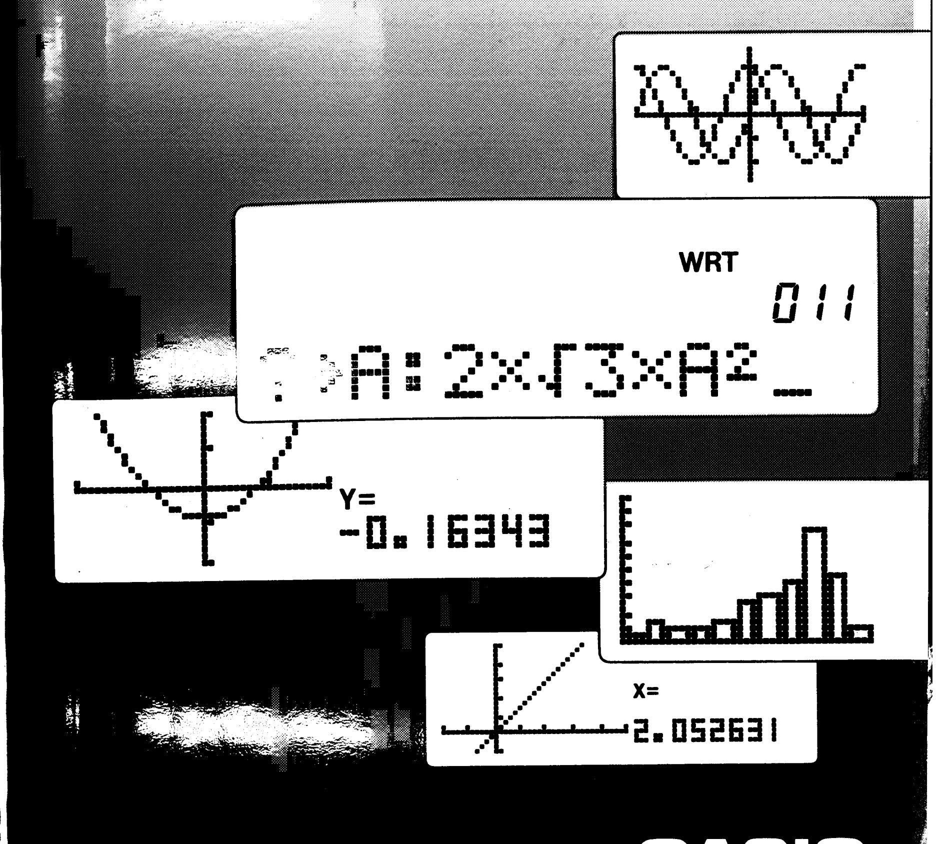
CASIO®

SA0911C ㊎ Printed in Taiwan U.S. Pat. 4,410,956
HDC00380E1 W30

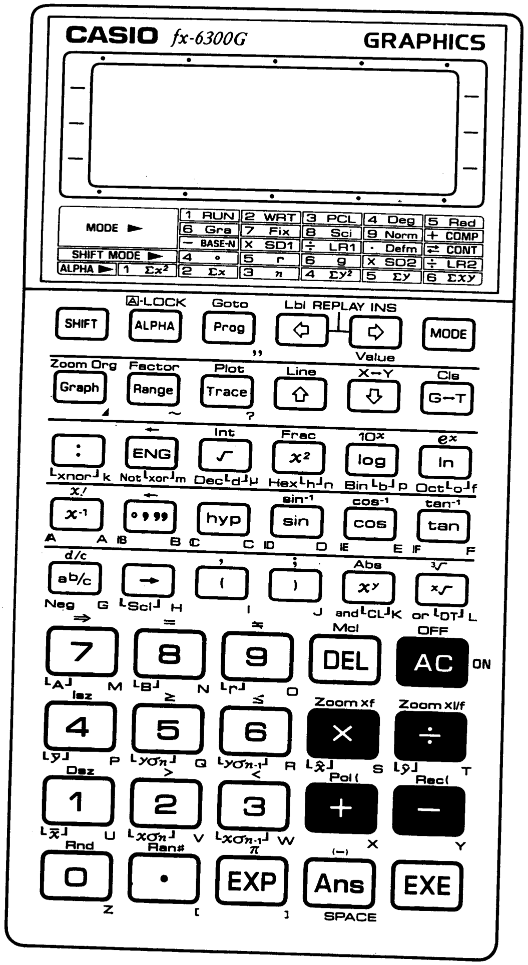
fx-6300G **Owner's manual**

CASIO®

GRAPHIC SCIENTIFIC
fx-6300G
Owner's manual

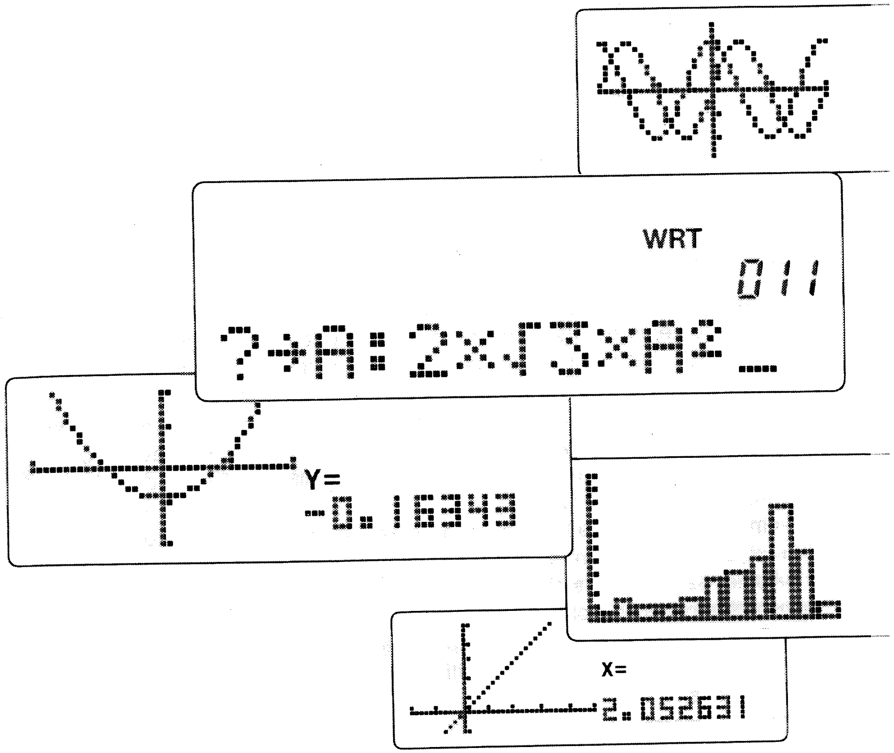


CASIO®



GRAPHIC SCIENTIFIC

fx-6300G



WRT

011

?→A:2×√3×A²_

Y=
-0.16343

X=
2.052631

- The contents of this manual are subject to change without notice.
- No part of this manual may be reproduced in any form without the express written consent of the manufacturer.
- 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 out of 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 out of 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.

Introduction

Thank you for your purchase of the CASIO fx-6300G.

This unit is a totally new type of advanced programmable calculator. Besides versatile scientific functions, graph functions also make it possible to produce a wide variety of useful graphs.

Manual calculations can be easily performed following written formulas (true algebraic logic). A replay function is provided that allows confirmation or correction when key operation errors occur. Programs can also be input by following true algebraic logic, so repeat and/or complex calculations are simplified.

All of this power built into a compact configuration that folds up to slip right into your pocket.

To use the fx-6300G to its full potential, be sure to carefully read this manual and keep it handy for future reference.

Important — Reset your calculator before using it for the first time!

See page 10 for details on the reset procedure.

Important — Always back up data!

This product features electronic memory that is capable of storing large volumes of data. You must also remember that your data is safely stored as long as power is being supplied to the memory. Data stored in memory will be irreparably damaged or lost entirely if you let battery power become too low, if you make a mistake while replacing batteries, or if power is cut off. Data can also be damaged by strong impact or electrostatic charge, or by environmental extremes. Once data is damaged or lost, it cannot be recovered, so we strongly recommend that you back up all important data.

Contents

About the Power Supply

■Replacing batteries

Precautions:

Incorrectly using batteries can cause them to burst or leak, possibly damaging the interior of the unit. Note the following precautions:

- Be sure that the positive ⊕ and negative ⊖ poles of each battery are facing in the proper direction.
- Never mix batteries of different types.
- Never mix old batteries and new ones.
- Never leave dead batteries in the battery compartment.
- Never try to recharge the batteries supplied with the unit.
- Do not expose batteries to direct heat, let them become shorted, or try to take them apart.

> Keep batteries out of the reach of small children. If swallowed, consult with a physician immediately.

Power is supplied to this unit by two lithium batteries (CR2032). If the power of the batteries should diminish, the display will weaken and become difficult to read. A weak display even after contrast adjustment (see page 29) may indicate power is too low, so the batteries should be replaced. When making replacements, be sure to replace all two batteries.

**If batteries are used for longer than 5 years, there is the danger of leakage. Be sure to replace batteries at least once every 5 years even if the unit is not used during that period.*

**The life of the original batteries supplied with the calculator is calculated from the date of installation at the factory, not from the date of purchase.*

**Stored programs or data are erased when batteries are replaced. Therefore, it is recommended that programs and data required for later use be recorded on a coding sheet before replacing batteries.*

● To replace the batteries

⟨Important⟩

- Always make sure that the power of the calculator is switched off before replacing batteries.
- Note that replacing batteries and performing the reset operation erases all data stored in the calculator. Be sure to make written copies of data before performing these operations.
- After replacing batteries, be sure to switch the calculator on and then perform the reset operation.

① After making sure that unit power is switched off, remove the 2 screws that hold the back cover of the calculator in place. Remove the back cover.

② Remove the screws that secure the battery holders in place, and remove the battery holders.

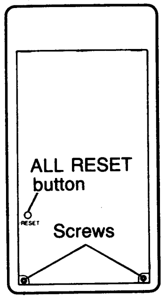
③ Remove the old batteries. If you face the battery compartment down and tap the top of the calculator, the batteries should fall out.

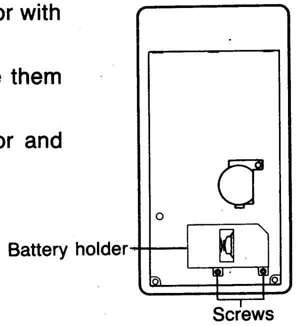
④ Wipe the surfaces of two new batteries with a soft, dry cloth, and load them into the calculator with their positive ⊕ sides facing up.

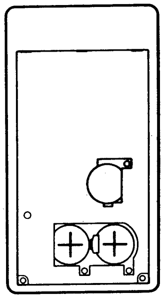
⑤ Replace the battery holders, and secure them with their screws.

⑥ Replace the back cover of the calculator and secure it in place with its 2 screws.

⑦ Switch power on.







■Auto Power Off function

The power of the unit is automatically switched off approximately 6 minutes after the last key operation (except during program calculations). Once this occurs, power can be restored by pressing the AC key. (Numeric values in the memories, programs or calculation modes are unaffected when power is switched off.)

■Reset operation

- **Strong external electrostatic charges can cause this calculator to malfunction. Should this happen, perform the following procedure to reset the calculator.**

Warning!

The following procedure clears all data from the memory of the calculator and cannot be undone! To avoid the loss of important data, be sure to always keep written backup copies.

1. Switch the power of the calculator on.
2. Press the RESET button on the back of the calculator with a thin, pointed object. A message appears on the display to confirm whether you want to reset the calculator and clear memory contents.

```
Reset ?
```

3. Press EXE to clear the calculator and clear the display. To abort the reset operation without clearing the calculator, press any key other than EXE.

Following the reset procedure described above, the calculator is initialized as follows:

1. RUN mode
2. COMP mode
3. DEG mode
4. NORM1 mode
5. Decimal mode (for BASE-N calculations)
6. Variable memories cleared
7. Defm 0 (400 program steps)
8. Answer memory clear
9. Program clear
10. Input buffer clear
11. Replay memory clear

- If the Auto Power Off function activates while the "Reset ?" message is displayed, press AC to restore power and then start again from step 1.

****Never press the RESET button while internal operations are being performed. Doing so can cause irreparable damage to the memory of your calculator.***

Handling Precautions

- Your calculator is made up of precision components. Never try to take it apart.
- Avoid dropping your calculator and subjecting it to other strong impacts.
- Do not store the calculator or leave it in areas exposed to high temperatures or humidity, or large amounts of dust. When exposed to low temperatures, the calculator may require more time to display results and may even fail to operate. Correct operation will resume once the calculator is brought back to normal temperature.
- The display will go blank and keys will not operate during calculations. When you are operating the keyboard, be sure to watch the display to make sure that all your key operations are being performed correctly.
- Replace batteries once every 5 years regardless of how much the calculator is used during that period. Never leave dead batteries in the battery compartment. They can leak and damage the unit.
- Avoid using volatile liquids such as thinner or benzine 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 a neutral detergent and wrung out.
- 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 out of loss of data and/or formulas arising out of malfunction, repairs, or battery replacement. The user should prepare physical records of data to protect against such data loss.
- Never dispose of batteries, the liquid crystal panel, or other components by burning them.
- Be sure that the power is off when replacing batteries.
- If the calculator is exposed to a strong electrostatic charge, its memory contents may be damaged or the keys may stop working. In such a case, perform the Reset operation to clear the memory and restore normal key operation.
- Note that strong vibration or impact during program execution can cause execution to stop or can damage the calculator's memory contents.
- Before assuming malfunction of the unit, be sure to carefully reread this manual and ensure that the problem is not due to insufficient battery power, programming or operational errors.

1 General Guide

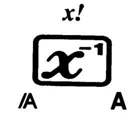
General Guide

Before using this unit for the first time, be sure to perform the Reset operation described on page 10.

