation (pseudorandom number from 0.000 to 0.999)	[SHIFT] [Ran#] [EXE]	(Ex.) 0.792	

$\sqrt{13^2-5^2}+\sqrt{3^2+4^2}$ =	[√] [(] 13 [x²] [-] 5 [x²] [)] [+] [√] [(] 3 [x²] [+] 4 [x²] [)] [EXE]	17.	
$\sqrt{1-\sin^2 40°}$ =? = cos40° (Proof of cos = √ 1-sin²)	[MODE] [4] [EXE] [√] [(] 1 [-] [(] [SIN] 40 [)] [x²] [)] [EXE] (continuing) [SHIFT] [cos⁻¹] [Ans] [EXE]	D 0.766044443 40.	
1/2!+1/4!+1/6!+1/8! =	2 [SHIFT] [x!] [x⁻¹] [+] 4 [SHIFT] [x!] [x⁻¹] [+] 6 [SHIFT] [x!] [x⁻¹] [+] 8 [SHIFT] [x!] [x⁻¹] [EXE]	0.543080357	
| log3/4| =	[SHIFT] [Abs] [log] [(] 3 [÷] 4 [)] [EXE]	0.124938736	
7800/96= ?integer	[SHIFT] [Int] [(] 7800 [÷] 96 [)] [EXE]	81.	
7800/96= ?fraction	[SHIFT] [Frac] [(] 7800 [÷] 96 [)] [EXE]	0.25	
2512549139÷2141 = ? aliquot part	2512549139 [÷] 2141 [EXE] [SHIFT] [Frac] [(] 2512549139 [÷] [EXE] 2141 [)] [EXE]	1173540. 0.99953	

- **Fractions**

Fractions are input and displayed in the following order: integer, numerator, denominator.

Example	Key Operation	Display	Remark
2/5 + 3⌟1/4 = ?fractions = ?decimal	2 [a b/c] 5 [+] 3 [a b/c] 1 [a b/c] 4 [EXE] (Conversion to decimal) [a b/c]	3⌟13⌟20. 3.65	Fractions can be converted to decimals and then converted back to fractions.
3⌟456/78 =8⌟11/13 (Reduced)	3 [a b/c] 456 [a b/c] 78 [ ]EXE (continuing) [SHIFT] [d/c]	8⌟11⌟13. 115⌟13.	Fractions and improper fractions which can be reduced become reduced fractions when a calculation command key is pressed. Press [SHIFT] [d/c] to convert to improper fraction.
1/2578+1/4572=	1 [a b/c] 2578 [+] 1 [a b/c] 4572 [EXE]	-04 6.066202547 (NORM1 mode)	When the total number of characters, including integer, numerator, denominator and delimiter mark exceeds 10, the input fraction is displayed in decimal format.
1/2 x 0.5=	1 [a b/c] 2 [×] [ ] 5 [EXE]	0.25	Calculations containing both fractions and decimals are calculated in decimal form.
1/3 x(-4/5)-5/6 =	1 [a b/c] 3 [×] [SHIFT] [(-)] 4 [a b/c] 5 [-] 5 [a b/c] 6 [EXE]	-1⌟1⌟10.	
1/2x1/3+1/4x1/5 =	1 [a b/c] 2 [×] 1 [a b/c] 3 [+] 1 [a b/c] 4 [×] 1 [a b/c] 5 [EXE]	13⌟60.	
1/2/3=1/6	[(] 1 [a b/c] 2 [)] [a b/c] 3 [EXE]	1⌟6.	
1/(1/3+1/4)=	1 [a b/c] [(] 1 [a b/c] 3 [+] 1 [a b/c] 4 [)] [EXE]	-1⌟5⌟7.	Fractional calculations can be performed by using parentheses in the numerator or denominator.

Binary, Octal, Decimal, Hexadecimal calculations

Binary, octal, decimal, and hexadecimal calculations, conversions, and logical operations are performed in the BASE-N mode ([MODE] [-]).

The number system (2,8,10,16) is set by respectively pressing [SIN], [OCT], [DEC] or [HEX] followed by [SXE]. A corresponding symbol – "b", "o", "d" or "h" appears on the display.

Number systems are specified for specific values by pressing [SHIFT], then the number system designator (b, o, d, or h), immediately followed by the value.

General function calculations cannot be performed in the BASE-N mode.

Only integers can be handled in the BASE-N mode. If a calculation produces a result that includes a decimal value, the decimal portion is cut off.

Octal, decimal, and hexadecimal calculations can be handled up to 32 bit, while binary can be handled up to 12 bits.

Number system	Number of digits displayed
Binary	Up to 12 digits
Octal	Up to 11 digits
Decimal	Up to 10 digits
Hexadecimal	Up to 8 digits

The total range of numbers handled in this mode is 0, 1, 2 , 3, 4, 5, 6, 7, 8, 9 , A, B, C, D, E, F. If values not valid for the particular number system are used, attach the corresponding designator (b, o, d or h), or an error message will appear.

Number system	Valid values
Binary	0, 1
Octal	0, 1, 2, 3, 4, 5, 6, 7
Desomal	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ,A, B, C, D, E, F

Negative numbers in binary, octal, and hexadecimal are expressed as two's complements.

To distinguish the A, B, C, D, E and F used in the hexadecimal system from standard letters, they appear as shown in the chart below.

Key	Display
[/A] (= [x^{-1}])	/A
[IB] (= [ ])	IB
[IC] (= [hyp])	IC
[ID] (= [sin])	ID
[IE] (= [cos])	IE
[IF] (= [tan])	IF

Calculation range (in BASE-N mode)

Binary	Positive	: 011111111 ≥ x ≥ 0
	Negative	: 111111111 ≥ x ≥ 1000000000
Octal	Positive	: 1777777777 ≥ x ≥ 0
	Negative	: 3777777777 ≥ x ≥ 2000000000
Decimal		2147483647 ≥ x ≥ -2147483648
Hexadecimal	Positive	: 7FFFFFFF ≥ x ≥ 0
	Negative	: FFFFFFFF ≥ x ≥ 80000000

You cannot specify the unit of angular measurement (degress, radians, grads) or the display format (Fix, Sci) while the calculator is in the BASE-N mode. Such specifications can only be made if you first exit the BASE-N mode.

Binary, octal, decimal, hexadecimal conversions

Example	Key Operation	Display	Remark
$2A_{16}$ = ?decimal 274_8= ?decimal	[MODE] [-] [Dec] [EXE] [SHIFT] [h] 2A [EXE] [SHIFT] [o] 274 [EXE]	BASE-N d d 42 d 188	
123_{10}= ? hexadecimal 1010_2= ? hexadecimal	[Hex] [EXE] [SHIFT] [d] 123 [EXE] [SHIFT] [b] 1010 [EXE]	BASE-N h h 7B h A	
15_{16}= ?octal 1100_2= ?octal	[Oct] [EXE] [SHIFT] [h] 15 [EXE] [SHIFT] [b] 1100 [EXE]	BASE-N o o 25 o 14	
36_{10}= ?binary $1F7_{16}$= ?binary	[Bin] [EXE] [SHIFT] [d] 36 [EXE] [SHIFT] [h] 1F7 [EXE]	BASE-N b b 100100 b 111110111	

- **Negative expressions**

Example	Key Operation	Display	Remark
Negation of 110010_2	[MODE] [-] [Bin] [EXE] [Neg] 110010 [EXE]	BASE-N b b 1111001110	
Negation of 72_8	[Oct] [EXE] [Neg] 72 [EXE]	BASE-N o o 7777777706	
Negation of $3A_{16}$	[Hex] [EXE] [Neg] 3A [EXE]	BASE-N h h FFFFFFC6	

- **Basic arithmetic operations using binary, octal, decimal and hexadecimal values**

Example	Key Operation	Display	Remark
$10111_2+11010_2$ = ?Binary	[MODE] [-] [Bin] [EXE] 10111 [+] 11010 [EXE]	BASE-N b b 110001	
$B47_{16}-DF_{16}$ = ?hexadecimal	[Hex] [EXE] B47 [-] DF [EXE]	BASE-N h h A68	
$123_8 \times ABC_{16}$ =? hexadecimal =?decimal	[SHIFT] [o] 123 [×] ABC [EXE] [Dec] [EXE]	BASE-N 0 37AF4 d 228084	
$1F2D_{16} \times 100_{10}$ =? hexadecimal	[SHIFT] [h] 1F2D [-] 100 [EXE] [Hex] [EXE]	BASE-N d 7881 h 1EC9	
$7654_8 \div 12_{10}$ = ?decimal = ?oct	[Dec] [EXE] [SHIFT] [o] 7654 [÷] 12 [EXE] [Oct] [EXE]	BASE-N d d 334 o 516	
$1234_{10}+1EF_{16} \div 24_8$ = ?decimal = ?decimal	[SHIFT] [d] 1234 [+] [SHIFT] [h] 1EF [÷] 24 [EXE] [Dec] [EXE]	BASE-N o 2352 d 1258	

- **Logical operations**

Logical operations are performed through logical products (and), logical sums (or), negation (Not), exclusive logic sums (xor), and negation of exclusive logical sums (xnor).

Example	Key Operation	Display	Remark
19_{16}AND $1A_{16}$ = ?hexadecimal	[MODE] [-] [Hex] [EXE] 19 [and] 1A [EXE]	BASE-N h h 18	
1110_2AND 36_8 = ?binary	[Bin] [EXE] 1110 [and] [SHIFT] [o] 36 [EXE]	BASE-N b b 1110	
23_8 OR 61_8 = ?oct	[Oct] [EXE] 23 [or] 61 [EXE]	BASE-N o o 53	
120_{16} OR 1101_2 = ?hexadecimal	[Hex] [EXE] 120 [or] [SHIFT] [b] 1101 [EXE]	BASE-N h h 12D	
1010_2 AND (A_{16} OR 7_{16}) = ?binary	[Bin] [EXE] 1010 [and] [(] [SHIFT] [h] A [or] [SHIFT] [h] 7 [)] [EXE]	BASE-N b b 1010	
5_{16} XOR3_{16} = ?hexadecimal	[Hex] [EXE] 5 [SHIFT] [xor] 3 [EXE]	BASE-N h h 6	
$2A_{16}$ XNOR $5D_{16}$ = ?hexadecimal	[Hex] [EXE] 2A [SHIFT] [xnor] 5D [EXE]	BASE-N h h FFFFFF88	
Negation of 1234_8	[Oct] [EXE] [Not] 1234 [EXE]	BASE-N o o 7777776543	
Negatuin of $2FFFFD_{16}$	[Hex] [EXE] [Not] 2FFFED [EXE]	BASE-N h h FFD00012	

Statistical calculations

Standard deviation

- Standard deviation calculations are performed in the SD mode ([MODE] [x]). "SD" appears on the display.
- Before beginning calculations, the statistical memories are cleared by pressing [SHIFT] followed by [Scl] and then [EXE]
- Individual data are input using [DT]
- Multiple data of the same value can be input either by repeatedly pressing [DT] or by entering the data, pressing [SHIFT], followed by [;] (that presents the number of times the data is repeated) and then [DT].

$$\sigma_n = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}} = \sqrt{\frac{\sum x^2 \ (\sum x)^2 n}{n}} =$$

Using all data from a finite population to detetmine the standard deviation for the population

$$\sigma_{n-1} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{\sum x^2 \ (\sum x)^2 n}{n-1}} =$$

Using sample data from a population to determine the standard deviation for the population

Mean

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{\sum x}{n}$$

NOTE: The values for n, $\sum x$, and $\sum x^2$ are stored in memories W, V, and U respectively and can be obtained by pressing [MODE] [ALPHA] followed by the memory name and then [EXE] (e.g. [MODE] [ALPHA] [W] [EXE]).

Example	Key Operation	Display	Remark
Data 55, 54, 51, 55, 53, 53, 54, 52	[MODE] [x] [SHIFT] [Scl] [EXE] (Clears memory) 55 [DT] 54 [DT] 51 [DT] 55 [DT] 55 [DT] [DT] 54 [DT] 52 [DT]	52.	*** You can press the function keys to obtain results in any sequence
	[SHIFT] [$x\sigma_n$] [EXE]	1.452368755	Standard deviation on
	[SHIFT] [$x\sigma_{n-1}$] [EXE]	1.552647509	Standard deviation on-1
	[SHIFT] [$\bar{x}$] [EXE]	53.875	Mean x
	[MODE] [ALPHA] [w] [EXE]	8.	Number of data n
	[MODE] [ALPHA] [v] [EXE]	431.	Sum total $\sum x$
	[MODE] [ALPHA] [u] [EXE]	23237.	Sum of aquares $\sum x^2$
To calculate the deviation of the unbiased variance, the difference between each datum, and mean of the above data	(continuing) [SHIFT] [$x\sigma_{n-1}$] [x^2] [EXE]	2.410714286	
	55 [-] [SHIFT] [$\bar{x}$] [EXE]	1.125	
	54 [-] [SHIFT] [$\bar{x}$] [EXE]	0.125	
	51 [-] [SHIFT] [$\bar{x}$] [EXE]	-2.875	
To calculate x and σ_{n-1} for the following data. Class no. Value Frequency 1 110 10 2 130 31 3 150 24 4 170 2 5 190 3	[SHIFT] [Scl] [EXE]		
	110 [SHIFT] [;] 10 [DT]	110.	
	130 [SHIFT] [;] 31 [DT]	130.	
	150 [SHIFT] [;] 24 [DT]	150.	
	170 [DT] [DT]	170.	
	190 [DT] [DT] [DT]	190.	
	[MODE] [ALPHA] [W] [EXE]	94.	
	[SHIFT] [$\bar{x}$] [EXE]	21.15390155	
	[SHIFT] [$x\sigma_{n-1}$] [EXE]	21.26732811	
		145.9574468	

*** Erroneous data clearing / correction I (correct data operation: 51 [DT])

1) If 50 [DT] is entered, enter correct data after pressing [CL]. CL
2) If 49 [DT] was input a number of entries previously, enter correct data sfter pressing 49 [CL].

*** Erroneous data clearing / correction II (correct data operation: 130 [SHIFT] [;] 31 [DT])

1) If 120 [SHIFT] [;] is entered, enter correct data after pressing [AC].
2) If 120 [SHIFT] [;] 31 is entered, enter correct data after pressing [AC].
3) If 120 [SHIFT] [;] 30 [DT] is entered, enter correct data after pressing [CL].
4) If 120 [SHIFT] [;] 30 [DT] was entered previously, enter correct data after pressing 120 [SHIFT] [;] 30 [CL].

- **Regression calculation**

Regression calculations are performed in the LR mode ([MODE] [÷]). "LR" appears on the display.

Before beginning calculations, the tabulation memories are cleared by pressing [SHIFT] followed by [Scl] and then [EXE].

Individual data are entered as x data [SHIFT] [,] y data [DT].

Multiple data of the same value can be entered by repeatedly pressing [DT]. This operation can also be performed by entering x data [SHIFT] [,] y data [SHIFT] [;] following by a value representing the number of time the data is repeated, and then [DT].

If only x data is repeated (x data having the samme calue) enter [SHIFT] [,] y data [DT] or [SHIFT] [,] y data [DT] or [SHIFT] [,] y data [SHIFT] [;] followed by a value repressenting the number of times the data repeated, and then [DT].

If only y data is repeated (y data having the same value), enter x data [DT] or x data [SHIFT] [;] followed by a value representing the total number of times the data is repeated, and then [DT].

Regression

The followings are the formulas used to calculate constant term A and regression coefficient B for the regression formula y= A+Bx.

Constant term of regression formula

$$A = \frac{\Sigma y - B \ \Sigma x}{n}$$

Regression coefficient of regression formula

$$B = \frac{n \cdot \Sigma xy - \Sigma x \ \Sigma y}{n \ \ \Sigma x^2 - (\Sigma x)^2}$$

Estimated value $\hat{x}$, and $\hat{y}$ based on the regession formula can be calculated using the following formulas:

$$\hat{y} = A + Bx \qquad \hat{x} = \frac{Y - A}{B}$$

(To obtain the estimated value $\hat{y}$, [SHIFT] [$\hat{y}$] is used, and to obtain estimated value $\hat{x}$, [SHIFT] [$\hat{x}$] is used.)

The correlation coefficient r for input data can be calculated using the following formula:

$$r = \frac{n \ \ \Sigma xy - \Sigma x \ \ \Sigma y}{\sqrt{\{n \ \ \Sigma x^2 - (\Sigma x)^2\} \ \{n \ \ \Sigma y^2 - (\Sigma y)^2\}}}$$

NOTE: The values for n, Σx, Σx^2, Σxy, Σy, and Σx^2 are stored in memories W, V, U, R, Q and P respectively, and can be obtained by pressing [MODE] [ALPHA] followed by the memory name and then [EXE] (i.e. [MODE] [ALPHA] [W] [EXE]).

- **Linear regression**

Example	Key Operation	Display	Remark
Relationship between temperature and the length of steel bar	[MODE] [+]		
	[SHIFT] [Scl] [EXE]		Clears memory
Temperature Length			
10°C 1003mm	10 [SHIFT] [,] 1003 [DT]	10.	
15°C 1005mm	15 [SHIFT] [,] 1005 [DT]	15.	
20°C 1010mm	20 [SHIFT] [,] 1010 [DT]	20.	
25°C 1010mm	25 [SHIFT] [,] 1011 [DT]	25.	
30°C 1014mm	30 [SHIFT] [,] 1014 [DT]	30.	
	[SHIFT] [A] [EXE]	997.4	Constant term A
The data in the above can be used to obtain the terms of the regression formula and the correlation coefficient. Based on the regression formula, the estimated length of the steel bar at 18 °C an the temperature when the bar is 1000mm long can be calculated. The critical coefficient (r^2) and covariance	[SHIFT] [B] [EXE]	0.56	Regression coefficient B
	[SHIFT] [r] [EXE]	0.982607368	Correlation coefficient r
	18 [SHIFT] [$\hat{y}$] [EXE]	1007.48	Length at 18°C
	1000 [SHIFT] [$\hat{x}$] [EXE]	4.642857143	Temperature at 1000mm
	[SHIFT] [r] [x^2] [EXE]	0.965517241	
$\left(\frac{\Sigma xy - n \ \bar{x} \ \bar{y}}{n-1} \right)$ Can also be calculated.	[(] [MODE] [ALPHA] [R] [-] [MODE] [ALPHA] [W] [x] [SHIFT] [$\bar{x}$] [x] [SHIFT] [$\bar{y}$] [)] [÷] [(] [MODE] [ALPHA] [W] [-] 1 [)] [EXE]	35	Covariance

*** Erroneous data clearing / correction (correct data operation: 10 [SHIFT] [,] 1003 [DT])

1) If 11 [SHIFT] [,] 1003 is entered, enter correct data after pressing [AC].
2) If 11 [SHIFT] [,] 1003 [DT] is entered, enter correct data after pressing [CL].
3) If 11 [SHIFT] [,] 1003 [DT] is entered previously, enter correct data after pressing 11 [SHIFT] [,] 1003 [CL]

- **Logarithmic regression**

The regression formula is y=A+B. lnx Enter the x data as the logarithm (ln) of x, and the y data inputs the same as that for linear regression.

Estimated values $\hat{x}$, and $\hat{y}$ based on the regression formula can be calculated using the following formulas:

$$y = A + B \quad \ln x \qquad x = \exp\left[\frac{y-A}{B}\right]$$

The same operation as with linear regression can be used to obtain the regression coefficient and for making corrections. To obtain the estimated calue $\hat{y}$, [ln]x [SHIFT] [$\hat{y}$] [EXE] is used, and to obtain estimated value x, y [SHIFT] [$\hat{x}$] [EXE] [SHIFT] [e^x] [Ans] [EXE] is used.

Furthermore, Σx, Σx^2 and Σxy are obtained by Σinx, $\Sigma(\ln x)^2$, and $\Sigma \ln x.y$ respectively.

Example	Key Operation	Display	Remark
	[MODE] [÷]		
xi yi	[SHIFT] [Scl] [EXE]		Clear memory
29 1.6	([ln] 29) [SHIFT] [,] 1.6 [DT]	3.36729583	
50 23.5	([ln] 50) [SHIFT] [,] 23.5 [DT]	3.912023005	
74 38.0	([ln] 74) [SHIFT] [,] 38.0 [DT]	4.304065093	
103 46.4	([ln] 103) [SHIFT] [,] 46.4 [DT]	4.634728988	
118 48.9	([ln] 118) [SHIFT] [,] 48.9 [DT]	4.770684624	
The data in the above can be used to obtain the terms	[SHIFT] [A] [EXE]	-111.1283976	Constant term A
of the regression formula and the correlation	[SHIFT] [B] [EXE]	34.0201475	Regression coefficient B
coefficient. Based on the regression formula,	[SHIFT] [r] [EXE]	0.994013946	Correlation coefficient r
estimated value $\hat{y}$ can be obtained for xi=80, and	([ln] 80) [SHIFT] [$\hat{y}$] [EXE]	37.94879482	$\hat{y}$ when xi=80
estimated value $\hat{x}$ can be obtained for yi=73.	73 [SHIFT] [$\hat{x}$] [EXE] [SHIFT] [e^x] [Ans] [EXE]	224.154133	$\hat{x}$ when yi=73

- **Exponential regression**

The regression formula is $y=A.e^{B.x}$ (lny=inA+Bx). Enter the y date as the logarithm of y(in), and the x data the same as that for linear regression.

Estimated values $\hat{x}$, and $\hat{y}$ based in the regression formula can be calculated using the following formulas:

$$\hat{y} = A + e^{Bx} \qquad \hat{x} = \frac{\ln y - \ln A}{B}$$

Correction is performed the same as in linear regression. Constant term A is obtained by [SHIFT] [e^x] [SHIFT] [A] [EXE] estimated value y is obtained by x [SHIFT] [$\hat{y}$] [EXE] [SHIFT] [e^x] [Ans] [EXE], and estimated value x is obtained by ([ln] y) [SHIFT] [$\hat{x}$] [EXE]. Σy, Σy^2 and Σxy are obtained by Σlny, $\Sigma(\ln y)^2$ and $\Sigma x.\ln y$ respectively.

Example	Key Operation	Display	Remark
	[MODE] [+]		
xi yi	[SHIFT] [Scl] [EXE]		Clear memory
6.9 21.4	6.9 [SHIFT] [,] ([ln] 21.4) [DT]	6.9	
12.9 15.7	12.9 [SHIFT] [,] ([ln] 15.7) [DT]	12.9	
19.8 12.1	19.8 [SHIFT] [,] ([ln] 12.1) [DT]	19.8	
26.7 8.5	26.7 [SHIFT] [,] ([ln] 8.5) [DT]	26.7	
35.1 5.2	35.1 [SHIFT] [,] ([ln] 5.2) [DT]	35.1	
The data in the above can be used to obtain the	[SHIFT] [e^x] [SHIFT] [A] [EXE]	30.49758743	Constant term A
terms of the regression formula and the	[SHIFT] [B] [EXE]	-0.049203708	Regression coefficient B
correlation coefficient. Based on the regression	[SHIFT] [r] [EXE]	-0.997247351	Correlation coefficient r
formula, estimated value $\hat{y}$ can be obtained for xi=16,	16 [SHIFT] [$\hat{y}$] [EXE] [SHIFT] [e^x] [Ans] [EXE]	13.87915739	y when xi=16
and estimated value $\hat{x}$ can be obtained for yi=20.	([ln] 20) [SHIFT] [$\hat{x}$] [EXE]	8.574868047	x when yi=20

- **Power regression**
- The regression formula is y=A.x^B (lny=lnA+Blnx). Enter both data x and y as logarithms (ln).
- Estimated values $\hat{x}$, and $\hat{y}$ vased on the regression formula can be calculated using the following formulas:

$$\hat{y} = A \quad e^{Bx} \qquad \hat{x} = \frac{\ln y - \ln A}{B}$$

- Correction is performed the same as in linear regression. Constant term A is obtained by [SHIFT] [e^x] [SHIFT] [A] [EXE], estimated value $\hat{y}$ is obtained by([ln] x) [SHIFT] [$\hat{y}$] [EXE] [SHIFT] [e^x] [Ans] [EXE], and estimated calue x is obtained by [ln] y [SHIFT] [$\hat{x}$] [EXE] [e^x] [Ans] [EXE]. Σx, Σx^2 Σy, Σy^2 and Σxy are obtained by $\Sigma \ln x$, $\Sigma(\ln x)^2$, $\Sigma \ln y$, $\Sigma(\ln y)^2$, and $\Sigma \ln x . \ln y$ respectively.

Example	Key Operation	Display	Remark
	[MODE] [+]		
xi yi	[SHIFT] [Scl] [EXE]		Clear memory
28 2410	([ln] 28) [SHIFT] [,] ([ln] 2410) [DT]		
30 3033	([ln] 30) [SHIFT] [,] ([ln] 3033) [DT]	3.401197382	
33 3895	([ln] 33) [SHIFT] [,] ([ln] 3895) [DT]	3.496507561	
35 4491	([ln] 35) [SHIFT] [,] ([ln] 4491) [DT]	3.555348061	
38 5717	([ln] 38) [SHIFT] [,] ([ln] 5717) [DT]	3.63758616	
The data in the above can be used to obtain the terms of the regression formula and the correlation coefficient. Based on the regression formula, estimated value $\hat{y}$ can be obtained for xi=40, and estimated value $\hat{x}$ can be obtained for yi=1000.	[SHIFT] [e^x] [SHIFT] [A] [EXE]	0.197752493	Constant term A
	[SHIFT] [B] [EXE]	2.823980627	Regression coefficient B
	[SHIFT] [r] [EXE]	0.99974582	Correlation coefficient r
	([ln] 40) [SHIFT] [$\hat{y}$] [EXE] [SHIFT] [e^x] [Ans] [EXE]	49.21091222 6611.627038	y when x i=40
	([ln] 1000) [SHIFT] [$\hat{x}$] [EXE] [SHIFT] [e^x] [Ans] [EXE]	20.49182029	x when yi=1000

3. Graphs

Built-in function graphs

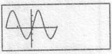
The COMP mode of the RUN mode should be used when graphing functions. Some graphs can be produced in the SD and LR modes, but certain graphs cannot be produced in these modes. The BASE-N graphs make it possible to produce the graphs of basic functions.

sin	cos	tan	$\sin^{-1}$	$\cos^{-1}$	$\tan^{-1}$
sinh	cosh	tanh	$\sinh^{-1}$	$\cos^{-1}$	$\tanh^{-1}$
$\sqrt{\ }$	x^2	log	ln	10^x	e^x
x^{-1}	$\sqrt[3]{\ }$				

Any time a built-in graph is executed, the ranges are automatically set to their optimum values, and any graph previously on the display is cleared.