Important)
The keys of a scientific calculator perform more than one function. The following explains all of the operations of each key, and so you should read this section carefully before using your calculator for the first time.

1-1 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 following, concerning the key illustrated above.

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

The keys of this calculator can perform a number of different functions. The keyboard 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 on the keyboard.

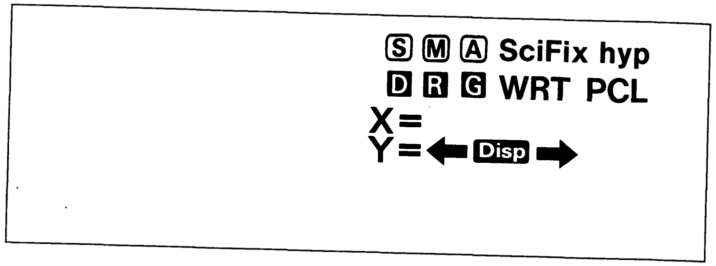
Keyboard Marking	Meaning
Yellow	Press SHIFT and then key.
Red	Press ALPHA and then key.
Green	Press key in BASE-N mode.
In blue brackets	Press key in SD or LR mode.

In addition to the above, there are a number of key sequences indicated on the panel beneath the display (such as [ALPHA ►] [1 Σx^2]). These key sequences can be used in the SD or LR mode only.

1-2 How 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.



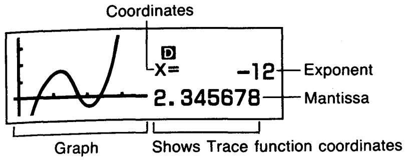
Ⓢ:	SHIFT key pressed.
Ⓜ:	MODE key pressed.
Ⓐ:	ALPHA key pressed.
Sci:	Number of significant digits specified.
Fix:	Number of decimal places specified.
hyp:	hyp key pressed.
D :	Degrees specified at the unit of angular measurement.
R :	Radians specified at the unit of angular measurement.
G :	Grads specified at 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

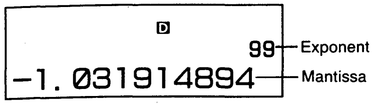
The display consists of a dot area for graphing, as well as an area for indicators and characters. You can monitor the status of the calculator and programs by viewing the display.

Example

Graph Display

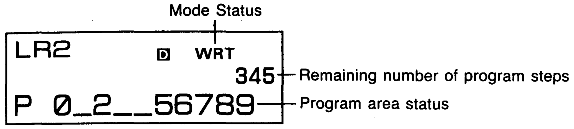


Calculation Display



Mode Status Display

Example **Program WRT mode**



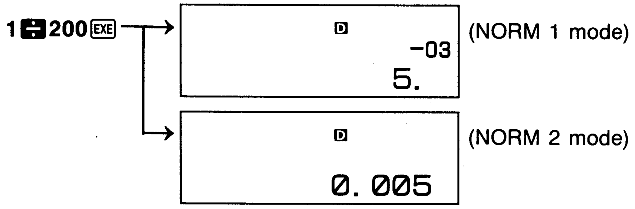
■Exponential display

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

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

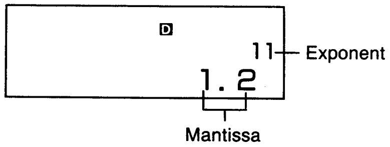
NORM 2 mode: $10^{-9}(0.000000001) > |x|, |x| \geqq 10^{10}$

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:



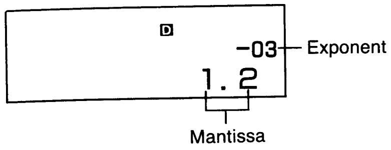
(All of the examples in this manual show calculation results using the NORM 1 mode.)

How to interpret exponential format



➡ 1.2×10^{11} ➡ 120,000,000,000

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.



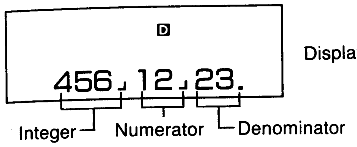
➡ 1.2×10^{-3} ➡ 0.0012

1.2^{-03} indicates that the result is equivalent to 1.2×10^{-3}. 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

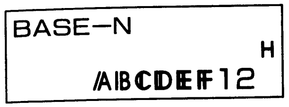
Special display formats are used for the representation of fraction, hexadecimal, and sexagesimal values.

● Fraction value display



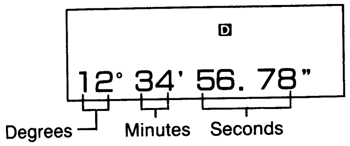
Display of $456\frac{12}{23}$

● Hexadecimal value display



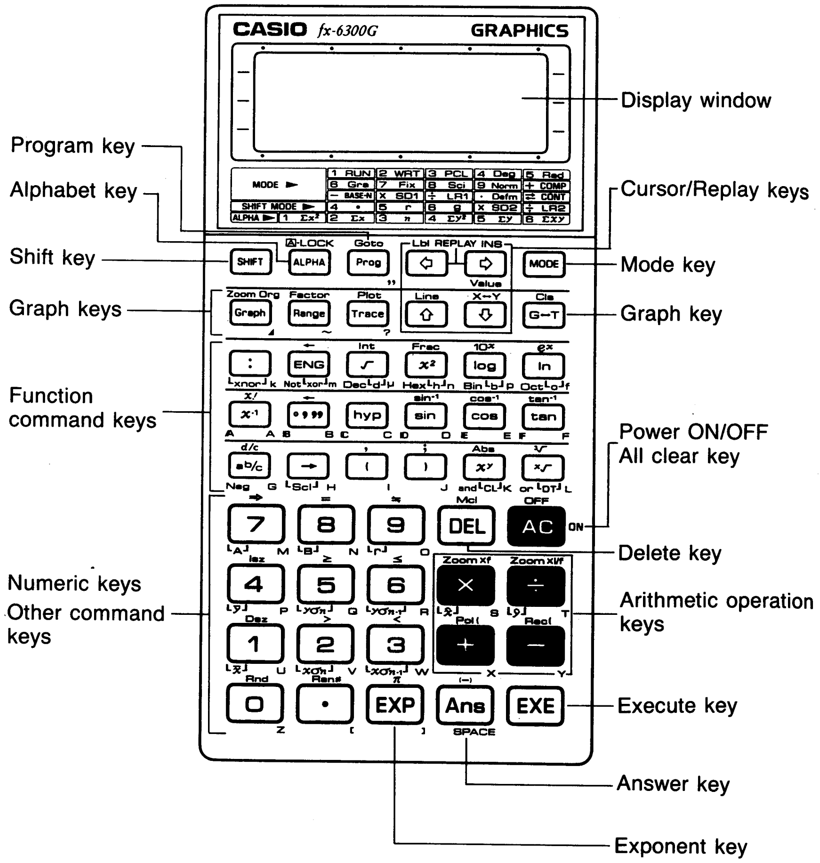
Display of ABCDEF12 (−1412567278)

● Sexagesimal value display



Display of 12°34'56.78"

1-3 Key Layout



1-4 Key Operations

■Special operation keys

[SHIFT] Shift key

Press when using the function commands and functions marked in yellow on the key panel. An [S] will blink on the display to indicate that [SHIFT] has been pressed. Pressing [SHIFT] again will cause the [S] to disappear from the display and the unit to return to the status it was in before [SHIFT] was originally pressed.

[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 [EXE] will cancel the specified number of decimal places or 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][EXE] and then change the range of the exponential display. (NORM 1/NORM 2) (see page 18.)*

**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.

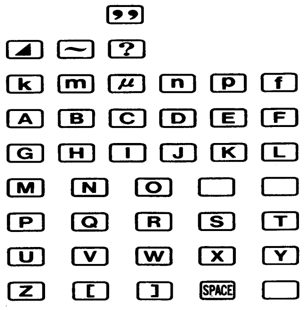
If EXE is pressed without entering a value, the current number of memories available and remaining steps will be displayed (see page 40.)

Example MODE · EXE   M-26 S-320

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 (SD1 mode).
MODE ÷ ... For regression calculations (LR1 mode).
SHIFT MODE × ... For production of a bar graph or normal distribution curve according to single variable statistical data (SD2 mode).
SHIFT MODE ÷ ... For production of a regression line according to paired variable statistical data (LR2 mode).
The x^y and $\sqrt[x]{\ }$ functions are not available in the LR1 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 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.

REPLAY
Lbl INS
⇦ ⇨
Value
Line X↔Y
⇧ ⇩

Cursor/Replay keys

The ⇦ key moves the cursor (blinking "_") 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 or regression lines.
After you draw a graph, press SHIFT Value to display a 7-digit (including the negative sign) 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. Pressing SHIFT Value changes the coordinate display to 11 digits (including the negative sign) for even more precision.

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
AC ON 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 (page 113).
Press to switch the power of the calculator on (even if power was switched off by the Auto Power Off function).
Pressing SHIFT OFF switches the power of the calculator 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 SPACE Answer/Minus key

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 of the unit off. When used during program execution, the last result calculated is 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 0 Z ~ ≠ 9 └r┘ O Ran# • [π EXP]

When entering numeric values, enter the number in order. Press the • key to enter the decimal point in the desired position.
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 +)

⇒ = ≠
Isz ≥ ≤
Dsz > <
Rnd Ran# π

BASE-N mode (MODE −)

⇒ = ≠
Isz ≥ ≤
Dsz > <
□ □

Pol(, Rec(, Rnd, Ran# and π cannot be used in this mode.

SD mode (MODE ×)

⇒ = ≠
Isz ≥ ≤
$\bar{x}$ $x\sigma_n$ $x\sigma_{n-1}$
Rnd Ran# π

Standard deviation functions can be used.

LR mode (MODE ÷)

A B r
$\bar{y}$ $y\sigma_n$ $y\sigma_{n-1}$
$\bar{x}$ $x\sigma_n$ $x\sigma_{n-1}$
Rnd Ran# π

Paired variable statistic functions can be used.

■Calculation keys

Pol(+ X Rec(− Y Zoom×f × └$\hat{x}$┘ S Zoom×1/f ÷ └$\hat{y}$┘ T Arithmetic operation keys

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

COMP mode

Zoom×f, Zoom×1/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

$\hat{x}$ $\hat{y}$ ... Estimated value calculation of x and y
Pol(Rec(... Coordinate transformation

■Graph keys

Used to produce a variety of graphs (see page 76 for details). 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 (see page 36).
- 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, and/or depending on the mode being used.

: Multistatement key

└xnor┘ k

- Press to separate formulas or commands in programmed calculations or consecutive calculations.
  The result of such combinations is known as a multistatement (see page 53).
- Press following SHIFT in the BASE-N mode to enter the logical operation for negation of logical sums (xnor).

←

ENG Engineering/Negation key

Not└xor┘m

- Press to convert a calculation result to an exponential display whose exponent is a multiple of three.
  $(10^3 = \overset{\text{kilo}}{k},\ 10^6 = \overset{\text{mega}}{M},\ 10^9 = \overset{\text{giga}}{G},\ 10^{-3} = \overset{\text{milli}}{m},\ 10^{-6} = \overset{\text{micro}}{\mu},\ 10^{-9} = \overset{\text{nano}}{n},\ 10^{-12} = \overset{\text{pico}}{p})$
- 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└d┘μ

- Press prior to entering a numeric value 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└h┘n

- Press after a numeric value is entered 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└b┘p

- Press prior to entering a value 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

ln Natural logarithm/Exponential key

Oct└o┘f

- Press prior to entering a value to obtain the natural 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 speciiied as an octal value.

$x!$

x^{-1} Reciprocal/Factorial key

/A A

- Press after entering a value 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)

/B 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

C C

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

$\sin^{-1}$ $\cos^{-1}$ $\tan^{-1}$

sin cos tan Trigonometric function/Inverse trigonometric function keys

D D E E F F

- 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.

d/c

a b/c Fraction/Negative key

Neg G

- Use this key for input of simple fractions and mixed fractions.
  Ex. To input 23/45: 23 a b/c 45
  To input 2-3/4: 2 a b/c 3 a b/c 4
- For improper fractions, press this key following SHIFT (indicated by SHIFT 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

LScl⅃ H

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

I J

- 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.

Abs

x^y Power/Absolute value key

and LCL⅃K

- 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.

$\sqrt[3]{\ }$

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

or LDT⅃L

- 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 adjustment

Pressing the ⇦ or ⇨ key following the MODE key adjusts the contrast of the display. Pressing ⇦ makes the screen lighter, while ⇨ makes it darker. Holding either key down will cause the display to successively become respectively lighter or darker.

Pressing any other key besides MODE, ⇦, or ⇨ (as well as ⇧, ⇩) cancels contrast adjustment.

**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 batteries as soon as possible. After replacing batteries, perform the RESET operation described on page 10.*

**Contrast adjustment is impossible during range display using the Range key or during factor display using the Factor key (see page 78, 84).*

1-5 Before Beginning Calculations....

■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}, $x!$, °, ^{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
⑪ or, xor, xnor] BASE-N calculations only.

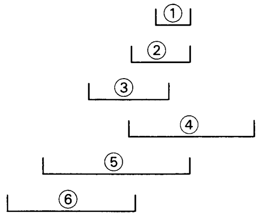
⑫ Relational operators <, >, =, ≠, ≤, ≥

**When functions with the same priority are used in series, execution is performed from right to left.*
$e^x \ln\sqrt{120} \rightarrow e^x\{\ln(\sqrt{120})\}$
Otherwise, execution is from left to right.

**Compound functions are executed from right to left.*

**Anything contained within parentheses receives highest priority.*

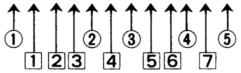
Example 2 + 3 × (log sin2π^2 + 6.8) = 22.07101691 (in the “Rad” mode)



■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 on the display.

Example 2 × ((3 + 4 × (5 + 4) ÷ 3) ÷ 5) + 8 =



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 the stack.*

■Calculation modes

This unit features modes for manual calculations, storing programs, and modes for general as well as statistical calculations. The proper mode to suit calculational requirements should be employed.

• 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. (See Section 4.)
3. **PCL mode**
   Deletion of stored programs. (See Section 4.)

• Calculation modes

There are a total of six 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. (See page 62.) Function calculations and graph drawing cannot be performed.
3. **SD1 mode**
   Standard deviation calculation (single-variable statistics). (See page 67.)
4. **SD2 mode**
   For production of bar graph or normal distribution curve according to single-variable statistical data. (See page 97.)
5. **LR1 mode**
   Regression calculation (paired-variable statistics). (See page 69.)
6. **LR2 mode**
   For production of regression line graph according to paired-variable statistical data. (See page 100.)

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

**IMPORTANT: 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 a 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 − 4 2 8 5 7 EXE → 0. 1428571

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

Example 123456789 × 9638 =

1 2 3 4 5 6 7 8 9
× 9 6 3 8 EXE

```
               12 — Exponent
1. 189876532 ——— Mantissa
```

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

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

3 EXP 5 ÷ 7 EXE → 42857. 14286

− 4 2 8 5 7 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 impossible. 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. (See page 181.)

(3) Improper operation during statistical calculations.
(Ex. Attempting to obtain $\bar{x}$ or $x\sigma_n$ 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+3×4EXE)

(5) Even though memory has not been expanded, a memory name such as Z [2] is used. (See page 41 for details on memory.)

(6) Input errors are made.
(Ex. 5××3EXE)

(7) When improper arguments are used in commands or functions that require arguments. (i.e. 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 are an "Ne ERROR" (nesting error) and a "Go ERROR". These errors mainly occur when using programs. See page 113 or the Error Message Table on page 179.

■Number of input characters

This unit features a 127-step area for calculation execution.
One function comprises one step. Each press of numeric or +, −, × and ÷ keys comprise one step. Though such operations as SHIFT x! require two key 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 of each other.

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

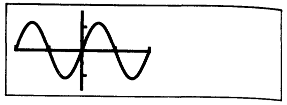
[Graph] [sin] [ALPHA] [X]

(Text display)

```
sin X_
```

[EXE]

(Graphic display)



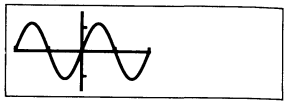
[G↔T]

(Text display)

```
                done
```

[G↔T]

(Graphic display)



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 press the correct keys.

Example To change an input of 122 to 123:

[1] [2] [2]	122_
[⇦]	122
[3]	123_

Example To change an input of cos60 to sin60:

[cos] [6] [0]	cos 60_
[⇦] [⇦] [⇦]	cos 60
[sin]	sin 60

**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).

Example To correct an input of 369 × × 2 to 369 × 2:

[3] [6] [9] [×] [×] [2]	369××2_
[⇦] [⇦] [DEL]	369×2

•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.

Example **To correct an input of 2.36² to sin2.36²:**

2 • 3 6 x²
```
2. 36²_
```
⇦ ⇦ ⇦ ⇦ ⇦
```
2. 36²
```
SHIFT INS
```
⌜2⌟. 36²
```
sin
```
sin ⌜2⌟. 36²
```

**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 perform* 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:**

1 2 3 • 4 5 → 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 + 7 8 • 9 → ALPHA B
EXE
```
A+78. 9→B_
        202. 35
```

Example **To add 74.12 to memory B:**

ALPHA B + 7 4 • 1 2 → 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 memories:**

SHIFT Mcl
EXE
```
McI_
             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:**

1 0 → 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.

**See page 116 for information on the number of program steps.*

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 · 3 0