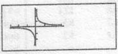
Example 1: Sine curve

[MODE] [+]
[Graph] [sin] [EXE]



Example 2: y = 1/x graph

[Graph] [x^{-1}] [EXE]



- **Overlaying built-in function graphs**

Two or more different built-in function graphs can be drawn together on the same display. Since the range for the first graph is automatically set, all subsequent graphs on the same display are produced according to the range of the first graph.

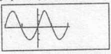
The first graph is produced by using the previously mentioned operation (Graph [function key] [EXE]).

Subsequent graphs are produced using the variable x in the operation Graph [function key] [ALPHA] [X] [EXE]. By inputting [ALPHA] [X] after the function key, the range is unchanged and the next graph is produced without clearing the existing display.

Example: Overlay the graph for y= cosx on the graph for y=sinx

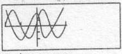
First, draw the graph for y=sinx.

[Graph] [sin] [EXE]



Next, draw the graph for y=cosx without changing the existing range

[Graph] [cos] [ALPHA] [X] [EXE]



*** Built-in function graphs cannot be used in multi-statements and cannot be written into program.

User Generated Graphs

Built-in function graphs can also be used in combination with each other. Graphing a formula such as ...

Ymax --- maximum value of the y-axis

Yscl --- scale of the y-axis

Specifying range parameters

Whenever you press the [Range] key (except in the BASE-N mode), the range parameter setting screen appears on the display. Enter the value you want to specify for the displayed parameter and then press [EXE]

Example: Change the range parameters on the left to those on the right.

Xmin:	0 → -5	Ymin:	-10 → -5
Xmax:	5 → 5	Ymax:	10 → 15
Xscl :	4 → 2	Yscl :	4 → 4

[Range]

Xmin?

● User Generated Graphs

Built-in function graphs can also be used in combination with each other. Graphing a formula such as $y=2x^2+3x^2-5$ makes it possible to visually represent the solution.

Unlike built-in functions, the ranges of user generated graphs are not set automatically, so graphs produced outside of the display range do not appear on the display.

3.2.1 Range parameters

After pressing the [Range] key, you can look up and specifty the range parameters for the x- and y- coordinates. Range parameters consist of maximum and minimum values for each axis, as well as their scales (distance between hash marks).

Before drawing a graph, you should first specify range parameters to set the size of the graph.

Range parameter types

Range parameters consist of the following:

Xmin --- minimum value of the x-axis

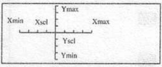
Xmax --- maximum value of the x-axis

Xscl --- scale of the x-axis

Ymin --- minimum value of the y-axis

Ymax --- maximum value of the y-axis

Yscl --- scale of the y-axis



Specifying range parameters

Whenever you press the [Range] key (except in the BASE-N mode), the range parameter setting screen appears on the display. Enter the value you want to specify for the displayed parameter and then press [EXE]

Example: Change the range parameters on the left to those on the right.

Xmin:	0 → -5	Ymin:	-10 → -5
Xmax:	5 → 5	Ymax:	10 → 15
Xscl :	4 → 2	Yscl :	4 → 4

[Range]

Xmin?
0.

● Specifies -5 for Xmin
[SHIFT] [(-)] [5]

Xmin?
-5_

[EXE]

Xmax?
5.

Xmax does not change, so simply press [EXE].
[EXE]

Xscl?
4.

Specifies 2 for Xscl
[2] [EXE]

Ymin?
-10.

Specifies -5 for Ymin
[SHIFT] [(-)] [5] [EXE]

Xmax?
10.

Specifies 15 for Ymax
[1] [5] [EXE]

Yscl?
4.

Yscl does not change, so simply press [EXE].
[EXE]

Xmin?
-5

Press [Range] to return to the display that was shown before entering the range display.

Checking range parameters

Press the [Range] key and the range parameter setting screen appears on the display. Press [EXE] to scroll through the range parameter settings without changing them.

[EXE]	Xmin? -5.
[EXE]	Xmax? -5.
[EXE]	Xscl? -2.
[EXE]	Xmin? -5.
[EXE]	ymax ? -5.
[EXE]	Xscl? -4.

Press [Range] to return to the display that was shown before entering the range display.

You can input range parameters as expressions (such as 2π) and these expressions are automatically converted to the values.

*** The input range for graph ranges is $-9.999999999_E{}^{+97}$ through $9.999999999_E{}^{+99}$.

*** If you enter a value that is outside the allowable range or if you try to perform some other illegal operation, an error message appears on the display. When this happens, press [◁] or [▷] to display the place in the calculation that caused the error (Replay function) and make the necessary corrections.

*** Inputting 0 for Xcal or Yscl does not set any scale.

*** Inputting a maximum value that is less than the minimum value will reverse the respective axis.

*** If the maximum and minimum values of an axis are equal, an error (Ma ERROR) will be generated when an attempt is made to produce a graph.

*** When a range setting is used that does not allow display of the axes, the scale for the y-axis is indicated on either the left or right edge of the display, while that for the x-axis is indicated on either thetop or bottom edge. (In both cases, the location of the scale is the edge which is closest to the origin (0,0).

*** When range values are changed (reset), the graph display is cleared and the newly set axes only are displayed.

*** Range setting may cause irregular scale spacing.

*** If the range is set too wide, the graph produced may not fit on the display.

*** Points of deflection sometimes exceed the capabilities of the display with graphs the change drastically as they approach the point of deflection.

*** An Ma ERROR is generated when ranges are extremely narrow.

Range reset

Range values are reset to their initial values by pressing [SHIFT] [Mcl] during range display.

The initial values are as follows.

Xmin	:	-3.8	Ymin	:	-2.2
Xmax	:	3.8	Ymax	:	2.2
Xscl	:	1	Yscl	:	1

- **Use generated function graphs**

After performing range settings, user generated, graphs can be drawn simply by entering the function (formula) after pressing [Graph].

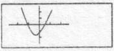
Here, let's try drawing a graph for $y=2x^2+3x-4$.

Set the ranges to the values shown below.

Xmin	:	-5	Ymin	:	-10
Xmax	:	5	Ymax	:	10
Xscl	:	2	Yscl	:	4

Input the functional formula after pressing the [Graph] key.

[Graph] 2 [ALPHA] [X] [x²] [+]
3 [ALPHA] [X] [-] 4 [EXE]



The result produces a visual representation of the formula.

- **Function graph overlay**

Two or more function graphs can be overlaid, which make it easy to determine intersection points and solutions that satisfy all the equations.

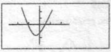
Example: Here, let's find the intersection points of the previously used $y=2x^2 + 3x - 4$ and $y= 2x +3$

First, clear the graph screen in preparation for the first graph.

[SHIFT] [Cls] [EXE]

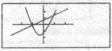
[Graph] 2 [ALPHA] [X] [x^2] [+]

3 [ALPHA] [X] [-] 4 [EXE]



Next, overlay the graph for y = 2x +3

[Graph] 2 [ALPHA] [X] [+] 3 [EXE]



In this way it can be easily seen that there are two intersections for the two function graphs. The approximate coordinates for these two intersections can be found using the Zoom function and the Trace function described in the following section.

*** Be sure to input variable x ([ALPHA] [X]) into the formula when using built-in graphs for overlay.

If variable x is not included in the second formula, the second garph is produced after clearing the first graph.

- **Zoom function**

This function lets you enlarge or reduce the x- and y-coordinates. If you use the Trqace or Plat function to locate he pointer at a specific point on the graph, the enlargement / reduction is performed using the pointer location as he center point.

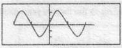
- Enlarging a graph

Example: to enlarge the graph for y=sinx by a factor of 1.5 on the x-axis and 2.0 on the y-axis. Use the following range parameters for the original graph.

Xmin	:	-360 Ymin :	Ymin	:	-1.6
Xmax	:	360	Ymax	:	1.6
Xscl :	:	180	Yscl	:	1

After specifying the range parameters, graph y=sinx.

[Graph] [sin] [ALPHA] [X] [EXE]



Press [SHIFT] [Factor] for the factor specification

[SHIFT] [Factor]

```
Xfact?
          2.
```

1.5

```
Xfact?
1.5_
```

[EXE]

```
Yfact?
          0
```

2

```
Yfact?
2_
```

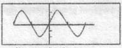
[EXE]

```
Xfact?
          1. 5
```

Press [SHIFT] [Factor] to return to the display that was shown before entering the factor display.

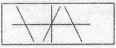
*** Whenever you try to change the factor value while a graph is displayed, the display changes to the text screen. To return to the graph screen after changing the factor value, press[SHIFT]or [EXE].

[EXE] (or[G↔T])



Press [SHIFT] [Zoomxf]to enlarge the graph according to the factors you specified.

[SHIFT] [Zoomxf]



Let's take another look at the range parameters.

Xmin	:	-240	Ymin	:	-0.8
Xmax	:	240	Ymax	:	0.8
Xscl :	:	180	Yscl	:	1

If you press[SHIFT] [Zoomxf]again, the graph is enlarged once more by the factors you specified. To return the graph to its original size, press [SHIFT] [Zoom Org].

● Reducing a graph

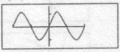
Example: To reduce the graph for y=sinx by a factor of 1.5 on the x-axis and 2.0 on the y-axis. Use the following range parameters for the original graph.

Xmin	:	-360	Ymin	:	-1.6
Xmax	:	360	Ymax	:	1.6
Xscl	:	180	Yscl	:	1

After specifying the range parameters, graph y=sinx.

[SHIFT] [Cls] [EXE]

[Graph] [sin] [ALPHA] [X] [EXE]



Press [SHIFT] [Factor] for the factor specification screen.

[SHIFT] [Factor] — Xfact? 2.

1.5 — Xfact? 1.5_

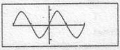
[EXE] — Yfact? 0.

2 — Yfact? 2_

[EXE] — Xfact? 1.5.

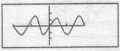
*** Whenever you try to change the factor value while a graph is displayed, the display changes to the text screen. To return to the graph screen after changing the factor value, press [SHIFT] or [EXE]

[EXE] (or [G↔T])
[SHIFT] [Factor]



Press [SHIFT] [Zoomxl/f] to reduce the graph according to the factors you specified.

[SHIFT] [Zoomxl/f]



Let's take another look at the range parameters.

Xmin	:	-540	Ymin	:	-3.2
Xmax	:	540	Ymax	:	3.2
Xscl	:	180	Yscl	:	1

If you press [SHIFT] [Zoomxl/f] again, the graph is reduced once more by the factors you specified. To return the graph to its original size, press [SHIFT] [Zoom Org].

● To specify the zoom factors within a program:

Use the following format to specify the zoom factor in a program.

Factor (Xfactor), (Yfactor)

● **Trace function**

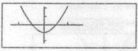
This function lets you move a pointer around a graph and display the x- and y-coordinates of the current pointer location. You enlarge or reduce the x- and y-coordinates. Using the Trace function

Example: To use the Trace function in combination with the Zoom function to analyze the graph for $y=x^2-3$. Use the following range parameters for the original graph (seven significant digits specified).

Xmin	:	-4	Ymin	:	-8
Xmax	:	4	Ymax	:	8
Xscl:	:	2	Yscl	:	4

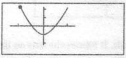
After specifying the range parameters, graph $y=x^2-3$.

[SHIFT] [Cls] [EXE]
[Graph] [ALPHA] [X] [x²] [-] 3 [EXE]



Activate the Trace function

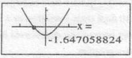
[Trace]



Use [◁] or [▷] to move the pointer along the graph. Each press moves the cursor one point.

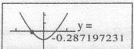
Move the pointer to where the graph intersects the x-axis and press [SHIFT] [Value] to display x-coordinate value.

[SHIFT] [Value]



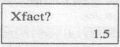
Press [SHIFT] [X↔Y] to change the coordinate display to show the y-coordinate.

[SHIFT] [X↔Y]

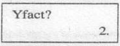


Now use the Zoom function to enlarge the graph. First specify a factor of 2 on the x- and y-axes.

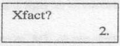
[SHIFT] [Factor]



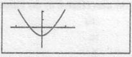
2 [EXE]



[EXE]



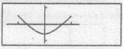
[SHIFT] [Factor]



*** Whenever you try to change the factor value while a graph is displayed, the display changes to the text screen. To return to the graph screen after changing the factor value, press [SHIFT] [Factor].

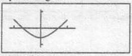
Now enlarge the graph according to the factors.

[SHIFT] [Zoomxf]



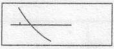
Activate the Trace function and move the pointer again.

[Trace] [SHIFT] [X↔Y] [▷]



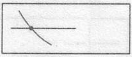
Enlarge the graph again to check the location of the pointer.