```
Defm 30_
```

EXE

```
M-56   S-160
```

Number of memories   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 a 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 of memories has been expanded by 5, memories Z[1] through Z[5] are available. The use of these memories is similar to that of a standard computer array, with a subscript being appended to the name. For more information concerning an array, see page 131.

2 Manual Calculations

Manual Calculations

2-1 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	Operation	Display
$23+4.5-53=-25.5$	23 + 4.5 − 53 EXE	−25.5
$56\times(-12)\div(-2.5)=268.8$	56 × SHIFT (−) 12 ÷ SHIFT (−) 2.5 EXE	268.8
$12369\times7532\times74103=$ 6.903680613×10^{12} (6903680613000)	12369 × 7532 × 74103 EXE	12 6.903680613
*Results greater than 10^{10} (10 billion) or less than10^{-2} (0.01) are displayed in exponential form.		
$(4.5\times10^{75})\times(-2.3\times10^{-79})$ $=-1.035\times10^{-3}$ (-0.001035)	4.5 EXP 75 × SHIFT (−) 2.3 EXP SHIFT (−) 79 EXE	−03 −1.035 (NORM 1)
$(2+3)\times10^2=500$	(2 + 3) × 1 EXP 2 EXE	500.
*The correct result cannot be derived by entering (2 + 3) EXP 2. Be sure to enter × 1 between the) and EXP in the above example.		
$(1\times10^5)\div7=14285.71429$	1 EXP 5 ÷ 7 EXE	14285.71429
$(1\times10^5)\div7-14285$ $=0.7142857$	1 EXP 5 ÷ 7 − 14285 EXE	0.7142857
*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.		

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

Example	Operation	Display
$3+5\times6=33$	3 + 5 × 6 EXE	33.
$7\times8-4\times5=36$	7 × 8 − 4 × 5 EXE	36.
$1+2-\underline{3\times4\div5}+6=6.6$	1 + 2 − 3 × 4 ÷ 5 + 6 EXE	6.6

■Parenthesis calculations

Example	Operation	Display
$100-(2+3)\times4=80$	100 − (2 + 3) × 4 EXE	80.
$2+3\times(4+5)=29$	2 + 3 × (4 + 5 EXE	29.
*Closed parentheses occurring immediately before operation of the EXE key may be omitted, no matter how many are required.		
$(7-2)\times(8+5)=65$	(7 − 2) (8 + 5 EXE	65.
*A multiplication sign × occurring immediately before an open parenthesis can be omitted.		
$10-\{2+7\times(3+6)\}=-55$	10 − (2 + 7 (3 + 6 EXE	−55.
*Henceforth, abbreviated style will not be used in this manual.		
$\frac{2\times3+4}{5}=(2\times3+4)\div5=2$	(2 × 3 + 4) ÷ 5 EXE	2.
$\frac{5\times6+6\times8}{15\times4+12\times3}=0.8125$	(5 × 6 + 6 × 8) ÷ (15 × 4 + 12 × 3) EXE	0.8125
$(1.2\times10^{19})-\{(2.5\times10^{20})$ $\times\frac{3}{100}\}=4.5\times10^{18}$	1.2 EXP 19 − (2.5 EXP 20 × 3 ÷ 100) EXE	18 4.5
$\frac{6}{4\times5}=0.3$	6 ÷ (4 × 5) EXE	0.3
*The above is the same as 6 ÷ 4 ÷ 5 EXE.		

■Memory calculations

- The contents of memories are not erased when power is off. They are cleared by pressing SHIFT followed by Mcl and then EXE.

Example	Operation	Display
	9.874 → ALPHA A EXE	9.874
9.874 × 7 = 69.118	ALPHA A × 7 EXE	69.118
9.874 × 12 = 118.488	ALPHA A × 12 EXE	118.488
9.874 × 26 = 256.724	ALPHA A × 26 EXE	256.724
9.874 × 29 = 286.346	ALPHA A × 29 EXE	286.346
*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 automatically replaced with the new value.)		
23 + 9 = 32	23 + 9 → ALPHA B EXE	32.
53 − 6 = 47	53 − 6 EXE	47.
−) 45 × 2 = 90	ALPHA B + Ans → ALPHA B EXE	79.
99 ÷ 3 = 33	45 × 2 EXE	90.
Total 22	ALPHA B − Ans → ALPHA B EXE	−11.
	99 ÷ 3 EXE	33.
	ALPHA B + Ans → ALPHA B EXE	22.
12 × (2.3 + 3.4) − 5 = 63.4	2.3 + 3.4 → ALPHA G EXE	5.7
	12 × ALPHA G − 5 EXE	63.4
30 × (2.3 + 3.4 + 4.5) − 15	4.5 → ALPHA H EXE	4.5
× 4.5 = 238.5	30 × (ALPHA G + ALPHA H)	
	− 15 ALPHA H EXE	238.5
*Multiplication signs (×) immediately before memory names can be omitted.		

■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.
- Pressing 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 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	Operation	Display
100 ÷ 6 = 16.66666666...	100 ÷ 6 EXE	16.66666667
	(Four decimal places specified.) MODE 7 4 EXE	Fix 16.6667
	(Specification cancelled.) MODE 9 EXE	16.66666667
	(Five significant digits specified.) MODE 8 5 EXE	Sci 01 1.6667
	(Specification cancelled.) MODE 9 EXE	16.66666667
*Values are displayed rounded off to the place specified.		
200 ÷ 7 × 14 = 400	(Three decimal places specified.) MODE 7 3 EXE	16.667
	(Continues calculation with 10 digits display.) 200 ÷ 7 EXE	Fix 28.571
	×	← Fix 8.57142857 ×_
	14 EXE	Fix 400.000

Example	Operation	Display
	If the same calculation is performed with the specified number of digits:	
	200 ÷ 7 EXE	Fix 28.571
(Value stored internally cut off at specified decimal place.)	SHIFT Rnd EXE	Fix 28.571
	×	Fix 28.571 × _
	14 EXE	Fix 399.994
	(Specification cancelled.) MODE 9 EXE	399.994
123m × 456 = 56088m	123 × 456 EXE	56088.
= 56.088km	ENG	03 56.088
78g × 0.96 = 74.88g	78 × 0.96 EXE	74.88
= 0.07488kg	SHIFT ENG	03 0.07488

2-2 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.

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

Example **123 + 456 = <u>579</u>**
789 − <u>579</u> = 210

Operation	Display
AC 1 2 3 + 4 5 6 EXE	579.
7 8 9 − 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 of the unit 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:**

Operation	Display
7 8 + 5 6 EXE	134.
Ans EXE ... Checking the content of Ans memory	134.
1 2 3 → ALPHA A EXE	123.
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 [×] immediately before [Ans] can be omitted.

Example **15 × 3 = 45**
78 × 45 − 23 = 3487

[AC][1][5][×][3][EXE]	45.
[7][8][Ans][−][2][3][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-digit mantissa of the displayed value.

Example **To calculate ÷ 3.14 after 3 × 4 = 12:**

[AC][3][×][4][EXE]	12.
(Continuing) [÷][3][•][1][4]	12. ÷3. 14_
[EXE]	3. 821656051

Example **To calculate 1 ÷ 3 × 3 = :**

[AC][1][÷][3][×][3][EXE]	1.
[1][÷][3][EXE]	0. 3333333333
(Continuing) [×][3][EXE]	0. 9999999999

This function can be used with memory and Type A functions (x^2, x^{-1}, $x!$, °' ", °, r, g, see page 30), +, −, x^y, and $\sqrt[x]{\ }$.

Example **To store the result of 12 × 45 in memory C:**

[1][2][×][4][5][EXE]	540.
(Continuing) [→][ALPHA][C]	540. →C_
[EXE]	540.

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

[AC][7][8][÷][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.

Example

[AC][1][2][3][×][4][5][6][EXE]	56088.
[⇨]	123×456
	**The formula appears after clearing the display.*
[EXE]	56088.
[⇦]	123×456_

**As with the number of input steps (see page 35), the replay function can accept input of up to 127 steps.*

**The replay function is not cleared even when [AC] is pressed or when power is turned off, so contents can be recalled even after [AC] is pressed.*

Example

[AC][1][2][3][×][4][5][6]	123×456_
[EXE]	56088.
[AC]	_
[⇦]	123×456_

**Replay function is cleared when mode or operation is switched.*

■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 values and formulas from the beginning.

Example **14 ÷ 0 × 2.3 mistakenly input instead of 14 ÷ 10 × 2.3:**

[AC][1][4][÷][0][×][2][•][3][EXE]	Ma ERROR
[⇨] (or [⇦])	14÷0×2. 3 Cursor indicates where error is generated
[⇦][SHIFT][INS][1]	14÷10×2. 3
[EXE]	3. 22

■Multistatement function

- The multistatement function available in program calculations can also be used in manual calculations.
- With the multistatement function, multiple statements are linked together with a colon ([:]) separating them.
- Pressing the [EXE] key after a multistatement 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 $\mathbf{6.9 \times \underline{123} = 848.7}$
$\mathbf{\underline{123} \div 3.2 = 38.4375}$

[1][2][3][→][ALPHA][A][:] [6][•][9][×][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 multistatement is always displayed, regardless of whether a "◢" symbol is input at the end of the last statement in the chain.*

**Consecutive calculations contained in multistatements cannot be performed.*

123 × 456 $\underset{\sim\sim}{:\ \times}$5
└── Invalid

2-3 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	Operation	Display
Conversion of 4.25 rad to degrees	MODE 4 EXE 4.25 SHIFT MODE 5 EXE	243.5070629
Conversion of 1.23 grad to radians	MODE 5 EXE 1.23 SHIFT MODE 6 EXE	0.0193207948
Conversion of 7.89 degrees to grads	MODE 6 EXE 7.89 SHIFT MODE 4 EXE	8.766666667
Result displayed in degrees 47.3° + 82.5 rad = 4774.20181°	MODE 4 EXE 47.3 + 82.5 SHIFT MODE 5 EXE	4774.20181
12.4° + 8.3 rad − 1.8 gra = 486.33497°	12.4 + 8.3 SHIFT MODE 5 − 1.8 SHIFT MODE 6 EXE	486.33497
Result displayed in radians 24°6′31″ + 85.34 rad = 85.76077464 rad	MODE 5 EXE 24 °’” 6 °’” 31 °’” SHIFT MODE 4 + 85.34 EXE	85.76077464
Result displayed in grads 36.9° + 41.2 rad = 2663.873462 gra	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	Operation	Display
$\sin 63°52'41'' = 0.897859012$	MODE 4 EXE → "D" sin 63 °’” 52 °’” 41 °’” EXE	0.897859012
$\cos\left(\frac{\pi}{3}\text{ rad}\right) = 0.5$	MODE 5 EXE → "R" cos (SHIFT π ÷ 3) EXE	0.5
$\tan(-35\text{gra}) = -0.612800788$	MODE 6 EXE → "G" tan SHIFT (−) 35 EXE	−0.612800788
$2\cdot\sin 45° \times \cos 65°$ $= 0.5976724775$	MODE 4 EXE → "D" 2 × sin 45 × cos 65 EXE ↑ ↑ Can be omitted.	0.5976724775
$\cot 30° = \frac{1}{\tan 30°}$ $= 1.732050808$	1 ÷ tan 30 EXE	1.732050808
$\sec\left(\frac{\pi}{3}\text{ rad}\right) = \frac{1}{\cos(\pi/3\text{rad})}$ $= 2$	MODE 5 EXE → "R" 1 ÷ cos (SHIFT π ÷ 3) EXE	2.
$\text{cosec}30° = \frac{1}{\sin 30°} = 2$	MODE 4 EXE → "D" 1 ÷ sin 30 EXE	2.
$\sin^{-1}0.5 = 30°$ (Determines x for $\sin x = 0.5$)	SHIFT $\sin^{-1}$ 0.5 EXE ↑ Can be entered as .5.	30.
$\cos^{-1}\frac{\sqrt{2}}{2} = 0.7853981634$ rad $= \frac{\pi}{4}$ rad	MODE 5 EXE → "R" SHIFT $\cos^{-1}$ (√ 2 ÷ 2) EXE ÷ SHIFT π EXE	0.7853981634 0.25
$\tan^{-1}0.741 = 36.53844577°$ $= 36°32'18.4''$	MODE 4 EXE → "D" SHIFT $\tan^{-1}$ 0.741 EXE SHIFT ←°’”	36.53844577 36°32’18.4”

*If the total number of digits for degrees/minutes/seconds exceeds 11 digits, 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 within the unit as a decimal value.

$2.5\times(\sin^{-1}0.8-\cos^{-1}0.9)$ $=68°13'13.53''$	2.5 × (SHIFT $\sin^{-1}$ 0.8 − SHIFT $\cos^{-1}$ 0.9) EXE SHIFT ←°′″	68°13'13.53"

■Logarithmic and exponential functions

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

Example	Operation	Display
log 1.23 ($\log_{10}1.23$) $=8.99051114\times10^{-2}$	log 1.23 EXE	0.0899051114
ln90 ($\log_e 90$) = 4.49980967	ln 90 EXE	4.49980967
log456 ÷ ln456 = 0.4342944819 (log/ln ratio = constant M)	log 456 ÷ ln 456 EXE	0.4342944819
$4^x=64$ $x\cdot\log 4=\log 64$		
$x=\dfrac{\log 64}{\log 4}=3$	log 64 ÷ log 4 EXE	3.
$10^{1.23}=16.98243652$ (To obtain the anti-logarithm of common logarithm 1.23)	SHIFT 10^x 1.23 EXE	16.98243652
$e^{4.5}=90.0171313$ (To obtain the anti-logarithm of natural logarithm 4.5)	SHIFT e^x 4.5 EXE	90.0171313
$10^4\cdot e^{-4}+1.2\cdot10^{2.3}$ = 422.5878667	SHIFT 10^x 4 × SHIFT e^x SHIFT (−) 4 + 1.2 × SHIFT 10^x 2.3 EXE	422.5878667
$5.6^{2.3}=52.58143837$	5.6 x^y 2.3 EXE	52.58143837
$\sqrt[7]{123}$ $(=123^{\frac{1}{7}})$ = 1.988647795	7 $\sqrt[x]{\ }$ 123 EXE	1.988647795

$(78-23)^{-12}$ $=1.305111829\times10^{-21}$	(78 − 23) x^y SHIFT (−) 12 EXE	1.305111829 $^{-21}$
$2+3\times\sqrt[3]{64}-4=10$ *x^y and $\sqrt[x]{\ }$ given calculation priority over × and ÷.	2 + 3 × 3 $\sqrt[x]{\ }$ 64 − 4 EXE	10.
$2\times3.4^{(5+6.7)}=3306232.001$	2 × 3.4 x^y (5 + 6.7) EXE	3306232.001

■Hyperbolic functions and inverse hyperbolic functions

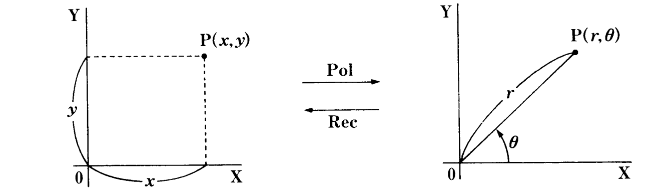
•The operations noted below cannot be performed in the BASE-N mode.

Example	Operation	Display
sinh 3.6 = 18.28545536	hyp sin 3.6 EXE	18.28545536
cosh 1.23 = 1.856761057	hyp cos 1.23 EXE	1.856761057
tanh 2.5 = 0.9866142981	hyp tan 2.5 EXE	0.9866142981
cosh 1.5 − sinh 1.5 = 0.2231301602 $=e^{-1.5}$ (Proof of cosh$x$ ± sinh$x$ = $e^{\pm x}$)	hyp cos 1.5 − hyp sin 1.5 EXE (Continuing) ln Ans EXE	0.2231301602 −1.5
$\sinh^{-1}30=4.094622224$	hyp SHIFT $\sin^{-1}$ 30 EXE	4.094622224
$\cosh^{-1}\left(\dfrac{20}{15}\right)=0.7953654612$	hyp SHIFT $\cos^{-1}$ (20 ÷ 15) EXE	0.7953654612
Determine the value of x when tanh $4x$ = 0.88 $x=\dfrac{\tanh^{-1}0.88}{4}$ = 0.3439419141	hyp SHIFT $\tan^{-1}$ 0.88 ÷ 4 EXE	0.3439419141
$\sinh^{-1}2\times\cosh^{-1}1.5$ = 1.389388923	hyp SHIFT $\sin^{-1}$ 2 × hyp SHIFT $\cos^{-1}$ 1.5 EXE	1.389388923
$\sinh^{-1}\left(\dfrac{2}{3}\right)+\tanh^{-1}\left(\dfrac{4}{5}\right)$ = 1.723757406	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 $-180° < \theta \leqq 180°$. The calculation range is the same for radians and grads.
- The operations noted below cannot be performed in the BASE-N mode.

Example	Operation	Display
If $x = 14$ and $y = 20.7$, what are r and θ°?	[MODE] [4] [EXE] → " D " [SHIFT] [Pol(] 14 [SHIFT] [,] 20.7 [)] [EXE] (Continuing) [ALPHA] [J] [EXE] [SHIFT] [←° ' "]	 24.98979792 (r) 55.92839019 55°55'42.2" (θ)
If $x = 7.5$ and $y = -10$, what are r and θ rad?	[MODE] [5] [EXE] → " R " [SHIFT] [Pol(] 7.5 [SHIFT] [,] [SHIFT] [(−)] 10 [)] [EXE] (Continuing) [ALPHA] [J] [EXE]	 12.5 (r) −0.927295218 (θ)
If $r = 25$ and $\theta = 56°$, what are x and y?	[MODE] [4] [EXE] → " D " [SHIFT] [Rec(] 25 [SHIFT] [,] 56 [)] [EXE] (Continuing) [ALPHA] [J] [EXE]	 13.97982259 (x) 20.72593931 (y)
If $r = 4.5$ and $\theta = \frac{2}{3}\pi$rad, what are x and y?	[MODE] [5] [EXE] → " R " [SHIFT] [Rec(] 4.5 [SHIFT] [,] [(] 2 [÷] 3 [×] [SHIFT] [π] [)] [)] [EXE] (Continuing) [ALPHA] [J] [EXE]	 −2.25 (x) 3.897114317 (y)

■Other functions ($\sqrt{\ }$, x^2, x^{-1}, $x!$, $\sqrt[3]{\ }$, Ran#, Abs, Int, Frac)

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

Example	Operation	Display
$\sqrt{2} + \sqrt{5} = 3.65028154$	[√] 2 [+] [√] 5 [EXE]	3.65028154
$2^2 + 3^2 + 4^2 + 5^2 = 54$	2 [x^2] [+] 3 [x^2] [+] 4 [x^2] [+] 5 [x^2] [EXE]	 54.
$\dfrac{1}{\frac{1}{3} - \frac{1}{4}} = 12$	[(] 3 [x^{-1}] [−] 4 [x^{-1}] [)] [x^{-1}] [EXE]	12.
$8!(= 1 \times 2 \times 3 \times \ldots\ldots \times 8)$ $= 40320$	8 [SHIFT] [$x!$] [EXE]	40320.
$\sqrt[3]{36 \times 42 \times 49} = 42$	[SHIFT] [$\sqrt[3]{\ }$] [(] 36 [×] 42 [×] 49 [)] [EXE]	 42.
Random number generation (pseudorandom number from 0.000 to 0.999)	[SHIFT] [Ran#] [EXE]	(Ex.) 0.792
$\sqrt{13^2 - 5^2} + \sqrt{3^2 + 4^2} = 17$	[√] [(] 13 [x^2] [−] 5 [x^2] [)] [+] [√] [(] 3 [x^2] [+] 4 [x^2] [)] [EXE]	 17.
$\sqrt{1 - \sin^2 40°} = 0.7660444431$ $= \cos 40°$ (Proof of $\cos\theta = \sqrt{1 - \sin^2\theta}$)	[MODE] [4] [EXE] → " D " [√] [(] 1 [−] [(] [sin] 40 [)] [x^2] [)] [EXE] (Continuing) [SHIFT] [$\cos^{-1}$] [Ans] [EXE]	 0.7660444431 40.
$\frac{1}{2!} + \frac{1}{4!} + \frac{1}{6!} + \frac{1}{8!} = 0.5430803571$	2 [SHIFT] [$x!$] [x^{-1}] [+] 4 [SHIFT] [$x!$] [x^{-1}] [+] 6 [SHIFT] [$x!$] [x^{-1}] [+] 8 [SHIFT] [$x!$] [x^{-1}] [EXE]	 0.5430803571
What is the absolute value of the common logarithm of $\frac{3}{4}$? $\left\lvert \log\frac{3}{4} \right\rvert = 0.1249387366$	[SHIFT] [Abs] [log] [(] 3 [÷] 4 [)] [EXE]	0.1249387366

What is the integer part of $\frac{7800}{96}$?	SHIFT Int (7800 ÷ 96) EXE	81.
What is the fraction part of $\frac{7800}{96}$?	SHIFT Frac (7800 ÷ 96) EXE	0.25
What is the aliquot part of 2512549139 ÷ 2141?	2512549139 ÷ 2141 EXE SHIFT Frac (2512549139 ÷ 2141) EXE	1173540. 0.99953

■Fractions

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

Example	Operation	Display
$\frac{2}{5}+3\frac{1}{4}=3\frac{13}{20}$ $=3.65$	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.		
$3\frac{456}{78}=8\frac{11}{13}$ (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.		
$\frac{1}{2578}+\frac{1}{4572}$ $=6.066202547\times10^{-4}$	1 a b/c 2578 + 1 a b/c 4572 EXE	6.066202547 −04 (NORM 1 mode)
*When the total number of characters, including integer, numerator, denominator and delimiter mark exceeds 10, the input fraction is automatically displayed in decimal format.		
$\frac{1}{2}\times0.5=0.25$	1 a b/c 2 × · 5 EXE	0.25
*Calculations containing both fractions and decimals are calculated in decimal format.		
$\frac{1}{3}\times(-\frac{4}{5})-\frac{5}{6}=-1\frac{1}{10}$	1 a b/c 3 × SHIFT (−) 4 a b/c 5 − 5 a b/c 6 EXE	−1⌟1⌟10.
$\frac{1}{2}\times\frac{1}{3}+\frac{1}{4}\times\frac{1}{5}=\frac{13}{60}$	1 a b/c 2 × 1 a b/c 3 + 1 a b/c 4 × 1 a b/c 5 EXE	13⌟60.
$\frac{\frac{1}{2}}{3}=\frac{1}{6}$	(1 a b/c 2) a b/c 3 EXE	1⌟6.
$\frac{1}{\frac{1}{3}+\frac{1}{4}}=1\frac{5}{7}$	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.		

2-4 Binary, Octal, Decimal, Hexadecimal Calculations

- Binary, octal, decimal and hexadecimal calculations, conversions and logical operations are performed in the BASE-N mode (press [MODE][−]).
- The number system (2, 8, 10, 16) is set by respectively pressing [BIN], [OCT], [DEC] or [HEX] followed by [EXE]. 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 bits, 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
Decimal	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
[C] (= [hyp])	C
[ID] (= [sin])	ID
[IE] (= [cos])	IE
[IF] (= [tan])	IF

- Calculation range (in BASE-N mode)

Binary	Positive	: $011111111111 \geqq x \geqq 0$
	Negative	: $111111111111 \geqq x \geqq 100000000000$
Octal	Positive	: $17777777777 \geqq x \geqq 0$
	Negative	: $37777777777 \geqq x \geqq 20000000000$
Decimal		$2147483647 \geqq x \geqq -2147483648$
Hexadecimal	Positive	: $7FFFFFFF \geqq x \geqq 0$
	Negative	: $FFFFFFFF \geqq x \geqq 80000000$

- You cannot specify the unit of angular measurement (degrees, 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	Operation	Display
What are the decimal values for $2A_{16}$ and 274_8?	MODE − → "BASE-N"	
	Dec EXE → "d"	
	SHIFT h 2A EXE	d 42
	SHIFT o 274 EXE	d 188
What are the hexadecimal values for 123_{10} and 1010_2?	Hex EXE → "H"	
	SHIFT d 123 EXE	H 0000007B
	SHIFT b 1010 EXE	H 0000000A
What are the octal values for 15_{16} and 1100_2?	Oct EXE → "o"	
	SHIFT h 15 EXE	o 00000000025
	SHIFT b 1100 EXE	o 00000000014
What are the binary values for 36_{10} and $3B7_{16}$?	Bin EXE → "b"	
	SHIFT d 36 EXE	b 000000100100
	SHIFT h 3B7 EXE	b 001110110111

■Negative expressions

Example	Operation	Display
	MODE − → "BASE-N"	
How is 110010_2 expressed as a negative?	Bin EXE → "b"	
	Neg 110010 EXE	b 111111001110
How is 72_8 expressed as a negative?	Oct EXE → "o"	
	Neg 72 EXE	o 37777777706
How is $3A_{16}$ expressed as a negative?	Hex EXE → "H"	
	Neg 3A EXE	H FFFFFFC6

■Basic arithmetic operations using binary, octal, decimal and hexadecimal values

Example	Operation	Display
	MODE − → "BASE-N"	
$10111_2+11010_2=110001_2$	Bin EXE → "b"	
	10111 + 11010 EXE	b 000000110001
$B47_{16}-DF_{16}=A68_{16}$	Hex EXE → "H"	
	B47 − DF EXE	H 00000A68
$123_8\times ABC_{16}=37AF4_{16}$	SHIFT o 123 × ABC EXE	H 00037AF4
$=228084_{10}$	Dec EXE	d 228084
$1F2D_{16}-100_{10}=7881_{10}$	SHIFT h 1F2D − 100 EXE	d 7881
$=1EC9_{16}$	Hex EXE	H 00001EC9
$7654_8\div 12_{10}=334.3333333_{10}$	Dec EXE → "d"	
$=516_8$	SHIFT o 7654 ÷ 12 EXE	d 334
	Oct EXE	o 00000000516

*Calculation results are displayed with the decimal portion cut off.

Example	Operation	Display
$1234_{10}+1EF_{16}\div 24_8$	SHIFT d 1234 + SHIFT h	
$=2352_8$	1EF ÷ 24 EXE	o 00000002352
$=1258_{10}$	Dec EXE	d 1258

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

■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	Operation	Display
	MODE − → "BASE-N"	
19_{16} AND $1A_{16}=18_{16}$	Hex EXE → "H" 19 and 1A EXE	H 00000018
1110_2 AND $36_8=1110_2$	Bin EXE → "b" 1110 and SHIFT o 36 EXE	b 0000000001110
23_8 OR $61_8=63_8$	Oct EXE → "o" 23 or 61 EXE	o 00000000063
120_{16} OR $1101_2=12D_{16}$	Hex EXE → "H" 120 or SHIFT b 1101 EXE	H 0000012D
1010_2 AND (A_{16} OR 7_{16}) $=1010_2$	Bin EXE → "b" 1010 and (SHIFT h A or SHIFT h 7) EXE	b 0000000001010
5_{16} XOR $3_{16}=6_{16}$	Hex EXE → "H" 5 SHIFT xor 3 EXE	H 00000006
$2A_{16}$ XNOR $5D_{16}=FFFFFF88_{16}$	Hex EXE → "H" 2A SHIFT xnor 5D EXE	H FFFFFF88
Negation of 1234_8	Oct EXE → "o" Not 1234 EXE	o 37777776543
Negation of $2FFFED_{16}$	Hex EXE → "H" Not 2FFFED EXE	H FFD00012

2-5 Statistical Calculations

■Standard deviation

- •Standard deviation calculations are performed in the SD1 mode (MODE ×). "SD1" 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 represents the number of times the data is repeated, and then DT.

•Standard Deviation

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

[Using all data from a finite population to determine the standard deviation for the population]

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

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

•Mean

$$\overline{x}=\frac{\sum_{i=1}^{n}x_i}{n}=\frac{\Sigma x}{n}$$

**The values for n, Σx, and Σx^2 are stored in memories W, V, and U respectively, and can be obtained by pressing ALPHA followed by the memory name and then EXE (i.e. ALPHA W EXE).*

Example	Operation	Display
Data 55, 54, 51, 55, 53, 53, 54, 52	MODE × SHIFT Scl EXE (Clears memory) 55 DT 54 DT 51 DT 55 DT 53 DT DT 54 DT 52 DT	52.
*You can press the function keys to obtain results in any sequence.		
	(Standard deviation σn) SHIFT $x\sigma n$ EXE	1.316956719
	(Standard deviation $\sigma n-1$) SHIFT $x\sigma n\text{-}1$ EXE	1.407885953

	(Mean $\bar{x}$) SHIFT $\bar{x}$ EXE	53.375
	(Number of data n) ALPHA W EXE	8.
	(Sum total Σx) ALPHA V EXE	427.
	(Sum of squares Σx^2) ALPHA U EXE	22805.
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	1.982142857
	55 − SHIFT $\bar{x}$ EXE	1.625
	54 − SHIFT $\bar{x}$ EXE	0.625
	51 − SHIFT $\bar{x}$ EXE	−2.375
	⋮	⋮
To calculate x and σ_{n-1} for the following data	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.
	ALPHA W EXE	70.
	SHIFT $\bar{x}$ EXE	137.7142857
	SHIFT $x\sigma_{n-1}$ EXE	18.42898069

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

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

① If 50 DT is entered, enter correct data after pressing CL.

② If 49 DT was input a number of entries previously, enter correct data after 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 LR1 mode (MODE ÷). "LR1" 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 ; followed by a value representing the number of times the data is repeated, and then DT.
- If only x data is repeated (x data having the same value), enter SHIFT , y data DT or SHIFT , y data SHIFT ; followed by a value representing the number of times the data is 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 following are the formulas the unit uses to calculate constant term A and regression coefficient B for the regression formula $y = \mathrm{A} + \mathrm{B}x$.

Constant term of regression formula

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

Regression coefficient of regression formula

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

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

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

(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:

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

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

■Linear regression

Example

•Relationship between temperature and the length of a steel bar

Temperature	Length
10°C	1003mm
15°C	1005mm
20°C	1010mm
25°C	1011mm
30°C	1014mm

The data in the above table 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 and the temperature when the bar is 1000 mm long can be calculated. The critical coefficient (r^2) and covariance $\left(\frac{\Sigma xy - n \bullet \bar{x} \bullet \bar{y}}{n-1}\right)$ can also be calculated.

Operation	Display
[MODE] [÷]	
[SHIFT] [Scl] [EXE] (Clears memory)	
10 [SHIFT] [,] 1003 [DT]	10.
15 [SHIFT] [,] 1005 [DT]	15.
20 [SHIFT] [,] 1010 [DT]	20.
25 [SHIFT] [,] 1011 [DT]	25.
30 [SHIFT] [,] 1014 [DT]	30.
(Constant term A) [SHIFT] [A] [EXE]	997.4
(Regression coefficient B) [SHIFT] [B] [EXE]	0.56
(Correlation coefficient r) [SHIFT] [r] [EXE]	0.9826073689
(Length at 18°C) 18 [SHIFT] [$\hat{y}$] [EXE]	1007.48
(Temperature at 1000mm) 1000 [SHIFT] [$\hat{x}$] [EXE]	4.642857143
(Critical coefficient) [SHIFT] [r] [x^2] [EXE]	0.9655172414
(Covariance) [(] [ALPHA] [R] [−] [ALPHA] [W] [×] [SHIFT] [$\bar{x}$] [×] [SHIFT] [$\bar{y}$] [)] [÷] [(] [ALPHA] [W] [−] 1 [)] [EXE]	35.

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

①If 11 [SHIFT] [,] 1003 is entered, enter correct data after pressing [AC].

②If 11 [SHIFT] [,] 1003 [DT] is entered, enter correct data after pressing [CL].

③If 11 [SHIFT] [,] 1003 [DT] was entered previously, enter correct data after pressing 11 [SHIFT] [,] 1003 [CL].

■Logarithmic regression

•The regression formula is $y = \text{A} + \text{B} \bullet \ln x$. 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:

$$\hat{y} = \text{A} + \text{B} \bullet \ln x \qquad \hat{x} = \exp\left(\frac{y - \text{A}}{\text{B}}\right)$$

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

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

Example

xi	yi
29	1.6
50	23.5
74	38.0
103	46.4
118	48.9

The data in the above table 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 $xi = 80$, and estimated value $\hat{x}$ can be obtained for $yi = 73$.

Operation	Display
[MODE] [÷]	
[SHIFT] [Scl] [EXE] (Clears memory)	
[ln] 29 [SHIFT] [,] 1.6 [DT]	3.36729583
[ln] 50 [SHIFT] [,] 23.5 [DT]	3.912023005
[ln] 74 [SHIFT] [,] 38.0 [DT]	4.304065093
[ln] 103 [SHIFT] [,] 46.4 [DT]	4.634728988
[ln] 118 [SHIFT] [,] 48.9 [DT]	4.770684624
(Constant term A) [SHIFT] [A] [EXE]	−111.1283976
(Regression coefficient B) [SHIFT] [B] [EXE]	34.02014749
(Correlation coefficient r) [SHIFT] [r] [EXE]	0.9940139464
($\hat{y}$ when $xi = 80$) [ln] 80 [SHIFT] [$\hat{y}$] [EXE]	37.94879482
($\hat{x}$ when $yi = 73$) 73 [SHIFT] [$\hat{x}$] [EXE] [SHIFT] [e^x] [Ans] [EXE]	224.1541314

■Exponential regression

- The regression formula is $y = A \cdot e^{B \cdot x}$ ($\ln y = \ln A + Bx$). Enter the y data as the logarithm of y(ln), and the x data 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:

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

- 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 x SHIFT $\hat{y}$ EXE SHIFT e^x Ans EXE, and estimated value $\hat{x}$ is obtained by ln y SHIFT $\hat{x}$ EXE. Σy, Σy^2 and Σxy are obtained by $\Sigma \ln y$, $\Sigma(\ln y)^2$ and $\Sigma x \cdot \ln y$ respectively.

Example

xi	*yi*
6.9	21.4
12.9	15.7
19.8	12.1
26.7	8.5
35.1	5.2

The data in the above table 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 $xi = 16$, and estimated value $\hat{x}$ can be obtained for $yi = 20$.

Operation	Display
MODE ÷	
SHIFT Scl EXE (Clears memory)	
6.9 SHIFT , ln 21.4 DT	6.9
12.9 SHIFT , ln 15.7 DT	12.9
19.8 SHIFT , ln 12.1 DT	19.8
26.7 SHIFT , ln 8.5 DT	26.7
35.1 SHIFT , ln 5.2 DT	35.1
(Constant term A) SHIFT e^x SHIFT A EXE	30.49758742
(Regression coefficient B) SHIFT B EXE	−0.049203708
(Correlation coefficient r) SHIFT r EXE	−0.997247351
($\hat{y}$ when $xi = 16$) 16 SHIFT $\hat{y}$ EXE SHIFT e^x Ans EXE	13.87915739
($\hat{x}$ when $yi = 20$) ln 20 SHIFT $\hat{x}$ EXE	8.574868046

■Power regression

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

$$\hat{y} = A \cdot x^B \qquad \hat{x} = \exp\left(\frac{\ln y - \ln A}{B}\right)$$

- 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 value $\hat{x}$ is obtained by ln y SHIFT $\hat{x}$ EXE SHIFT 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 $\Sigma \ln x \cdot \ln y$ respectively.

Example

xi	*yi*
28	2410
30	3033
33	3895
35	4491
38	5717

The data in the above table 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 $yi = 1000$.

Operation	Display
MODE ÷	
SHIFT Scl EXE (Clears memory)	
ln 28 SHIFT , ln 2410 DT	3.33220451
ln 30 SHIFT , ln 3033 DT	3.401197382
ln 33 SHIFT , ln 3895 DT	3.496507561
ln 35 SHIFT , ln 4491 DT	3.555348061
ln 38 SHIFT , ln 5717 DT	3.63758616
(Constant term A) SHIFT e^x SHIFT A EXE	0.2388010829
(Regression coefficient B) SHIFT B EXE	2.771866148
(Correlation coefficient r) SHIFT r EXE	0.9989062562
($\hat{y}$ when $xi = 40$) ln 40 SHIFT $\hat{y}$ EXE SHIFT e^x Ans EXE	6587.674743
($\hat{x}$ when $yi = 1000$) ln 1000 SHIFT $\hat{x}$ EXE	
SHIFT e^x Ans EXE	20.26225659

Graphs

The innovative graphing function of this calculator employs a dot display that gives you detailed representation of mathematical functions and statistics. In addition to using the built-in functions, you can also graph any function by simply inputting its formula. Graphing commands can be used alone (direct input) or within programs.
For the sake of simplicity, the examples in this section show direct input graphing commands. For full details on using graphing commands within programs, see page 137.

Graphs

3-1 Built-in Function Graphs

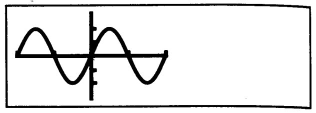
The COMP mode of the RUN mode should be used when graphing functions. Some graphs can be produced in the SD and LR modes, but certain graphs cannot be produced in these modes. The BASE-N mode cannot be used for graphs. This unit contains a total of 20 built-in graphs making it possible to produce the graphs of basic functions.

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

Any time a built-in graph is executed, the ranges (see page 78) are automatically set to their optimum values, and any graph previously on the display is cleared.

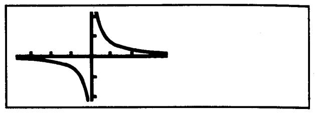
Example 1 **Sine curve**

MODE +
Graph sin EXE



Example 2 $y = \frac{1}{x}$ **graph**

Graph x^{-1} EXE



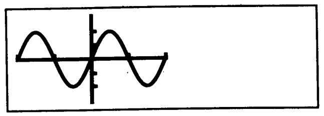
■Overdrawing built-in function graphs

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 (see page 83).

Example **Overdraw the graph for $y=\cos x$ on the graph for $y=\sin x$.**

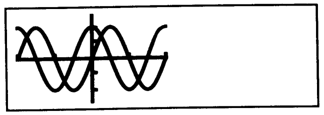
First, draw the graph for $y=\sin x$.

Graph sin EXE



Next, draw the graph for $y=\cos x$ without changing the existing range.

Graph cos ALPHA X EXE



⟨**Note**⟩
Built-in function graphs cannot be used in multistatements (see page 53) and cannot be written into programs.

3-2 User Generated Graphs

Built-in function graphs can also be used in combination with each other. Graphing a formula such as $y=2x^2+3x-5$ 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.

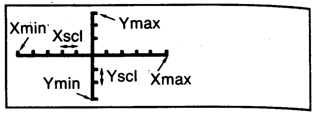
■Range parameters

After pressing the Range key, you can look up and specify 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



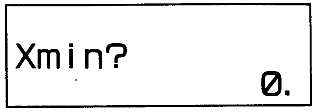
•Specifying range parameters

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.

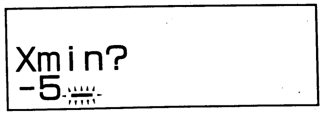
Example **Change the range parameters on the left to those on the right.**

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

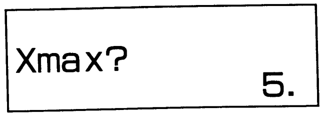
Range


Xmin?
0.

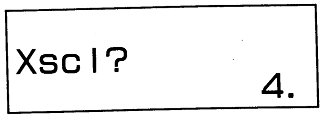
Specifies −5 for Xmin.
SHIFT (−) 5


Xmin?
-5

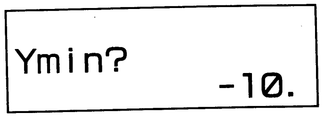
EXE


Xmax?
5.

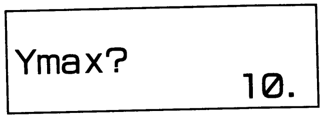
Xmax does not change, so simply press EXE.
EXE


Xscl?
4.

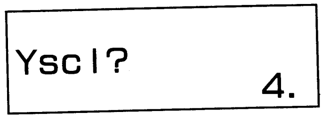
Specifies 2 for Xscl.
2 EXE


Ymin?
-10.

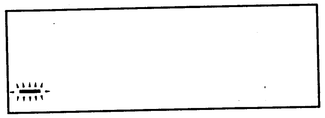
Specifies −5 for Ymin.
SHIFT (−) 5 EXE


Ymax?
10.

Specifies 15 for Ymax.
1 5 EXE


Yscl?
4.

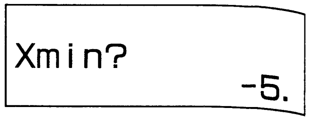
Yscl does not change, so simply press EXE.
EXE



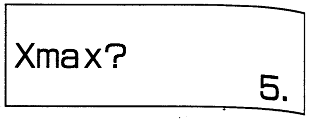
•Checking range parameters

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.

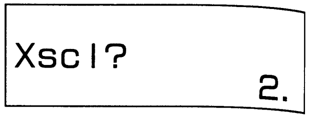
Range


Xmin?
-5.

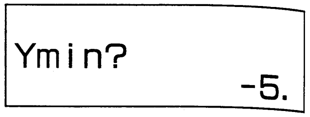
EXE


Xmax?
5.

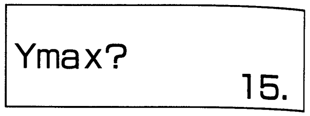
EXE


Xscl?
2.

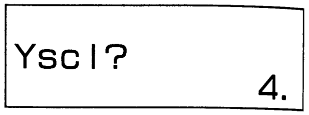
EXE


Ymin?
-5.

EXE


Ymax?
15.

EXE


Yscl?
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.

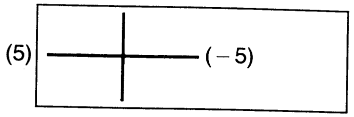
The input range for graph ranges is $-9.999999999_{E}+97$ through $9.999999999_{E}+97$.

**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 Xscl or Yscl does not set any scale.*

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

Example **Xmin: 5**
Xmax: −5



**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 the top 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 settings 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 that 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

⟨Reference⟩

Range settings are performed within programs using the following format:

Range Xmin value, Xmax value, Xscl value, Ymin value, Ymax value, Yscl value

Up to six data items are programmed after the Range command. When less than six items are programmed, range setting is performed in the order from the beginning of the above format.

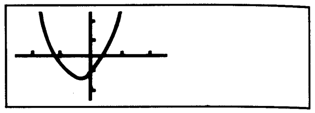
■User generated function graphs

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 x^2 +
3 ALPHA X − 4 EXE



The result produces a visual representation of the formula.

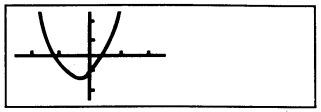
■Function graph overdraw

Two or more function graphs can be overdrawn, which makes 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 $y = 2x + 3$.**

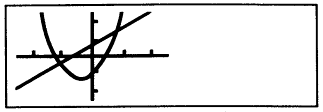
First, clear the graph screen in preparation for the first graph.

SHIFT Cls EXE
Graph 2 ALPHA X x^2 +
3 ALPHA X − 4 EXE



Next, overdraw 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 overdraw.*
If variable X is not included in the second formula, the second graph 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 Trace or Plot function to locate the pointer at a specific point on the graph, the enlargement/reduction is performed using the pointer location as the center point.

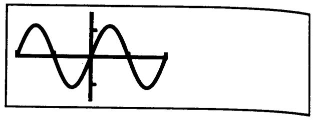
•Enlarging a graph

Example **To enlarge the graph for $y=\sin x$ 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

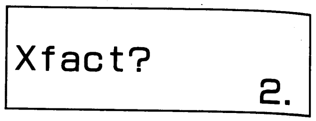
After specifying the range parameters, graph $y=\sin x$.

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



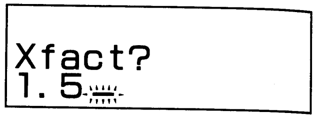
Press [SHIFT][Factor] for the factor specification screen.

[SHIFT][Factor]



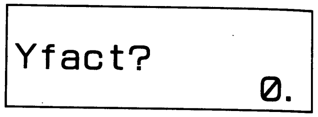
```
Xfact?
                2.
```

[1][•][5]



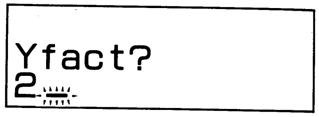
```
Xfact?
1.5
```

[EXE]



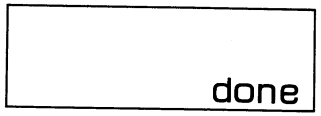
```
Yfact?
                0.
```

[2]