[SHIFT] [Zoomxf]



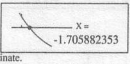
Activate the Trace function and move the pointer again.

[◁] [◁] [Trace] [SHIFT] [X↔Y] [▷] -



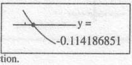
/i: v the x-coordinate value.

[SHIFT] [Value]



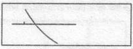
Press [SHIFT] [X↔Y] to display the x-coordinate.

[SHIFT] [X↔Y]



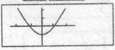
Press [Trace] again to exit the Trace function.

[Trace]



As you can see above, the Trace and Zoom functions can be used to locate the pointer at an approximate point, and then SHIFT value produces a readout of the coordinates.
To return the graph to its original size, press [SHIFT] [Zoom Org]

[SHIFT] [Zoom Org]



(Important)

The pointer does not move at fixed intervals. It follows the dots on the display. Because of this, the values provided for coordinates are approximate.

*** The Trace function only be uesd immediately after a graph is drawn. This function cannot be used if other calculations or operations (except [Range] [Factor] and [G↔T]) have been employed after a graph has been drawn.
*** The x-y coordinate values consist of a 10 digit (max.) mantissa or a 7 digit (max) mantissa plus a 2 digit exponent. Negative values are one digit shorter because one digit is used for the negative sign.
*** The Trace function cannot be written into a program.
*** The Trace function can be used during a "Disp" display.

*** When the format "[Graph] formula ◢ [Graph] formula [EXE]" is executed and a graph is drawn by pressing [EXE] directly after executing the Trace function during halt status, the previous coordinate value remains on the display. After the Trace function is execulted and the text display is brought up using the [G↔T] key, pressing [EXE] causes the next graph to appear and the coordinate value to clear.
Examine the above using [Graph] [ALPHA] [X] [x] [ALPHA] [◢] [Graph] 2 [ALPHA] [X] [+] 5.

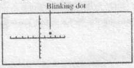
- **Plot function**

The Plot function is used to mark a point on the screen of a graph display. The point can be moved left, right, up and down using the cursor keys, and the coordinates for the graph displayed can be read. Two points can also be connected by a straight line.
Press [SHIFT] [Plot] and specify the x- and y-coordinate after the "Plot" message.
Example: Plot a point at x=2 and y=2 on the axes created by the following range values:

Xmin	:	-5	Ymin	:	-10
Xmax	:	5	Ymax	:	10
Xscl :		1	Yscl	:	2

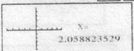
[SHIFT] [Plot] [2] [SHIFT] [,] [2] [EXE]

Blinking dot



The blinking pointer is positioned at the specified coordinates. Press [SHIFT] [Value] to display the x-coordinate value.

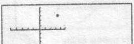
[SHIFT] [Value]



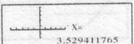
*** Due to limitations caused by the resolution of the display, the actual position of the pointer can only be approximate.

The pointer can be moved left, right, up and down using the cursor keys. The current position of the pointer is always shown at the bottom of the display.

[▷] [▷] [▷] [▷] [▷]



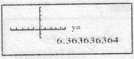
[SHIFT] [Value]



To view the y-coordinate value:

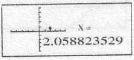
[△] [△] [△] [△] [△]

[SHIFT] [X↔Y]



Now, inputting a new coordinate value causes the new pointer to blink without clearing the present pointer.

[SHIFT] [Plot] 2 [SHIFT] [,] 6.5 [EXE]
[SHIFT] [Value]

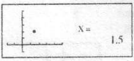


f x-y coordinates are not specified for the Plot function, the pointer appears at the center of the screen.

Set the following range values:

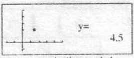
Xmin	:	-2	Ymin	:	-2
Xmax	:	5	Ymax	:	10
Xscl	:	1	Yscl	:	2

[SHIFT][Plot][EXE]
[SHIFT] [Value]



To find the y-coordinate value:

[SHIFT] [X↔Y]



*** Attempting to plot a point outside of the preset range is disregarded.

*** the x- and y-coordinates of the pointer used in the Plot function are respectively stored in the X memory and y memory.

*** A blinking pointer becomes a fixed point (not blinking) when a new pointer is created.

- **Line function**

The Line function makes it possible to connect two points (including the blinking pointer) created with the Plot function with a straight line. With this function, user generated lines can be added to graphs to make them easier to read.

Example: draw perpendiculars from point (2,0) on the x-axis to its intersection with the graph for y=3x. Then draw a line from the point of intersection to the y-axis.

The range values for the graph are as follows:

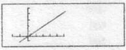
Xmin	:	-2	Ymin	:	-2
Xmax	:	5	Ymax	:	10
Xscl	:	1	Yscl	:	2

Clear the graph display and draw the graph for y=3x.

[SHIFT][Cls][EXE]
[Graph] 3[ALPHA] [X] [EXE]

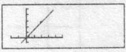
Next, use the Plot function to locate a point at (2,0)

[SHIFT] [Plot] 2 [SHIFT] [,] 0 [EXE]



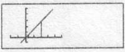
Now plot a point at (2,0) again and use the cursor key (▢) to move the pointer up to the point on the point on the graph (y=3x).

[SHIFT] [Plot] 2 [SHIFT] [,] 0 [EXE]
▢ ~ ▢
(Move the pointer up to the point on the graph for y=3x.) press [EXE] to confirm the point



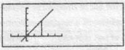
Draw a line using the Line function.

[SHIFT] [Line] [EXE]

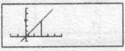


Next, a perpendicular will be drawn from the same point on the graph to the y-axis. First, plot the point on the graph and use the cursor key ▢ to move the pointer to the y-axis. This can be accomplished using Plot X, Y since the x-y coordinates of the point on the graph are stored in the X and Y memories.

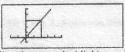
[SHIFT] [Plot] [ALPHA] [X] [SHIFT]
[,] [ALPHA] [X] [EXE]



[SHIFT] [Plot] [ALPHA] [X] [SHIFT]
[,] [ALPHA] [Y] [EXE] ▢ ~ ▢
(Move the pointer to the y-axis)



[SHIFT] [Line] [EXE]



*** The Line function can only be used to draw lines between the blinking pointer and a fixed point created using the Plot function.

- **Graph scroll function**

Immediately after you have drawn a graph, you can scroll it on the display. Use the cursor keys to scroll the graph left, right, up and down.

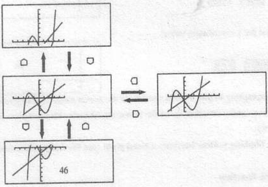
To scroll the graph on the display

Example: To draw the graph for y=0.25 (x+2) (2x+1) (2x-5), y=2x-3, and then scroll it.

Xmin	:	-5	Ymin	:	-8
Xmax	:	5	Ymax	:	8
Xscl :	:	1	Yscl	:	2

SHIFT Cls EXE
Graph 0.25 (ALPHA X + 2)
(2 ALPHA X + 1) (2
ALPHA X - 5) :
Graph 2 ALPHA X - 3 EXE

Press SHIFT Zoom Org to return the graph to its original position after scroll operations.



● Graphing Examples

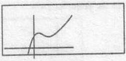
The following examples are presented to show you some ways that the graphing functions can be used effectively.

Example 1: to graph the function $y=x^3-9x^2+27x+50$

Use the following range parameters.

Xmin	:	-5	Ymin	:	-30
Xmax	:	10	Ymax	:	150
Xscl	:	2	Yscl	:	20

SHIFT Cls EXE
Graph ALPHA X x^y 3 -
9 ALPHA X x^2 +
27 ALPHA X + 50 EXE

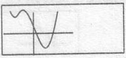


Example 2: To graph the function $y=x^4+4x^3-36x^2-160x+300$ and determine minimum and maximum

Use the following range parameters.

Xmin	:	-10	Ymin	:	-600
Xmax	:	10	Ymax	:	600
Xscl	:	2	Yscl	:	200

SHIFT Cls EXE
Graph ALPHA X x^y 3 +
4 ALPHA X x^y 3 -
36 ALPHA X x^2 - 160
ALPHA X + 300 EXE



4 Program calculations

● Definition of a Program

This unit has a built-in program feature that facilitates repeat calculations. The program feature is used for the consecutive execution of formulas in the same way as the "multi-statement" feature is used in manual calculations. Programs will be discussed here with the aid of illustrative examples.

Example: Find the surface area and volume of a regular octahedron when the length of one side is given.

Length of one side (A)	Surface area (S)	Volume (V)
10cm	( )cm^2	( )cm^2
7	( )	( )
15	( )	( )

*** Fill in the parentheses.

1) Formulas

For a surface area S, volume V and one side A, S and V for a regular octahedron are defined as:

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

2) Programming

Creating a program based on calculation formulas is known as "programming". Here a program will be created based upon the formulas given above. The basis of a program is manual calculation, so first of all, consider the operational method used for manual calculation.

Surface area (S): 2 [x] [√] 3 [x] Numeric value A [x²] [EXE]

Volume (V): [√] 2 [÷] 3 [x] Numeric value A [xʸ] 3 [EXE]

In the above example, mumeric value A is used twice, so it should make sense to store it in memory A before the calculations.

Numeric value A [→] [ALPHA] [A] [EXE]

2 [x] [√] 3 [x] [ALPHA] [A] [x²] [EXE] .... S

[√] 2 [÷] 3 [x] [ALPHA] [A] [xʸ] 3 [EXE] V

With this unit, the operations performed for manual calculations can be used as they are in program. Once program execution starts, it will continue in order without stopping, therefore, commands are required to request the input of data and to display results. The command to request data input is "?", while that to display results is "◢".

A "?" within a program will cause execution to stop temporarily and a "?" to appear on the display as the unit waits for data input. This command cannot be used independently, and is used together, with [→] as "[ALPHA] [?] [→] memory name". **To store a numeric value in memory A, for example: ? → A**

When "?" is displayed, calculation commands and numeric values can be input within 127 steps.

The "◢" command causes program execution to stop temporarily and the latest formula result or alphanumeric characters and symbols to be displayed. This command is used to mark positions in formulas where results are to be displayed. Since programs are ended and their final results dispiayed automatically, this command can be omitted at the end of program. However, if the BASE-N mode is specified for base conversion during a program, do not omit final "◢".

Here these two commands will be used in the previously presented procedure:

[ALPHA] [?] [→] [ALPHA] [A] [:] 2 [x] [√] 3 [x] [ALPHA] [A] [x²] [ALPHA] [◢]

Input to memory A ... Display S

[√] 2 [÷] 3 [x] [ALPHA] [A] [xʸ] 3 (◢ omitted)

3) Program storage

The storage of programs is performed in the WRT mode which is specified by pressing [MODE] [2].

[MODE] [2]

```
WRT
        400   ← Number of remaining steps
P0123456789
```

When [MODE] [2] is pressed, the system mode changes to the WRT mode. Then, the number of remaining steps is indicated. The number of remaining steps is decreased when programs are input or when memories are expanded. If no programs have been input and the number of memories equals 26 (the number of memories at initialization), the number of usable steps should equal 400. The larger figures located below indicate the program areas. If the letter "P" is followed by the number 0 through 9, it indicates that there are no programs stored in areas P0 through P9. The blinking zero here indicates the current program area is P0. Areas into which programs have already been stored are indicated by "_" instead of numbers.

```
WRT
        226
P01_34_6789
```

Here the previously mentioned program will be stored to program area P0 (indicated by the blinking zero):

[EXE] (Start storage)

```
WRT

_
```

[Graph] [sin]
[ALPHA] [?] [→] [ALPHA] [A] [:] 2 [x] [√] 3
[x] [ALPHA] [A] [x²]

```
WRT
        011   ← Number of steps used for program area P0.
?→A:2x√3xA²_
```

[ALPHA] [◢]

```
WRT
        012
→A:2x√3xA²◢_
```

[√] 2 [÷] 3 [x] [ALPHA] [A] [xʸ] 3

```
WRT
        020
²◢√2÷3xAxʸ3_
```

After these operations are complete, the program is stored.

*** After the program is stored, press [MODE] [1] to return to the RUN mode.

4) Program execution

Program are executed in the RUN mode ([MODE] [1]).The program area to be executed is specified using the [Prog] key.

To execute P0: [Prog] 0 [EXE]

To execute P3: [Prog] 3 [EXE]

To execute P8: [Prog] 8 [EXE]

Here the sample program that has been stored will be executed. The surface (S) and volume (V) for the regular octahedron in the sample problem are calculated.

Length of one side (A)	Surface area (S)	Volume (V)
10cm	(346.410615)cm^2	(471.4045208)cm^3
75	(169.7409791)	(161.6917506)
20	(779.4228634)	(1590.990258)

Key operation	Display	
[MODE] [1]	_	
[Prog] [0] [EXE]	?	
[1] [0] [EXE] (Value of A)	Disp 346.4101615	Indicates answer displayed by ◢ (S when A=10)
[EXE]	471.4045208	(V when A=10)
[Prog] [0] [EXE]	?	
[7] [EXE] (Value of A)	Disp 169.7409791	(S when A=7)
[EXE]	161.6917506	(V when A=7)
[Prog] [0] [EXE]	?	
[1] [5] [EXE] (Value of A)	Disp 779.4228634	(S when A=15)
[EXE]	1590.990258	(V when A=15)

*** Program calculations are performed automatically with each press of [EXE] when it is pressed after data is input or after the result is read.