```
Yfact?
2
```

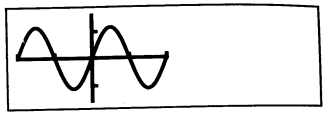
[EXE]



```

              done
```

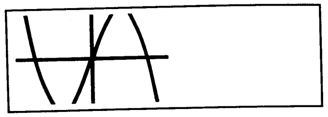
Whenever you try to change the factor value while a graph is displayed, the display changes to the text screen automatically. To return to the graph screen after changing the factor value, press [G↔T] or [EXE].

[EXE](or [G↔T])



Press [SHIFT][Zoom×f] to enlarge the graph according to the factors you specified.

[SHIFT][Zoom×f]



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

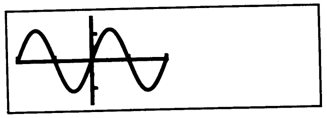
•Reducing a graph

Example **To reduce the graph for $y=\sin x$ 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

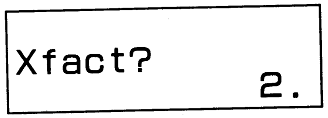
After specifying the range parameters, graph $y=\sin x$.

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



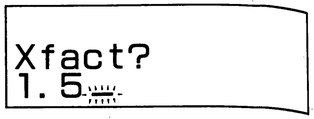
Press [SHIFT][Factor] for the factor specification screen.

[SHIFT][Factor]

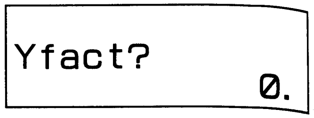


```
Xfact?
                2.
```

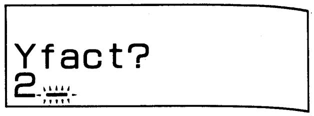
1 • 5


Xfact?
1.5

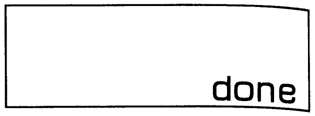
EXE


Yfact?
0.

2

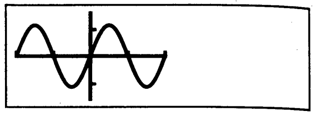

Yfact?
2

EXE


done

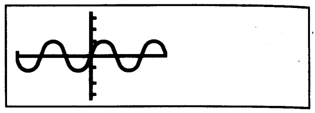
**Whenever you try to change the factor value while a graph is displayed, the display changes to the text screen automatically. To return to the graph screen after changing the factor value, press G-T or EXE.*

EXE (or G-T)



Press SHIFT Zoom×½ to reduce the graph according to the factors you specified.

SHIFT Zoom×½



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 Zoom×½ 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 factors 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. You can display the coordinates using either seven digits or eleven digits (including negative sign).

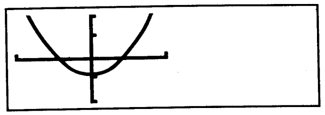
•Using the trace function

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.**

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

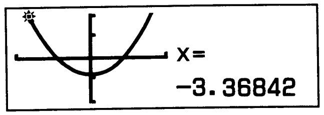
After specifying the range parameters, graph $y=x^2-3$.

SHIFT Cls EXE
Graph ALPHA X x^2 − 3 EXE



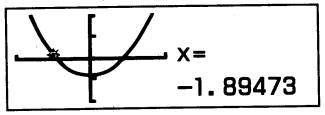
Activate the Trace function.

Trace


X=
−3.36842

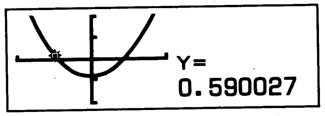
Use ⇨ and ⇦ to move the pointer along the graph. Each press moves the cursor one point. Holding down either key moves the pointer at high speed.

⇨ ~ (Hold down.)

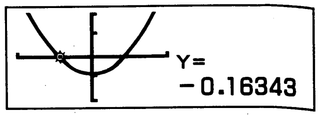

X=
−1.89473

Move the pointer to the point where the graph intersects the x-axis. Note that this point will be represented by a y-axis value of 0, so first press SHIFT X↔Y to change the coordinate display to show the y-coordinate.

SHIFT X↔Y

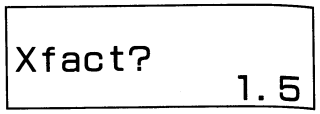

Y=
0.590027

⇨

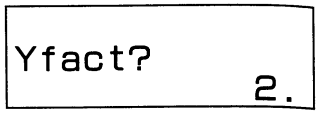

Y=
−0.16343

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

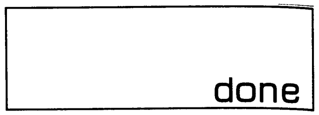
SHIFT Factor


Xfact?
1.5

2 EXE

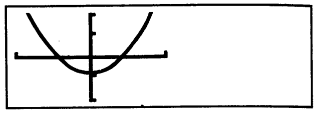

Yfact?
2.

2 EXE


done

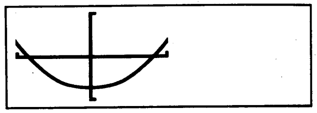
**Whenever you try to change the factor value while a graph is displayed, the display changes to the text screen automatically. To return to the graph screen after changing the factor value, press G↔T or EXE.*

EXE (or G↔T)



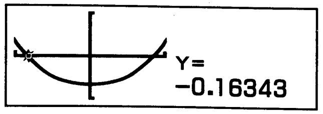
Now enlarge the graph according to the factors.

SHIFT Zoom×f



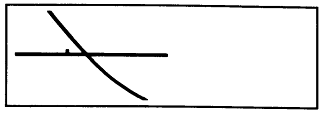
Activate the Trace function and move the pointer again.

Trace SHIFT X↔Y
⇨


Y=
−0.16343

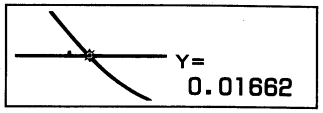
Enlarge the graph again to check the location of the pointer.

SHIFT Zoom×f



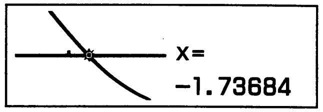
Activate the Trace function and move the pointer again.

Trace SHIFT X↔Y
⇨ ~ (Hold down)


Y=
0.01662

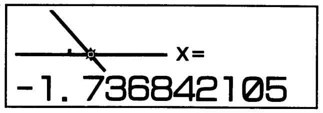
View the x-coordinate value.

SHIFT X↔Y


X=
−1.73684

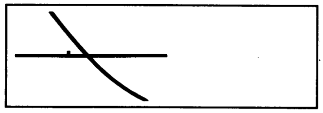
Press SHIFT Value to display the x-coordinate.

SHIFT Value


X=
−1.736842105

Press Trace again to exit the Trace function.

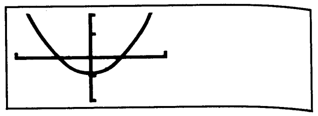
Trace



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 can only be used 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 11-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 executed 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^2][ALPHA][◢][Graph]2[ALPHA][X][+]5.

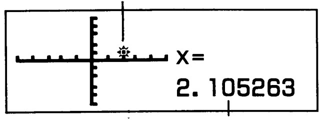
■Plot function

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 (see Line function, page 93).
Press [SHIFT][Plot] and specify the x- and y-coordinates 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]

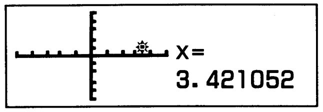


X=
2. 105263

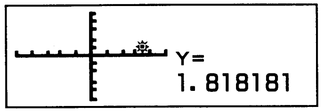
The blinking pointer is positioned at the specified coordinates.
**Due to limitations caused by the resolution of the display, the actual position of the pointer can only be approximate.*

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.

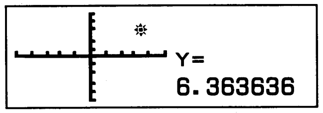
[⇨][⇨][⇨][⇨][⇨]


X=
3. 421052

To find the y-coordinate value:
[SHIFT][X↔Y]

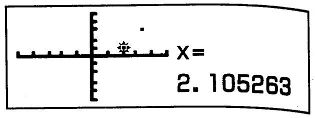

Y=
1. 818181

[⇧][⇧][⇧][⇧][⇧]


Y=
6. 363636

When the pointer is at the location you want, press EXE to plot a point. At this time, the pointer returns to the original point you specified (2, 2 in this example).

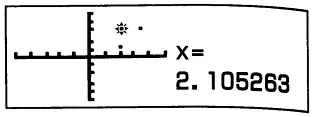
EXE



X=
2.105263

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

SHIFT Plot 2 SHIFT ,
6 • 5 EXE

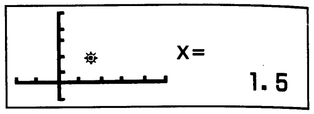


X=
2.105263

If 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

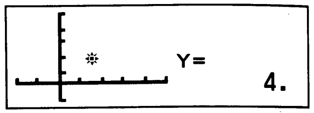
SHIFT Plot EXE



X=
1.5

To find the y-coordinate value:

SHIFT X↔Y



Y=
4.

**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.

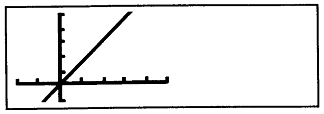
Example **Draw perpendiculars from point (2,0) on the x-axis to its intersection with the graph for $y=3x$. Then draw a line from the point of intersection to the y-axis.**

The range values for the graph are as follows:

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

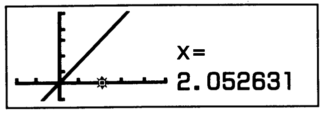
Clear the graph display and draw the graph for $y=3x$.

SHIFT Cls EXE
Graph 3 ALPHA X EXE



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

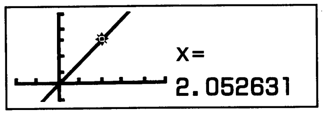
SHIFT Plot 2 SHIFT , 0 EXE



X=
2.052631

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

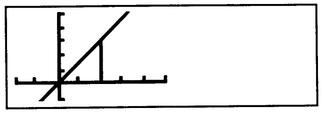
SHIFT Plot 2 SHIFT , 0 EXE
⇧ ~ ⇧
(Move the pointer up to the point on the graph for $y=3x$.)



X=
2.052631

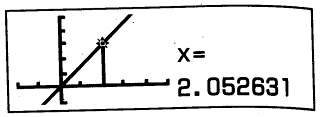
Draw a line using the Line function.

SHIFT Line EXE



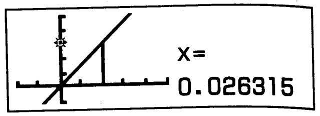
Next, a perpendicular will be drawn from the same point on the graph to the y-axis. First, plot the point on the graph and use the cursor key (⇦) to move the pointer to the y-axis. This can be accomplished using Plot X, Y since the x-y coordinates of the point on the graph are stored in the X and Y memories.

SHIFT Plot ALPHA X SHIFT ,
ALPHA Y EXE



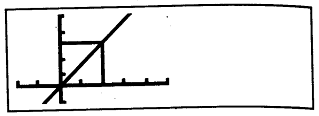
X=
2.052631

⇦ ~ ⇦
(Move the pointer to the y-axis.)



X=
0.026315

SHIFT Line EXE



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

■Graph scroll function

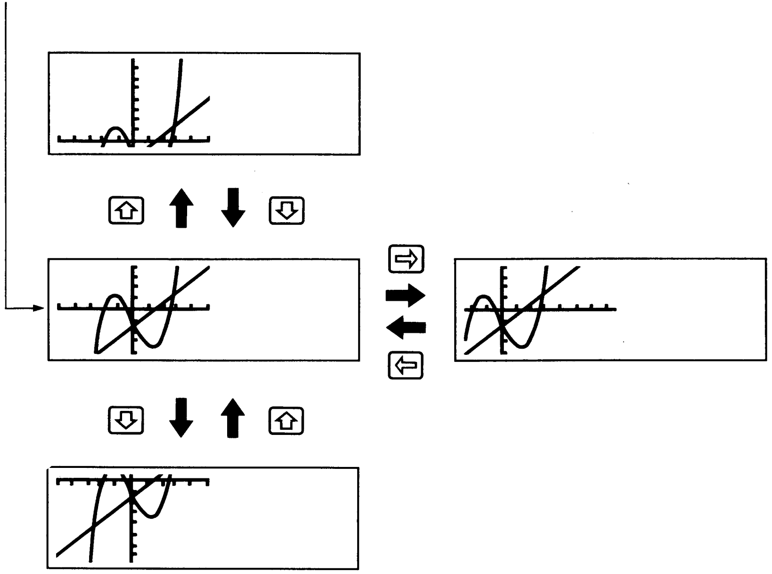
Immediately after you have drawn a graph, you can scroll it on the display. Use the cursor keys to scroll the graph left, right, up and down.

•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 • 2 5 (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.

3-3 Some Graphing Examples

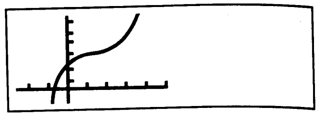
The following examples are presented to show you some ways that the graphing functions can be used effectively.

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 +
2 7 ALPHA X + 5 0 EXE

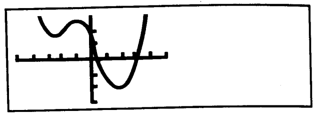


Example 2 **To graph the function $y=x^4+4x^3-36x^2-160x+300$ and dertermine its minimum and maximum**

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 4 +
4 ALPHA X x^y 3 −
3 6 ALPHA X x^2 − 1 6 0
ALPHA X + 3 0 0 EXE



3-4 Single-Variable Statistical Graphs

- Single-variable statistical graphs are drawn in the SD2 mode (SHIFT MODE ×). "SD2" appears on the display.
- Bar graphs and normal distribution curves can be produced as single-variable statistical graphs.
- Function graphs are also possible in the SD2 mode, so graphs of theoretical values and graphs of actual values can be overdrawn.
  **Abs and $^3\sqrt{\ }$ cannot be used in the SD2 mode.*
- The maximum number of data items is identical to the number of memories. You can expand memories up to a maximum of 19. If you specify a number greater than 19, the unit automatically sets the number of memories as 19.
- Graphs are drawn with the x-coordinate as the data range and the y-coordinate as the number of items (frequency) of each data.
- The DT key is used for data input.
- The CL key is used for data correction.

■Drawing single-variable statistical graphs

•Procedure

① Specify the SD2 mode (SHIFT MODE ×).
② Set the range values (Range).
③ Expand the memory in accordance with the number of bars (MODE • n EXE).
④ Clear the statistical memories (SHIFT Scl EXE).
⑤ Input data (Data DT).
⑥ Draw the graph.

- Bar graph .... Graph EXE
- Normal distribution curve .... Graph SHIFT Line 1 EXE

**Data input method in step 5 is the same as that for standard deviation calculations (see page 67).*

Example **Use the following data to draw a ranked graph.**

Rank No.	1	2	3	4	5	6	7	8	9	10	11
Rank	0	10	20	30	40	50	60	70	80	90	100
Frequency	1	3	2	2	3	5	6	8	15	9	2

Perform graph preparation in accordance with the following procedure:

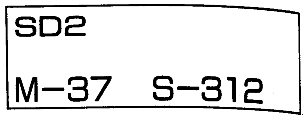
① Specify the SD2 mode (SHIFT MODE ×).

② Set the range values.
The highest value to be plotted on the x-axis is 100, but for graphing purposes the maximum value (Xmax) is set at 110. (The general rule is that the minimum value should be equal to or greater than the minimum range value and the maximum value should be less than the maximum range value, so here we set the x-axis ranges to 0 through 110).
Ymax value is set to 20 for the y-axis because the maximum frequency is 15.

Xmin :	0	Ymin :	0
Xmax :	110	Ymax :	20
Xscl :	10	Yscl :	2

③ Since the number of bars is 11(0~9, 10~19, 20~29.... 100~109), expand memories by 11.

MODE · 1 1 EXE



```
SD2

M-37   S-312
```

④ Clear the statistical memory.

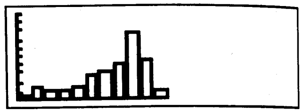
SHIFT Scl EXE

⑤ Input the data.

0 DT 10 DT DT DT 20 DT DT 30 DT DT 40 DT DT DT
50 SHIFT ; 5 DT 60 SHIFT ; 6 DT 70 SHIFT ; 8 DT
80 SHIFT ; 15 DT 90 SHIFT ; 9 DT 100 DT DT

⑥ First, draw a bar graph.

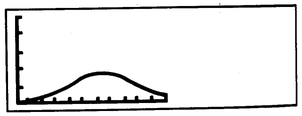
Graph EXE



- Next, draw a normal distribution curve. Since the y-axis value is relatively small when compared with the bar graph, the same range values cannot be used. Change the range values to those shown below.

Xmin :	0	Ymin :	0
Xmax :	110	Ymax :	0.05
Xscl :	10	Yscl :	0.01

Graph SHIFT Line 1 EXE
↑
Inputting the number 1 causes a normal distribution curve to be drawn.



- Be sure to expand the memory in accordance with the number of bars. A Mem ERROR is generated if memory expansion is not performed.
- If the number of expanded memories is changed during data input, the number of data divisions also changes, thus making it impossible to produce a proper graph.
- When a value that exceeds the preset ranges is input, it is input to the statistical memory, but not into the graph memory.
- When more data than the preset y-axis range is input, the bar graph is drawn to the upper limit of the display, and the points outside the range cannot be connected.
- The formula used for normal distribution curves is:

$$y=\frac{1}{\sqrt{2\pi}\,\sigma}e^{-\frac{(x-m)^2}{2\sigma^2}}$$

**Keyboard designation of σ is $x\sigma n$. m is $\bar{x}$.*

- The following must be true in the case of range settings: Xmin<Xmax.
- After a bar graph is executed, "done" is displayed in the text display.

3-5 Paired-Variable Statistical Graphs

- Paired-variable graphs are drawn in the LR2 mode (SHIFT MODE ÷). "LR2" appears on the display.
- Paired-variable graphs can be drawn as regression lines.
- Standard function graphs can also be drawn in the LR2 mode, so theoretical graphs, data distribution and regression line graphs can be overdrawn.
- After data input in the LR2 mode, points are displayed immediately, and data is input to the statistical memory.
- When a value that exceeds the preset range is input, it is input to the statistical memory, but the point is not displayed.
- Data is input using the DT key in the following format: x data SHIFT , y data SHIFT ; frequency DT.
- The CL key is used to edit data after input is complete, but points that are produced on the display are not cleared. (Point appears even when data is corrected by the CL key).
- Points on the display cannot be retrieved if the display is cleared (SHIFT Cls EXE).

■Drawing paired-variable statistical graphs

•Procedure

①Specify the LR2 mode (SHIFT MODE ÷).
②Set the range values (Range).
③Clear the statistical memory (SHIFT Scl EXE).
④Input data (x data SHIFT , y data SHIFT ; frequency DT).
⑤Draw the graph (Graph SHIFT Line 1 EXE).

**Data input method in step 4 is the same as that for Regression calculation (page 69).*

Example **Perform linear regression on the following data and draw a regression line graph.**

x_i	y_i
−9	−2
−5	−1
−3	2
1	3
4	5
7	8

①Specify the LR2 mode (SHIFT MODE ÷).
②Set the range values to those shown in the table.

Xmin :	−10	Ymin :	−5
Xmax :	10	Ymax :	15
Xscl :	2	Yscl :	5

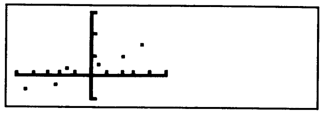
**According to the general rule of the x-axis range values, the values for x are: $-10 \leqq x < 10$.*

③Clear the statistical memories.

SHIFT Scl EXE

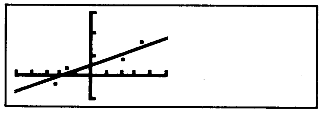
④Input the data.

SHIFT (−) 9 SHIFT , SHIFT (−) 2 DT
SHIFT (−) 5 SHIFT , SHIFT (−) 1 DT
SHIFT (−) 3 SHIFT , 2 DT
1 SHIFT , 3 DT
4 SHIFT , 5 DT
7 SHIFT , 8 DT



⑤Draw the graph.

Graph SHIFT Line 1 EXE



**When data is input that is outside of the preset range values, a point does not appear.*
**An Ma ERROR is generated when there is no data input and the following key operation is performed: Graph SHIFT Line 1 EXE.*
**The following must be true in the case of range settings: Xmin < Xmax.*

4 Program Calculations

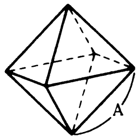
4 Program Calculations

4-1 What is 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 "multistatement" 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^3
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[×][√]3[×] Numeric value A[x^2][EXE]
Volume (V): [√]2[÷]3[×] Numeric value A[x^y]3[EXE]

In the above example, numeric 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[×][√]3[×][ALPHA][A][x^2][EXE] .... S
[√]2[÷]3[×][ALPHA][A][x^y]3[EXE] V

With this unit, the operations performed for manual calculations can be used as they are in a 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 (see page 135) 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 displayed 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:

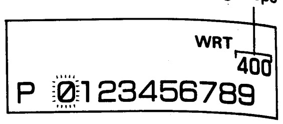
```
[ALPHA][?][→][ALPHA][A][:]2[×][√]3[×][ALPHA][A][x²][ALPHA][◢]
└──────────────────────┘                          └─────────┘
      ↑                                                ↑
      └──── Input to memory A                          └─ Display S