*** Directly after a program in P0 is executed by pressing [Prog] [0] [EXE] as in this example, the Prog 0 command is stored by the replay function. Therefore, subsequent executions of the same program can be performed by simply pressing [EXE].

Operation [Prog] 0 [EXE] (P0 program execution)
10 [EXE] (Input 10 for A)
[EXE] (Display V when A=10)
[EXE] (Re-execute)
7 [EXE] (Input 7 for A)
[EXE] (Display V when A=7)

• Program Checking and Editing (Correction, Addition, Deletion)

Recalling a stored program can be performed in order to verify its contents. After specifying the desired program area using ▷ or ◁ in the WRT mode ([MODE] [2]), the program contents will be displayed by pressing the [EXE] key. Once the program is displayed, the ▷ or ◁ key is used to advance the program one step at a time for verification. When the program has been improperly stored, editing can also be performed by adding to it or erasing portions. Here a new program will be created by checking and editing the previous sample program (the surface area and volume of a regular tetrahedron).

Example: Find the surface area and volume of a regular tetrahedron when the length of one side is given.

Length of one side (A)	Surface area (S)	Volume (V)
10cm	( )cm^2	( )cm^3
7.5	( )	( )
20	( )	( )

- Formulas

For a surface area S, volume V and one side A, S and V for a regular octahedron are defined as:

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

- Programming

As with the previous example, the length of one side is stored in memory A and the program then constructed.

Numeric value A [→] [ALPHA] [A] [EXE]

[√] 3 [×] [ALPHA] [A] [x²] [EXE] .... S

[√] 2 [÷] 12 [×] [ALPHA] [A] [x^y] 3 [EXE] V

When the above is formed into a program, it appears as follows:

[ALPHA] [?] [→] [ALPHA] [A] [:] [√] 3 [×] [ALPHA] [A] [x²] [ALPHA] [◢]
[√] 2 [÷] 12 [×] [ALPHA] [A] [x^y] 3

- Program editing

First, a comparison of the two programs would be helpful.

Octahedron: [ALPHA] [?] [→] [ALPHA] [A] [:] 2 [×] [√] 3 [×] [ALPHA] [A] [x²] [ALPHA] [◢]
[√] 2 [÷] 3 [×] [ALPHA] [A] [x^y] 3

Tetrahedron: [ALPHA] [?] [→] [ALPHA] [A] [:] [√] 3 [×] [ALPHA] [A] [x²] [ALPHA] [◢]
[√] 2 [÷] 12 [×] [ALPHA] [A] [x^y] 3

The octahedron program can be changed to a tetrahedron program by deleting the parts marked with wavy lines, and changing those that are marked with straight lines.

In actual practice, this would be performed as follows:

Key operation	Display	
[MODE] [2]	WRT 380 P_123456789	
[EXE]	WRT 000 ?→A:2×√3×A²◢	
▷ ▷ ▷ ▷ [DEL] [DEL]	WRT 004 ?→A:√3×A²◢√2	Locate cursor at position to be deleted and delete two characters.
▷ × 9 times [SHIFT] [INS] [1] [2]	WRT 015 √3×A²◢√2÷123	Insert two characters.
[SHIFT] [INS] [DEL]	WRT 015 √3×A²◢√2÷12×	Delete unnecessary 3.
[MODE] [1]	_	Editing complete. Return to the RUN mode.

4) Program execution

Now this program will be executed.

Length of one side (A)	Surface area (S)	Volume (V)
10cm	(173.2050808)cm²	(117.8511302)cm³
7.5	(97.42785793)	(49.71844555)
20	(692.820323)	(942.8090416)

Keys	Display
MODE 1	_
Prog 0 EXE	?
1 0 EXE	173.2050808
EXE	117.8511302
Prog 0 EXE	?
7 [ ] 5 EXE	97.42785793
EXE	49.71844555
Prog 0 EXE	?
2 0 EXE	692.820323
EXE	942.8090416

(Summary)

	Operation	Keys used
Program check	• WRT mode specification • Program area specification (Omitted if P0) • Start verification • Verification of contents	MODE 2 ◁ ▷ EXE ◁ ▷
Correction	• Move the cursor to the position to be corrected • Press correct keys.	◁ ▷
Deletion	• Move the cursor to the position to be corrected • Delete	◁ ▷ DEL
Insertion	• Move the cursor to the position to be inserted into • Specify the insert mode. • Press desired keys.	◁ ▷ SHIFT INS

● Program debugging (Correcting Errors)

After a program has been created and input, it will sometimes generate error messages when it is executed, or it will produce unexpected results. This indicates that there is an error somewhere within the program that needs to be corrected. Such programming errors are referred to as "bugs", while correcting them is called "debugging".

● Debugging when an error message is generated

An error message is displayed as follows:

```
P0 Syn ERROR
```

Type of error — "Syn ERROR"; Program area where error was generated — "P0"

The error message informs the operator of the program area (P0 to P9) in which the error was generated. It also states the type of error, which gives an idea of the proper countermeasure to be taken.

● Error messages

There are a total of seven error messages.

1) Syn ERROR (Syntax error): Indicates a mistake in the formula or a misuse of program commands.
2) Ma ERROR (Mathematical error): Indicates the calcultaion result of a numeric expression exceeds 10^{100}, an illogical operation (e.g. division by zero), or the input of an argument that exceeds the input range of the function.
3) Go ERROR (Jump error): Indicate a missing Lbl for the Goto command, or that the program area for the Prog command does not contain a program.
4) Stk ERROR (Stack error): indicates the calculation performed exceeds the capacity of the stack for numeric values or for commands.
5) Mem ERROR (Memory error): Indicates the attempt to use a memory name such as Z[5] without having expanded memories.
6) Arg ERROR (Argument error): Indicates the argument of a command or specification in a program exceeds the input range (e.g. Sci 10, Goto 11)

Further operation will become impossible when an error message is displayed.

Press AC, ◁ or ▷ to cancel the error.

Press [AC] cancels the error and new key input becomes possible.
With this operation, the RUN mode is maintained.
Pressing ◁ or ▷ cancels the error and changes the system mode to the WRT mode. The cursor is positioned at the location where the error was generated to allow modification of the program to eliminate the error.

- **Checkpoint for each type of error**

The following are checkpoints for each type of error:

1) Syn error: Verify again that there are no errors in the program.
2) Ma ERROR: For calculations that require use of the memories. Check to see that the numeric values in the memories do not exceed the range of the arguments. This type of error often occurs with division by 0 or the calculation of negative square roots.
3) Go ERROR: Check to see that there is a corresponding Lbl *n* when Goto *n* is used. Also chick to see that the program in P n has been correctly input when Prog n is used.
4) Stk ERROR: Check to see that the formula is not too long thus causing a stack overflow, If this is the case, the formula should be divided into two or more parts.
5) Mem ERROR: Check to see that memories were properly expanded using " [MODE] [.] *n* [EXE]" (Defm). When using array-type memories, check to see that the subscripts are correct.
6) Arg ERROR: Check whether values specified by [MODE] [7] (Sci) or [MODE] [8] (Fix) are within the range of 0~9. Also check whether values specified by Goto, Lbl, or Prog commands are within 0—9. Also ensure that memory expansion using [MODE] [.] (Defm) is performed within the remaining number of steps and that the value used for expansion is not negative.

● Counting the Number of Steps

The program capacity of this unit is 400 steps.
The number of steps indicates the amount of storage space available for program., and it will decrease as programs are input. The number of remaining steps will also be decreased when steps are converted to memories. There is one method to determine the current number of remaining steps:

Specify the WRT mode ([MODE] [2]), and the number of remaining steps will appear. At this time the status of the program areas can also be determined.

[MODE] [2]

```
WRT
        381
P_123456789
```

381 Number of remaining steps

Basically, one function requires a single step, but there are some commands where one function requires two steps.

- One function / one step: sin、cos、tan、log、(,)、:、1、2、3 etc.
- One function / two steps: Lbl 1 、Prog 8 etc.

Each step can be verified by the movement of the cursor:

Example:

```
WRT
                  000
? →A: √3×A2◢ √2
```

Present cursor position

At this time, each press of a cursor key (◁ or ▷) will cause the cursor to move to the next sequential step.

Example:

▷ ▷ ▷ ▷ ▷ ▷

```
WRT
                  006
? →A: √3×A2◢ √2
```

6th step

● Program Areas and Calculation Modes

This unit contains a total of 10 program areas are all utilized in the same manner, and 10 independent programs can be input. One main program (main routine) and a number of secondary programs (subroutines) can also be stored. The total number of steps available for storage in program areas P0 through P9 is 400 maximum. Specification of a program area is performed as follows:

RUN mode: Press any key from 0 through 9 after pressing the Prog key. Then press [EXE].

Example: P0 [Prog] [0] [EXE]
P8 [Prog] [8] [EXE]

*** In this mode, program execution begins when [EXE] is pressed.

WRT mode: Use ◁ or ▷ to move the cursor under the program area to be specified and press [EXE].
Only the numbers of the program areas that do not yet contain programs will be displayed. "_" symbols indicate program areas which already contain program.

Example:

```
        WRT
        292
P_123__67_9
```

Programs already stored in these program area

● Program area and calculation mode specification in the WRT mode

Besides normal function calculations, to perform binary, octal, decimal and hexadecimal calculatons and conversions, standard deviation claculations, and regression calculations in a program, a calculation mode must be specified. Program mode specification and program area specification are performed at the same time. First the WRT mode is specified ([MODE] [2]), and then a calculation mode is specified. Next, the program area is specified, and, when [EXE] is pressed, the calculation mode is memorized in the program area.

Henceforth, stored programs will be accompanied with the calculation mode.

Example: Memorizing the BASE-N mode in P2

Key operation	Display	
[MODE] [2]	WRT 400 P0123456789	As suming that nothing is stored
▷ ▷	BASE-N WRT 400 P0123456789	Specify P2
[MODE] [-]	BASE-N WRT 400 P0123456789	Specify the BASE-N mode
[EXE]	BASE-N WRT 000 _	

As shown above, the calculation mode will be memorized into a program area.

● Cautions concerning the calcutation modes

All key operations available in each calculation mode can be stored as programs, but, depending on the calculation mode, certain commands of functions cannot be used.

BASE-N mode

- Function calculations cannot be performed.
- Units of angular measurement cannot be specified.
- All program commands can be used.
- Be sure to include a "◢" at the final result output to return to the previous calculation mode when a program execution is terminated. Failure to do so may result in a decimal display or an error.

SD1, SD2 mode

- Among the functions, Abs and $\sqrt[3]{\ }$ cannot be used.
- Among the program commands, Dsz, > and < cannot be used.

LR1, LR2 mode

- Among the functions, Abs and $\sqrt[3]{\ }$ cannot be used.
- Among the program commands, ⇒, =, ≠, Isz, ≥, ≤, Dsz, < and > cannot be used.

● Erasing Programs

Erasing of programs is performed in the PCL mode. Press [MODE] [3] to specify the PCL mdoe. There are two methods used to erase programs: erasing a program located in a single program area and erasing all programs.

● Erasing a single program

To erase a program in a single program area, specify the PCL mode and press the [AC] key after specifying the program area.

Example: erase the program in P3 only.

Key operation	Display	
[MODE] [3]	PCL 302 P_12_45678_	P0, P3 and P9 already contain programs
▷ ▷ ▷	PCL 302 P_12_45678_	Align cursor with P3
[AC]	BASE-N 345 P_12345678_	Number 3 appears after deletion
[MODE] [1]	_	Return to RUN mode

- **Erasing all programs**

To erase all programs stored in program areas 0 through 9, specify the PCL mode and press [SHIFT] and the [Mcl].
Example: erase the programs stored in P0, P4, P8 and P9.

Key	Display
[MODE] [3]	PCL 273 P_123_567__
[SHIFT] [Mcl]	PCL 400 P0123456789
[MODE] [1]	.

● Convenient Program Commands

The programs for this unit are made based upon manual calculations. Special program commands, however, are available to allow the selection of the formula, and repetitive execution of the same formula.
Here, some of these commands will be used to produce more convenient programs.

- **Jump commands**

Jump commands are used to change the flow of program execution.
Programs are executed in the order that they are input (from the lowest step number first) until the end of the program is reached. This system is not very convenient when there are repeat calculations to be performed or when it is desirable to transfer execution to another formula. It is in these cases, however, that the jump commands are very effective. There are three types of jump commands: a simple unconditional jump to a branch destination, conditional jumps that decide the branch destination by whether a certain condition is true or not, and count jumps the increase or decrease a specific memory by one and then decide the branch destination after checking whether the value stored equals zero or not.

- **Unconditional Jump**

The unconditional jump is composed of "Goto" and "Lbl". When program execution reaches the statement "Goto n" (where n is number from 0 through 9), execution then jumps to "Lbl n" (n is the same value as Goto n). The unconditional jump is often used in simple program to return execution to the beginning for repetitive calculatons, or to repeat calculations from a point within a program. Unconditional jumps are also used in combination with conditional and count jumps.

Example: The previously presented program to find the surface area and volume of a regular tetrahedron will be rewritten using "Goto 1" and "Lbl 1" to allow repeat calculations.
The previous program contained:

?, →, A, :, √, 3, ×, A, x², ◢,
√, 2, +, 1, 2, ×, A, xʸ, 3   19 steps

** Here after, commas (,) will be used to separate steps for the sake of clarity.
Add "Goto 1" to the end of the program, and add "Lbl 1" to the beginning of the program as the branch destination.
If this is simply left the way it is, however, the volume will not be displayed and execution will move immediately to the input of one side at the beginning. To prevent this situation, insert a display command (◢) in front of the Goto1".
he complete program with the unconditional jump added should look like this:

Lbl, 1, :, ?, →, A, :, √, 3, ×, A, x², ◢,
√, 2, +, 1, 2, ×, A, xʸ, 3, ◢, Gpto, 1   25 steps

ow let's try executing this program.
* Henceforth, the displays will only show calculation result output.

Key	Display	
[Prog] [0] [EXE]	?	Stored in P0. The length
10 [EXE]	173.2050808	of the side =10
[EXE]	117.8511305	
[EXE]	?	
7.5 [EXE]	97.42785793	The length of the side =7.5
[EXE]	49.71844555	
[EXE]	?	

ce the program is in all endless loop, it will continue execution. To terminate execution, press [MODE] [1].

Key	Display
[MODE] [1]	_

sides the beginning of the program, branch destinations can be designated at any point within the program.
ample: calculate y=ax+b when the value for x changes each time, while a and b can also change ending upon the calculation.

?, →, A, :, ?, →, B, :, Lbl, 1, :, ?, →, X, :,
A, ×, X, +, B, ◢, Goto, 1   23 steps

When this program is executed, the values for a and b are stored in memories A and B respectively. After that, only the value for x can be changed.

In this way an unconditional jump is made in accordance with "Goto" and "Lbl", and the flow of program execution is changed. When there is no "Lbl n" to correspond to a "Goto n", an error (Go ERROR) is generated.

- **Conditional jumps**

The conditional jumps compare a numeric value in memcry with a constant or a numeric value in another memory. If the condition is true, the statement following the "⇒" is executed, and if the condition is not true, execution skips the statement and continues following the next " : " or "◢".

Conditional jumps take on the following form:

Left side Relational operator Right side ⇒ Statement [: / ◢]* Statement

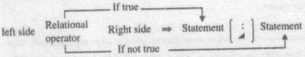
* Any one can be used.

One memory name (alphabetic character from A through Z), constant numeric values or calculation formulas (Ax2, D—E, etc.) are used for " left side" and "right side".

The relational operator is a comparision symbol; there are 6 types of relational operators: =, ≠, ≥, ≤, >, <.

Left side = right side (left side equals right side)
Left side ≠ right side (left side doesn't equal right side)
Left side ≥ right side (left side is greater than or equal right side)
Left side ≤ right side (left side is less than or equal right side)
Left side > right side (left side is greater than right side)
Left side < right side (left side is less than right side)

The "⇒" is displayed when SHIFT ⇒ are pressed. If the condition is true, execution advances to the statement following ⇒. If the condition is not true, the statement following ⇒ is skipped and execution jumps to the statement following the next " : " or "◢".



A statement is a calculation formula (sin A×5, etc.) or a program command (Goto, Prog, etc.), and everything up to next " : " or " is " is regarded as one statement.

Example: If an input numeric value is greater than or equal to zero, calculate the square root or that value. If the input value is less than zero, reinput another value.

Program: Lbl, 1, : , ?, →, A, : , A, ≥, 0, ⇒, √, A, ◢, Goto, 1   16 steps

In this program, the input numeric value is stored in memory A, and then it is tested to determine whether it is greater than, equal to or less than zero. If the contents of memory A are greater than or equal to 0 (not less than zero), the statement (calculation formula) located between "⇒" and "◢" will be executed, and then Goto 1 returns execution to Lbl 1. If the contents of memory A are less than zero, execution will skip the following statement to the next "◢" and return to Lbl 1 by Goto 1.

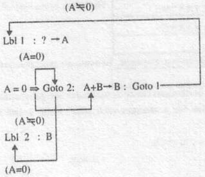
Example: Calculate the sum of input numeric values. If a 0 is input, the total should be displayed.

Program 0, →, B, : ,
Lbl, 1, : , ?, →, A, : , A, =, 0, ⇒, Goto, 2, : ,
A, +, B, →, B, : , Goto, 1, : ,
Lbl, 2, : , B   31 steps

In this program, a 0 is first stored in memory B to clear it for calculation of the sum. Next, the value input by "? →A" is stored in memory A by "A=0 ⇒" and it is determined whether or not the value stored in memory A equals zero. If A=0, goto 2 causes execution to jump to Lbl 2. If memory A does not equal 0, Goto then Goto 1 returns execution to Lbl 1.

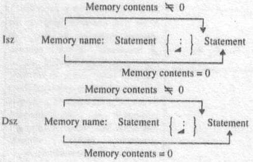
Execution from Lbl 2 will display the sum that has been stored in memory B. Actually, the display command "◢" is inserted following B, but here it can be omitted.

The following illustration shows the flow of the program:



- **Count jumps**

The count jumps cause the value in a specified memory to be increased or decreased by 1. If the value does equal 0, the following statement is skipped, and the statement following the next " : " or "◢" is excuted. The "Lsz" command is used to increase the value in memory by 1 and decide the subsequent excution, while the "Dsz" command is used to decrease the value by 1 and decide.



Example: Increase memory A by one .... Isz A
Decrease memory B by one .... Dsz B

Program:
```
1, 0, →, A, : , 0, →, C, : ,
Lb1, 1, : , ?, →, B, : , B, +, C, →, C, : ,
Dsz. A, : , Goto, 1, :. , C, +, 1, 0
```
32 steps

In this program. first 10 is stored in memory A, and 0 is stored in memory C. Memory A is used as the "counter" and countdown is performed the specified number of times by the Dsz command. Memory C is used to store the sum of the inputs, and so first must be cleared by inputting 10. The numeric value input in response to "?" is stored in memory B, and then the sum of the input values is stored in memory C by "B+C C".

The statement Dsz A then decreases the value stored in memory A by 1. If the result does not equal 0, the following statement, Goto 1 is executed. If the result equals 0, the following Goto 1 is skipped and "C+10" is executed.

Example: Determine the altitude at one-second intervals of a ball thrown into the air at an initial velocity of V (m/sec) and an angle of S° . The formula is expressed as: h = Vsin θ t- ½ gt , with g=9.8, with the effects of air resistance being disregarded.

Program:
```
Deg, : , 0, →, T, : , ?, →, V, : , ?, →, S, : ,
Lb1, 1, : , Isz, : , V,× , sin, S,× , T, -,
9,   , 8, ×, T, x², +, 2, ◢, Goto, 1
```
38 steps

38 steps

In this program the unit of angular measurement is set and memory T is first initialized (cleared). Then the initial velocity and angle are input into memories V and S respecticely.

LbI 1 is used at the beginning of the repeat calculations. The numeric value stored in memory T is counted up (increased by 1) by Isz T. In this case, the Isz command is used only for the purpose of increasing the value stored in memory T, and the subsequent jump does not depend upon any comparison or decision. The Isz command can also be used in the same manner as seen with the Dsz command for jumps that require decisions, but, as can be seen here, it can also be used to simply increase values. If, in place of the Isz command, another method such as "T+1" is used, five steps are required instead of the two for the (Isz T) method shown here. Such commands are convenient ways of conserving memory space. Each time memory T is increased, calculation is performed according to the formula, and the altitude is displayed. It should be noted that this program is endless, so when the required value is obtained MODE 1 are pressed to terminate the program.

(summary)

command	Formula	Operation
Unconditional jump	Lbi n Goto n (n = natural number from 0 through 9)	Performs unconditional jump to Lbl n corresponding to Goto n
Conditional jumps	Left Side Relational Operator Right Side ⇒ Statement { : / ◢ } Statement (Relational operators: =, ≠, >, <, ≧, ≦)	Left and right sides are compared. If the conditional expression is true, the statement after ⇒ is executed. If not true, execution jumps to the statement following the next : or ◢ Statements include numeric expressions, Goto commands, etc
Count jumps	Isz Memory name: Statement { : / ◢ } Statement Dsz Memory name: Statement { : / ◢ } Statement (Memory name consists of single character from A through Z, such as A{ }, etc.)	Numeric value stored in memory is increased (Isz) or decreased (Dsz) by one. If result equals 0, a jump is performed to the statement following the next : or ◢. statements include numeric expressions, Goto commands, etc.

● Array-Type Memories

● Using array-type memories

Up to this point all of the memories used have been referred to by single alphabetic characters such as A, B, X, or Y. With the array-type memory introduced here, a memory name (one alphabetic character from A through Z) is appended with a subscript such as [1] or [2].

*** Brackets are input by [ALPHA] [[] and [ALPHA] []].

Standard memory	Array-type memory	
A	A[0]	C[-2]
B	A[1]	C[-1]
C	A[2]	C[0]
D	A[3]	C[1]
E	A[4]	C[2]

Proper utilization of subscripts shortens programs and makes them easier to use. Negative values used as shbscripts are counted in relation to memory zero as shown above.

Example: Input the numbers 1 through 10 into memories A throught J.

Using standard memories

```
1, →, A. : , 2. →, B.  , 3. →, C. : , 4. →, D. : ,
5. →, E. : , 6. →, F. : , 7. →, G. : , 8. →, H. : ,
9. →, I. : , 1, 0. →, J                          40 steps
```

using array-type memories

```
0. →, Z. : , Lbl, 1. : , Z, +, 1. →, A, [, Z, ], : ,
Isz, Z. : , Z. <, 1, 0, ⇒, Goto, 1               26 steps
```

In the case of using standard memories, inputting values into memories one by one is both inefficient and time consuming. What happens if we want to see a value stored in a specific memory?

Using standard memories

```
Lbl, 1. : , ?, →, Z. : ,
Z, =, 1, ⇒, A, ◢, Z, =, 2, ⇒, B, ◢,
Z, =, 3, ⇒, C, ◢, Z, =, 4, ⇒, D, ◢,
Z, =, 5, ⇒, E, ◢, Z, =, 6, ⇒, F, ◢,
Z, =, 7, ⇒, G, ◢, Z, =, 8, ⇒, H, ◢,
Z, =, 9, ⇒, I, ◢, Z, =, 1, 0, ⇒, J, ◢ ,
Goto, 1                                          70 steps
```

Using array-type memories

```
Lbl, 1. : , ?, →, Z. : , A, [, Z, -, 1, ], ◢.
Goto, 1                                          16 steps
```

The difference is readily apparent. When using the standard memories, the input value is compared one by one with the value assigned to each memory (e.g. A=1, B=2,…)

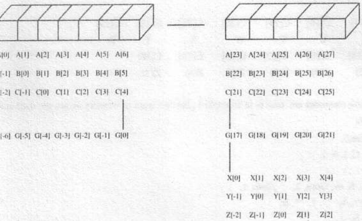
With the array-type memories, the input value is immediately stored in the proper memory determined by "[Z-1]". Formulas (Z-1 A+10, etc.) can even be used for the subscript.

• Cautions when using array-type memories

When using array-type memories, a subscript is appended to an alphabetic character that represents a standard memory from A through Z.

Therefore, care must be taken to prevent overlap of memories.

The relation is as follows:



The following shows a case in which array-type memories overlap with standard format memories. This situation should always be avoided.

Example: Store the numeric values from 1 through 5 in memories A[1] through A[5] respectively.

```
5, →,C, : ,Lbl, 1, : ,C →,A, [,C ,], : ,
Dsz, C. : , Goto, 1, : ,
A, [,1 , ], ◢,A, [, 2, ], ◢, A, [, 3, ], ◢,
A, [, 4, ], ◢ A, [, 5, ]                        44 steps
```

In this program, the values 1 through 5 are stored in the array-type memories A[1] through A[5], and memory C is used as a counter memory. When this program is executed.the following results are obtained:

Prog 0 EXE	1.
EXE	0.
EXE	3.
EXE	4.
EXE	5.

As can be seen, the second displayed value (which should be 2) inA[2] is incorrect. This problem has occurred because memory A[2] is the same as memory C.

A	B	C	D	E	F
	A[1]	A[2]	A[3]	A[4]	A[5]

The content of memory C (A[2]) is decreased from 5 to 0 in steps of 1.

Therefore, the content of memory A[2] is displayed as 0.

• Application of the array-type memories

It is sometimes required to treat two different types of data as a single group. In this case, memories for data processing and those for data storage should be kept separate.

Example: store data x and y in memories. When an x value is input, the corresponding y value is displayed. There will be a total of piece of data.

Example program 1

Memory A is used as the data control memory, and memory B is used for temporary storage of the x data. The x data are stored in memories C[1] (memory D),through C[15](memory R), and the y data are stored in memories C[16] (memory S) throughC[30] (memory Z(7)).

```
1, →, A, : ,Defm, 7, : ,
Lbl,1, : , ?, →, C,[, A, ], : ,
?, →, C, [,A, +, 1, 5, ], : ,
Isz, A, a, , A, =, 1, 6,⇒ Goto, 2, : , Goto, 1, : ,
Lbl, 2, : , 1, 5, →, A, : , ?, →, B, : ,
B, =, 0, ⇒, Goto, 5, : ,
Lbl, 3, : , B, =, C,[, A, ], ⇒, Goto, 4, : ,
Dsz, A, : , Goto, 3, : , Goto, 2, : ,
Lbl,4, : , C, [,A, +, 1, 5, ],◢, Goto, 2, : ,
Lb, 5                                                98 steps
```

In this program, memories are used as follows:

x data:	C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]	C[8]
	D	E	F	G	H	I	J	K
	C[9]	C[10]	C[11]	C[12]	C[13]	C[14]	C[15]	
	L	M	N	O	P	Q	R	
y data:	C[16]	C[17]	C[18]	C[19]	C[20]	C[21]	C[22]	C[23]
	S	T	U	V	W	X	Y	Z
	C[24]	C[25]	C[26]	C[27]	C[28]	C[29]	C[30]	
	Z(1)	Z(2)	Z(3)	Z(4)	Z(5)	Z(6)	Z(7)	

Example program 2: The same memories are used as in Example 1 , but two types of memory names are used and the x and y data kept separate

```
1, →, A, : , Defm,7, : ,
Lbl,1, : , ?, , →, C, [, A ,], : ,
?, →, R, [, A, ], : , :
Isz,A, : : , A, =,1, 6, ⇒, Goto, 2, : ,Goto, 1, : ,
Lbl,2, : , 1, 5, →, A, : , ?, →, B, : ,
B, =,0, ⇒, Goto, 5, : ,
Lbl,3, : , B, =, C, [, A, ], ⇒, Goto, 4, : ,
Dsz,A, : , Goto, 3, : , Goto, 2, : ,
Lbl. 4, : , R, [, A, ], ◢, Goto 2 ,] : ,
Lb, 5                                          92 steps
```

Memories are used as follows:

x data:	C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]	C[8]
	D	E	F	G	H	I	J	K
	C[9]	C[10]	C[11]	C[12]	C[13]	C[14]	C[15]	
	L	M	N	O	P	Q	R	
y data:	R[1]	R[2]	R[3]	R[4]	R[5]	R[6]	R[7]	R[8]
	S	T	U	V	W	X	Y	Z
	R[9]	R[10]	R[11]	R[12]	R[13]	R[14]	R[15]	
	Z(1)	Z(2)	Z(3)	Z(4)	Z(5)	Z(6)	Z(7)	

In this way, the memory names can be changed. However , since memory names are restricted to the letters from A through Z, the expanded memories ([MODE] [.]) can only he used as array- type memories.

*** The memory expansion command (Defm) can be used in a program.

Example : Expand the number of memories by 14 to make a total of 40 available.

Defm, 1,4, : , ……

● Displaying Alpha-Numeric Characters and Symbols

Dispiaying Alpha-Numeric Characters and Symbols Alphabetic characters, numbers, calculation command symbols, etc. can be displayed as messages. They are enclosed in quotation marks ([ALPHA] [""]),

Alpha-numeric characters and symbols

- Characters and symbols displayed when pressed following [ALPHA] :
  [,], k, m, µ n, p, f, space, A, B, C, D, E, F, G, H,
  I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z,
- Other numbers, symbols, calculation commands, program command
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
  (,), √, ₓ, +, −, ×, ÷, …
  sin, cos, tan, log, ln,
  =, ≠, >, <, ≥, ≤, …
  A, B, C, D, E, F. d, h, b, o
  Neg, Not, and, or, xor
  x, y, xon, son-1, …
  ° ([SHIFT] [MODE] [4]), ʳ ([SHIFT] [MODE] [5]), ᴳ ([SHIFT] [MODE] [6])

*** All of the above noted characters can be used in the same manner as the alphabetic characters.

In the preceding example requiring an input of two types of data (x, y), the prompt "?" does not give any information concerning the type of input expected. A message can be inserted before the "?" to verify the type of data required for input.

Lbl, 1, : , ?, →, X, : , ?, →, Y, : , …

The message "X=" and "Y=" will be inserted into this program.

Lbl, 1, : , ", X, =, ", ?, →, X, : ,

", Y, =, ", ?, → Y, : , …

If messages are included as shown here, the display is as follows(Assuming that the program is stored in P1)

[Prog] [1] [EXE]	X = ?
10 [EXE]	Y = ?

Messages are also convenient when displayed result in program calculations.

Example:

Lbl, 0, : , ", N, =, ", ?, →, B, -, C, : ,
0, →, A, : ,
Lbl, 1, : , C, ÷, 2, →, C, : , Frac, C, ≠, 0, →, Goto, 3,
: , Isz, A, : , C, =, 1, →, Goto, 2, : , Goto, 1, : ,
Lbl, 2, : , ", X, =, ", ◢, A, ◢, Goto, 0, : ,
Lbl, 3, : , ", N, O, ", ◢, Goto, 0 70 steps

This program calculates the x power of 2, A prompt of "N=?" appears for data input. The result is displayed by pressing [EXE] while "X=" is displayed. When an input data is not the x power of 2, the display "NO" appears and execution returns to the beginning for reinput.

*** Always follow a message with a ◢ whenever a formula follows the message.

Assuming that the program is stored in P2:

[Prog] [2] [EXE]	N = ?
4096 [EXE]	X =
[EXE]	12.
[EXE]	N = ?
3124 [EXE]	NO
[EXE]	N = ?
512 [EXE]	X =
[EXE]	9.

The display is capable of showing up to 12 alpha characters at one time. For messages that are longer than 12 characters use ◢ (Disp) to divide the message.

● Using the Graph Function in Programs

Using the graph function within programs makes it possible to graphically represent long. complex equations and to overwrite graphs repeatedly. All graph commands (except the Trace function) can be included in programs. Range values can also be written into the program.

Generally, manual graph operations can be uesd in programs without modification.

Example 1: Graphically determine the number of solutions (real roots) that satisfy both of the following two equations.

$y=x^4-x^3-24x^2+4x+80$

$y=10x-30$

The range values are as follows.

Xmin	:	-10	Ymin	:	-120
Xmax	:	10	Ymax	:	150
Xscl	:	2	Yscl	:	50

First, program the range setting. Note that values are separated from each other by commas ",".

Range, (-), 1, 0, ' , 1, 0, ' , 2, ' , (-), 1, 2, 0, ' , 1, 5, 0, ' , 5, 0

Next, program the equation for the first graph.

Graph, X, x^y, 4, -, X, x^y, 3, -, 2, 4, X, x^2, +, 4, X, +, 8, 0

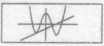
Finally program the equation for the second graph.

Graph, 1, 0, X, -, 3, 0 Total 47 steps

When inputting this program, press [:] after input of the ranges and the first equation.

The following should appear on the display when the program is executed:

[Prog] [0] [EXE]



A "◢" can be input after the first equation to suspend execution after the first graph is produced. To continue execution to the next graph, press [EXE].

The procedare outlined above can be used to produce a wide variety of graphs.

The library of this manual includes a number of examples of graph programming.

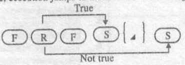
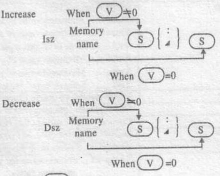
(B) Function Reference

1. Manual Calculations-

Mode specification	COMP mode ([MODE] [+])	Four airthmetic and function calculations.
	BASE-N mode ([MODE] [-])	Binary, octal, decimal, hexdeicimal conversions and calculations, logical operations.
	SD1 mode ([MODE] [x])	Standard deviation calculations (1-variable statistical).
	LR1 mode ([MODE] [+])	Regression calculations (paired variable statistical).
	SD2 mode ([SHIFT] [MODE] [x])	For production of single variable statistical graphs. (Bar graphs, normal distribution curves)
	LR2 mode ([SHIFT] [MODE] [+])	For Production of paired variable statistical graphs. (regression lines)
Functions	Type A function	Function command input immediately after numeric value. (x2, x^{-1}, x!, ° r g, " ' ")
	Type B function	Function command input immediately before numeric value. (sin, cos, tan, $\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$, sinh, cosh, tanh, $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$, log in, e^x, 10^x, $\sqrt{\ }$, $\sqrt[3]{\ }$, Abs, Int, Frac, etc.)
	Variable function	Function command input immediately before numeric value. A x^y B (A to the Bth power), B $\sqrt[x]{\ }$A (A to the 1/Bth power), Pol (A,B), Rec (A,B) *** A and B are numeric values.
← ←	Immediately executed functions	Displayed value changed with each press of a key. (ENG, ENG, ° ' ")
Binary, octal, decimal, hexadecimal calculations	Setting number system	Decimal.................. [Dec] [EXE] Hexadecimal........... [Hex] [EXE] Brnary..................... [Bin] [EXE] Octal........................ [Oct] [EXE]
	Number system specification	Number system for the numeric value entered immediately after can be specified regardless of the currently Set number system. To specify: Decimal................. [SHIFT] [d] Hexadeciaml .... [SHIFT] [h] Binary..................... [SHIFT] [b] Octal....................... [SHIFT] [o]
	Logical operations	Input numeric values are converted to binary and each bit is rested. Result is converted back to number system used for input, and then display. not .... Reverse of each bit and .... Logical product of each bit or .... Logical sum of each bit ror .... Exclusive logical sum of each bit xnor .... Exclusive negative logical sum of each bit
Standard deviation calculations ([MODE] [x])	Data clear	[SHIFT] [Cls] [EXE]
	Data imput	Data [frequency] [DT] *** Frequency can be omitted.
	Data deletion	Data [frequency] [CL] *** Frequency can be omitted.
	Result display	Number of data (n) ....Mode [ALPHA] [3] (n) [EXE] Sum (Σx) ....Mode [ALPHA] [2] (Σx) [EXE] Sum of squares (Σx^2) ....Mode [ALPHA] [1] (Σx^2) [EXE] Mean (x) .... [SHIFT] [x̄] (n) [EXE] Population standard deviation (xσn) .... [SHIFT] [xσn] [EXE] Sample standard deviation (xσn-1) .... [SHIFT] [xσn-1] [EXE]

Regression calculation ([MODE] [+])	Data clear	[SHIFT] [Scl] [EXE]
	Data input	x data, y data [frequency] [DT] *** Frequency can be omitted.
	Data deletion	x data, y data [frequency] [CL] *** Frequency can be omitted.
	Result display	Number of data (n) .... [MODE] [ALPHA] [3] (n) [EXE] Sum of x (Σx) .... [MODE] [ALPHA] [2] (Σx) [EXE] Sum of y (Σy) .... [MODE] [ALPHA] [5] (Σy) [EXE] Sum of squares of x (Σx^2) .... [MODE] [ALPHA] [1] (Σx^2) [EXE] Sum of squares of y (Σy^2) .... [MODE] [ALPHA] [4] (Σy^2) [EXE] Sum of squares of y (Σxy) .... [MODE] [ALPHA] [6] (Σxy) [EXE] Mean of x (x) .... [SHIFT] [$\bar{x}$] [EXE] Mean of y (y) .... [SHIFT] [y] [EXE] Population standard deviation of x (xon) .... [SHIFT] [xon] [EXE] Population standard deviation of y (xon) .... [SHIFT] [yon] [EXE] Sample standard deviation of x (xon-1) .... [SHIFT] [xon-1] [EXE] Sample standard deviation of y (yon-1) .... [SHIFT] [yon-1] [EXE] Constant term of regression formula (A) .... [SHIFT] [A] [EXE] Regression coefficient (B) .... [SHIFT] [B] [EXE] Correlation coefficient (r) .... [SHIFT] [r] [EXE] Estimated value of x (x) .... [SHIFT] [$\hat{x}$] [EXE] Estimated value of y (y) .... [SHIFT] [y] [EXE]
Special functions	Ans	The latest result obtained in manual or program calculations are stored in memory. It is recalled by pressing Ans. *** Mantissa of numeric value is digits.
	Replay	● After calculation results are obtained, the formula can be recalled by pressing either ◁ or ▷ . ● The replay function is not cleared even when [AC] is pressed or when power is turned off. ● If an error is generated, pressing either ◁ or ▷ will cancel the error and the point where the error was generated will be indicated by a blinking cursor.
	Multi-statement	Colons are used to join a series of statements or calculation formulas. If joined using " : " the calculation result to that point is displayed.
	Memory	The number of memories can be expanded from the standard Memories can be expanded in units of one up to (for a total of 76). Eight stepsare required for one memory. [MODE] [·] number of memories [EXE].
Graph function	Range	Graph range settings Xmin .... Minimum value of x Xmax ...... Maximum value of x Xscl .... Scale of X-axis (space between points) Ymin .... Minimum value of y Ymax ...... Maximum value of y Yscl .... Scale of Y-axis (space between points)
	Trace	Moves pointer on graph. Current coordinate location is displayed.
	Plot	Marks pointer (blinking dot) at any coordinate on the graph display.
	Line	Connects with a straight line two points created with plot function.
	Factor	Defines factor for zoom in / zoom out.
	Zoom	Zoomx f ..... Zooms in on the graph in accordance with the zoom factors
		Zoomx 1/f ... Zooms out on the graph in accordance with the inverse of the zoom factor. Zoom Org ... Returns zoomed graph to original dimensions
	Scroll	Scrolls screen to view parts of graphs that are off the display.

2. Program calculations

Program input	Input mode	WRT mode (MODE 2)
	Calculation mode	Mode that conform with program specified by: MODE + , MODE − , MODE × , MODE ÷ .
	Program area specification ·	Cursor is moved to the desired program area name (PO through P9) using ◁ and ▷ , and EXE is pressed.
Program execution	Execution mode	RUN mode (MODE 1)
	Program area specification	Execution starts with Prog program area name EXE Program area name: P0 through P9
	Input mode	WRT mode (MODE 2)
	Program area specification	Cursor is moved to the desired program area name (PO through P9) using ◁ and ▷ , and EXE is pressed.
	Editing	Cursor is moved to position to be edited using ◁ or ▷ . ● Press correct key for corrections. ● Press DEL for deletions. ● Press SHIFT INS to specify insert mode for insertion.
Program delete	Clear mode	PCL mode (MODE 3)
	Deletes specific program	Cursor is moved to the desired program area name (PO through P9) using ◁ and ▷ , and AC is pressed.
	Clears all programs	Press SHIFT Mcl.
Program commands	Unconditional jump	Program execution jumps to the Lbl n which corresponds to Goto n. *** n= 0 through 9
	Conditional jumps	If conditional expression is true, the statement after "⇨" is executed. If not true, ececution jumps to the statement following next ";" or "◢".  F : Formula R : Relational operator S : Statement *** The relational operator is: =, ≠ , >, <, ≥, ≤.
	Count jumps	The value in a memory is increased or decreased. If the value does not equal 0, the next statement is executed, If it is 0, a jump is performed to the statement following the next ";" or "◢".  S : Statement V : Value in memory
	Sub-program	A display of Prog n (n=0 to 9) will appear when a sub-program is executed from a main program. The execution of program will be located to the position after the fundamental Prog n when the sub-program finishes.

(C) Error Message Table

Message	Meaning	Countermeasure
Syn ERROR	1. Calculation formula contains an error. 2. Formula in a program contains an error.	1. Use ◁ or to ▷ to display the point where the error was generated and correct it. 2. Usr ◁ or ▷ to display the point where the error was generated, press AC and then correct the program in the WRT mode.
Ma ERROR	1. Calculation result exceeds calculation range. 2. Calculation is performed outside the input range of a function. 3. Illogical operation (division by zero, etc.)	1. Check the input numeric value and correct it. 2. When using memories, check that the numeric values stored in memories are correct.
Go ERROR	1. No corresponding Lbl n for Goto n. 2. No program stored in program area P n which corresponds to Prog n.	1. correctly input a Lbl n to correspond to the Goto n if not required. 2. Store a program in program area P n to correspond to Prog n, or delete the Prog n if not required.
Nc ERROR	• Nesting of subroutines by Prog n exceeds 10 levels.	• Ensure that Prog n is not used to return from subroutines to main routine. If used, delete any unnecessary Progn.. • Trace the subroutine jump destinations and ensure that no iumps are made back to the original program area. Ensure that returns are made correctly.
Srk ERROR	• Execution of calculations that exceed the capacity of the stack for numeric values or stack for calculations.	• Simplify the formulas to keep stacks within 10 levels for the numeric values and 24 levels for the calculations. • Divide the formula into two or more parts.
Mem ERROR	• Memory expansion exceeds level remaining in program • Attempt to use a memory such as Z[5] when no memory has been expanded.	• Press MODE . (Defm) to expand memory to necessary level. • Use memories within the current number of memories.
Arg ERROR	• Argument input incorrectly. Ex. Negative valuc input for Defm, value other that 1~9 input for n, etc.	• Re- enter argument correctly.

(D) Input Ranges of Functions

Function	Input range	Internal digits	Accuracy	Notes
sinx cosx tanx	(Deg) $\lvert x \rvert < 9\times10^{9\circ}$ (Rad) $\lvert x \rvert < 5\times10^{7}\pi$ rad (Gra) $\lvert x \rvert < 1\times10^{10}$grad	12 digits	As a rule, accuracy is ±1 at the 10th digits.	However, for tanx: $\lvert x \rvert \neq 90(2n+1)$:Deg $\lvert x \rvert \neq \pi/2\ (2n+1)$:Rad $\lvert x \rvert \neq 100(2n+1)$:Gra
$Sin^{-1}x$ $cos^{-1}x$	$\lvert x \rvert \leqq 1$	"	"	
$tan^{-1}x$	$\lvert x \rvert < 1\times10^{100}$			
sinhx coshx	$\lvert x \rvert \leqq 230.2585092$	"	"	Note: For sinh and tanh, when x=0, errors are cumulative and accuracy is affected at a certain point.
tanhx	$\lvert x \rvert < 1\times10^{100}$			
$sinh^{-1}x$	$\lvert x \rvert < 5\times10^{99}$	"	"	
$cosh^{-1}x$	$1 \leqq x < 5\times10^{99}$			
$tanh^{-1}x$	$\lvert x \rvert < 1$			
logx lnx	$1\times10^{-99} \leqq x < 1\times10^{100}$	"	"	
10^x	$-1\times10^{100} < x < 100$	"	"	
e^x	$-1\times10^{100} < x < 230.2585092$	"	"	
$\sqrt{x}$	$0 \leqq x < 1\times10^{100}$	"	"	
x^2	$\lvert x \rvert < 1\times10^{50}$			
$1/x$	$\lvert x \rvert \leqq 1\times10^{100}$, $x \neq 0$	"	"	
$\sqrt[3]{x}$	$\lvert x \rvert \leqq 1\times10^{100}$			
x!	$0 \leqq x \leqq 69$ (x is an integer)	"	"	
Pol(x,y)	$\sqrt{x^2+y^2} < 1\times10^{100}$	"	"	
Rec(r,\θ)	$0 \leqq r < 1\times10^{100}$ (Deg) $\lvert \theta \rvert < 9\times10^{9\circ}$ (Rad) $\lvert \theta \rvert < 5\times10^{7}\pi$ rad (Gra) $\lvert \theta \rvert < 1\times10^{10}$grad	"	"	However, for tanθ: $\lvert \theta \rvert \neq 90(2n+1)$:Deg $\lvert \theta \rvert \neq \pi/2(2n+1)$:Rad $\lvert \theta \rvert \neq 100(2n+1)$:Gra
° ' "	$\lvert a \rvert$, b, c $< 1\times10^{100}$ $0 \leqq$ b, c	"	"	
← ° ' "	$\lvert x \rvert < 2.777777777\times10^{96}$ Hexadecimal display: $\lvert x \rvert \leqq 2777777.777$			
x^y	x>0: $-1\times10^{100} < y\log x < 100$ x=0: y>0 x<0: y=n ,1/(2n+1) (n is an integer) However; $-1\times10^{100} < 1/y \log\lvert x \rvert < 100$	"	"	
$\sqrt[x]{y}$	y>0: $x \neq 0$ $-1\times10^{100} < 1/x \log y < 100$ y=0: x>0 y<0: x=2n+1, 1/n ($n \neq 0$, n is an integer) However; $-1\times10^{100} < 1/x \log\lvert y \rvert < 100$	"	"	

Function	Input range	Internal digits	Accuracy	Notes
a *b/c*	Results Total of integer, numerator and denominator must be within 10 digits(includes division marks). Input Result displayed as a fraction for integer when integer, numerator and denominator are less than 1×10^{10}.	12 digits	As a rule, accuracy is ±1 at the 10th digits.	
SD(LR)	$\lvert x \rvert < 1\times10^{50}$ $\lvert y \rvert < 1\times10^{50}$ $\lvert n \rvert < 1\times10^{100}$ xσn, yσn, $\bar{x}$, $\bar{y}$, A, B, r: $n \neq 0$ xσn-1, yσn-1: $n \neq 0$, 1	"	"	

Function	Input range
BASE-N	Values after variable within following range: Dec: -2147483648 ≤ x ≤ 2147483647 Bin: 1000000000 ≤ x ≤1111111111 (negative) 0 ≤ x ≤ 0111111111 (0, positive) Oct: 2000000000 ≤ x ≤ 37777777777 (negative) 0 ≤ x ≤ 1777777777 (0, positive) Hex: 80000000 ≤ x ≤ FFFFFFFF (negative) 0 ≤ x ≤ 7FFFFFFF (0, positive)

Errors may be cumulative with internal continuous calculations such as x^y, $\sqrt[x]{y}$, x!, $\sqrt[3]{x}$ sometimes affecting accuracy.

(E) Specifications

Graph functions

Built-in function graphs: (20 types) sin, cos, tan, $\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$, sinh, cosh, tanh, $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$, log, In 10^x, e^x, x^2, $\sqrt{\ }$, $\sqrt[3]{\ }$, x^{-1}.

Types of graphs: User generated function graphs
Rectangular coordinates
Single-variable statistics; bar graphs, normal distribution curves
Paired-variable statistics; regression lines

Graph functions: Range specification, Overlay, Trace, Zoom (xf, x1/f, factor, original (resume)), Plot, Line, Scroll

Calculations

Basic calculation functions: Negative numbers, exponents, parenthetical addition / subtraction/ multiplication/ division (with priority sequence judgement function -- true algebraic logic)

Built-in scientific function: trigonometric/inverse trigonometric functions(units of angular measurement: degrees, radians, grads), hyperbolic/inverse hyperbolic functions, logarithmic/exponential functions, reciprocals, factorials, square roots, cube roots, powers, roots, squares, decimal-sexagesimal conversions, binary-octal-hexadecimal calculations, coordinate transformations

Statistics: standard deviation - number of data, sum, sum of squares, means, standard deviation(two types)
Linear regression - number of data, sum of x, sum of y, sum of squares of x, sum of squares of y, mean of x, mean of y, standard deviation of x (two types), standard deviation of y (two types), constant term, regression coefficient, correlation coefficient, estmated value of x, estimated value of y.

Special functions: insert, delete, replay functions, substitution(=), multi-statement (: and ◢)

Memories: 26 standard (maximum 76), Ans memory

Calculation range: $\pm 1\times10^{-99} \sim 9.999999999\times10^{99}$ and 0. Internal operation uses 12 digit mantissa

Rounding: performed according to the specified number of significant digits or the number of specified decimal places.

Exponential display: Norm1 $-10^{-2} > |x|, \; |x| \geq 10^{10}$ Norm2 $-10^{-9} > |x|, \; |x| \geq 10^{10}$

Program function

Number of steps: 400 maximum

Jump functions: unconditional jump (Goto), 10 maximum
Conditional jump (=, ≠, <, >, ≦, ≧)
Count jumps (Isz, Dsz)

Subroutines: 9 levels

Number of stored ptograms: 10 maximum (P0 to P9)

Check functions: program checking , debugging, deletion, addition, insertion, etc.

General:

Power supply: one lithium battery (CR2032)

Battery life: approximately 300 hours

Ambient temperature range : 32°F - 105°F

Dimensions: 5.9" x 3" x 0.4"

- The contents of this manual are subject to change without notice.
- In no event will the manufacturer and its suppliers be liable to you or any other damages arising from the use of or inability to use this calculator or manual.
- In no event will the manufacturer and its suppliers be liable to you or any other person for any damages, expenses, lost profits, lost savings or any other damages arising from the loss of data and/or formulas caused by use of this calculator or manual.
- Due to limitations imposed by printing processes, the displays shown in this manual are only approximations and may differ somewhat from actual displays.

LIMITED WARRANTY

This item is warranted to be free from defects in materials or workmanship for a period of one year from the original date of purchase under normal conditions of household use. Within the warranty period, we will repair / replace the product or refund your money, at our option, at the place of purchase upon presentation of sales receipt or other proof of purchase. This warranty extends only to the original retail purchaser, and excludes any damage to the product resulting from accident or misuse. The product is not warranted when used in a commercial or business environment.

The above warranty is complete and exclusive. The warrantor expressly disclaims liability for any special, incidental, indirect, or consequential damages in connection with the pruchase or use of this product or costs over the original purchase price.

Any implied warranties arising by operation of law shall be limited in duration to the term of this warranty.

This warranty gives you specific legal rights; you may also have other rights that vary by jurisdiction. This warranty is valid only within the United States of America (USA).

Item# A17C0139
WIC# 560375
SVC Code: 1974

Distributed By:
East West Distributing Co.
Deerfield, IL 60015-4616

Made in China

LIMITED WARRANTY

This item is warranted to be free from defects in materials or workmanship for a period of one year from the original date of purchase under normal conditions of household use. Within the warranty period, we will repair / replace the product or refund your money, at our option, at the place of purchase upon presentation of sales receipt or other proof of purchase. This warranty extends only to the original retail purchaser, and excludes any damage to the product resulting from accident or misuse. The product is not warranted when used in a commercial or business environment.

The above warranty is complete and exclusive. The warrantor expressly disclaims liability for any special, incidental, indirect, or consequential damages in connection with the pruchase or use of this product or costs over the original purchase price.

Any implied warranties arising by operation of law shall be limited in duration to the term of this warranty.

This warranty gives you specific legal rights; you may also have other rights that vary by jurisdiction. This warranty is valid only within the United States of America (USA).

Item# A17C0139
WIC# 560375
SVC Code: 1974

Distributed By:
East West Distributing Co.
Deerfield, IL 60015-4616

Made in China