[√]2[÷]3[×][ALPHA][A][xʸ]3
                          ↑
                          └─ ◢ omitted
```

Now the program is complete.

③ Program storage

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

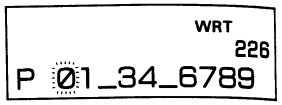
[MODE][2]



Number of remaining steps

WRT
400
P Ø123456789

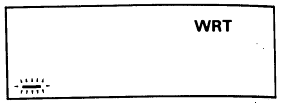
When [MODE][2] are pressed, the system mode changes to the WRT mode. Then, the number of remaining steps (see page 116) is indicated. The number of remaining steps is decreased when programs are input or when memories are expanded. If no programs have been input and the number of memories equals 26 (the number of memories at initialization), the number of usable steps should equal 400. The larger figures located below indicate the program areas (see page 117). If the letter "P" is followed by the numbers Ø 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
P Ø1_34_6789

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

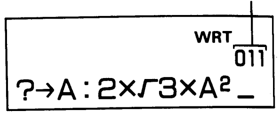
[EXE](Start storage)



WRT
-

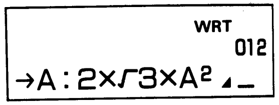
Number of steps used for program area P0.

[ALPHA][?][→][ALPHA][A][:][2][×][√][3]
[×][ALPHA][A][x²]



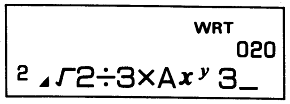
WRT
011
?→A:2×√3×A²_

[ALPHA][◢]



WRT
012
→A:2×√3×A²◢_

[√][2][÷][3][×][ALPHA][A][xʸ][3]



WRT
020
2◢√2÷3×Axʸ3_

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

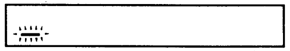
④ Program execution

Programs 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 as:

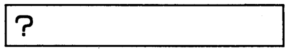
Length of one side (A)	Surface area (S)	Volume (V)
10cm	(346.4101615)cm²	(471.4045208)cm³
7	(169.7409791)	(161.6917506)
15	(779.4228634)	(1590.990258)

[MODE][1]



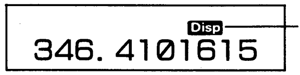
-

[Prog][0][EXE]



?

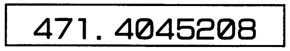
[1][0][EXE]
(Value of A)



Disp
346.4101615

Indicates answer displayed by ◢.
(S when A = 10)

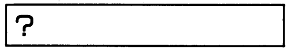
[EXE]



471.4045208

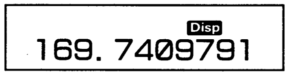
(V when A = 10)

[Prog][0][EXE]



?

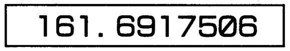
[7][EXE]
(Value of A)



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]	(Reexecute)
	7[EXE]	(Input 7 for A)
	[EXE]	(Display V when A = 7)
	⋮	

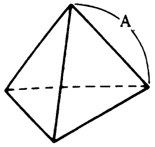
4-2 Program Checking and Editing (Correction, Addition, Deletion)

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 octahedron).

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)
10 cm	( )cm^2	( )cm^3
7.5	( )	( )
20	( )	( )

① Formulas

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

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

② 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][xʸ]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][xʸ]3

③ Program editing

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

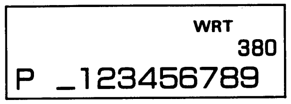
Octahedron: ALPHA ? → ALPHA A : 2 × √ 3 × ALPHA A x^2 ALPHA ◢
√ 2 ÷ 3 × ALPHA A x^y 3

Tetrahedron: ALPHA ? → ALPHA A : √ 3 × ALPHA A x^2 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.

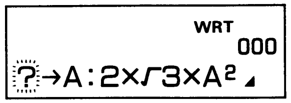
In actual practice, this would be performed as follows:

MODE 2



```
                WRT
                380
P _123456789
```

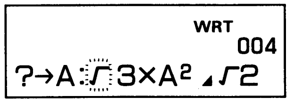
EXE



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

Cursor located at beginning. Press SHIFT EXE to bring cursor to end.

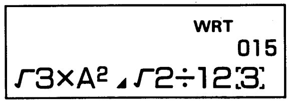
⇨ ⇨ ⇨ ⇨
DEL DEL



```
                WRT
                004
?→A:√3×A²◢√2
```

Locate cursor at position to be deleted, and delete two characters.

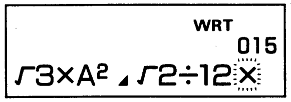
⇨ × 9 times
SHIFT INS 1 2



```
                WRT
                015
√3×A²◢√2÷12:3
```

Insert two characters.

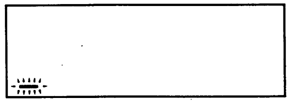
SHIFT INS DEL



```
                WRT
                015
√3×A²◢√2÷12×
```

Delete unnecessary 3.

MODE 1



```
_
```

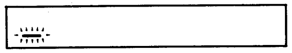
Editing complete. Return to the RUN mode.

④ Program execution

Now this program will be executed.

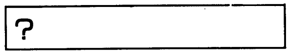
Length of one side (A)	Surface area (S)	Volume (V)
10 cm	(173.2050808)cm²	(117.8511302)cm³
7.5	(97.42785793)	(49.71844555)
20	(692.820323)	(942.8090416)

MODE 1



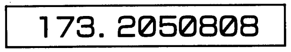
```
_
```

Prog 0 EXE



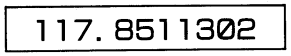
```
?
```

1 0 EXE



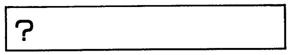
```
 173.2050808
```

EXE



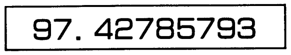
```
 117.8511302
```

Prog 0 EXE



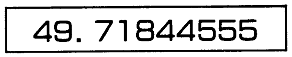
```
?
```

7 • 5 EXE



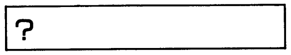
```
 97.42785793
```

EXE



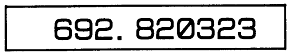
```
 49.71844555
```

Prog 0 EXE



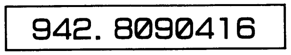
```
?
```

2 0 EXE



```
 692.820323
```

EXE



```
 942.8090416
```

⟨**Summary**⟩

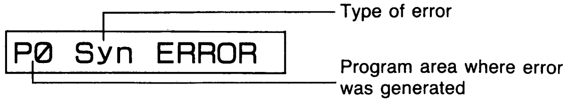
	Operation	Keys used
Program check	● WRT mode specification ● Program area specification (Omitted if P0) ● Start verification ● Verification of contents	MODE 2 ⇦ ⇨ EXE ⇦ ⇨
Correction	● Move the cursor to the position to be corrected. ● Press correct keys.	⇦ ⇨
Deletion	● Move the cursor to the position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

4-3 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
```

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 calculation 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)
Indicates a missing Lbl for the Goto command (see page 120), or that the program area (see page 117) for the Prog command (see page 127) does not contain a program.

④ **Ne ERROR** (Nesting error)
Indicates a subroutine nesting overflow by the Prog command (see page 127).

⑤ **Stk ERROR** (Stack error)
Indicates the calculation performed exceeds the capacity of the stack for numeric values or for commands (see page 31).

⑥ **Mem ERROR** (Memory error)
Indicates the attempt to use a memory name such as Z [5] without having expanded memories.

⑦ **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.
Pressing 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.

■ Checkpoints 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 check to see that the program in P *n* has been correctly input when Prog *n* is used.

④ **Ne ERROR**
Check to ensure that the Prog command is not used in the branched program area to return execution to the original program area.

⑤ **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.

⑥ **Mem ERROR**
Check to see that memories were properly expanded using "MODE • *n* EXE" (Defm). When using array-type memories (see page 130), check to see that the subscripts are correct.

⑦ **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.

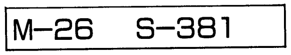
4-4 Counting the Number of Steps

The program capacity of this unit consists of a total of 400 steps.
The number of steps indicates the amount of storage space available for programs, and it will decrease as programs are input. The number of remaining steps will also be decreased when steps are converted to memories. (See page 40.)
There are two methods to determine the current number of remaining steps:

① When MODE • EXE are pressed in the RUN mode, the number of remaining steps will be displayed together with the number of memories.

Example

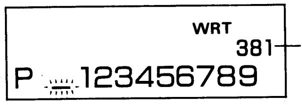
MODE • EXE



M-26 S-381

② 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

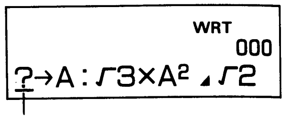
Number of remaining steps

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, (,), :, A, B, 1, 2, 3, etc.
- One function/two steps: Lbl 1, Goto 2, Prog 8, etc.

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

Example

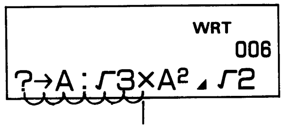


WRT
000
?→A:√3×A²◢√2

Present cursor position

At this time, each press of a cursor key (⇦ or ⇨) will cause the cursor to move to the next sequential step. For example:

⇨ ⇨ ⇨ ⇨ ⇨ ⇨



WRT
006
?→A:√3×A²◢√2

6th step

4-5 Program Areas and Calculation Modes

This unit contains a total of 10 program areas (P0 through P9) for the storage of programs. These 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.

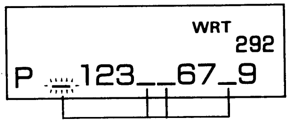
Example P0 Prog 0 EXE
P8 Prog 8 EXE

**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 programs.

Example



WRT
292
P 123__67_9

Programs already stored in these program areas.

■Program area and calculation mode specification in the WRT mode

Besides normal function calculations, to perform binary, octal, decimal and hexadecimal calculations and conversions, standard deviation calculations, 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**

MODE 2	WRT 400 P 0123456789	Assuming that nothing is stored.
⇨ ⇨	WRT 400 P 0123456789	Specify P2.
MODE –	BASE–N WRT 400 P 0123456789	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 calculation 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.

4-6 Erasing Programs

Erasing of programs is performed in the PCL mode. Press MODE 3 to specify the PCL mode. 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.**

MODE 3	PCL 302 P -12_45678_	P0, P3 and P9 already contain programs.
⇨ ⇨ ⇨	PCL 302 P _12-45678_	Align cursor with P3.
AC	PCL 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 then Mcl.

Example **Erase the programs stored in P0, P4, P8 and P9.**

MODE 3	PCL 273 P -123_567__
SHIFT Mcl	PCL 400 P 0123456789
MODE 1	-

4-7 Convenient Program Commands

The programs for this unit are made based upon manual calculations. Special progam 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 that 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 a 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 programs to return execution to the beginning for repetitive calculations, 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, x^2, ◢,
√‾, 2, ÷, 1, 2, ×, A, x^y, 3 19 steps

**Hereinafter, 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 "Goto 1".

The complete program with the unconditional jump added should look like this:

Lbl, 1, :, ?, →, A :, √‾, 3, ×, A, x^2, ◢,
√‾, 2, ÷, 1, 2, ×, A, x^y, 3, ◢, Goto, 1 25 steps

Now let's try executing this program.

**For details on inputting programs and editing programs, see sections 4-1 and 4-2.*
**Henceforth, the displays will only show calculation result output.*

Prog 0 EXE	?	Stored in P0. The length of the side = 10
1 0 EXE	173. 2050808	
EXE	117. 8511302	
EXE	?	
7 • 5 EXE	97. 42785793	The length of the side = 7.5
EXE	49. 71844555	
EXE	?	

Since the program is in an endless loop, it will continue execution. To terminate execution, press MODE 1.

MODE 1	- _ -

Besides the beginning of the program, branch destinations can be designated at any point within the program.

Example **Calculate $y = ax + b$ when the value for x changes each time, while a and b can also change depending upon the calculation.**

Program

?, →, 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 memory 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:

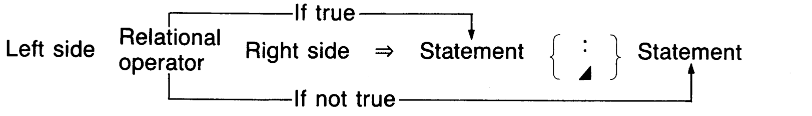
Left side Relational operator Right side ⇒ Statement $\left\{ \begin{matrix} : \\ ◢ \end{matrix} \right\}^*$ Statement

**Anyone can be used.*

One memory name (alphabetic character from A through Z), constant numeric values or calculation formulas (A×2, D—E, etc.) are used for "left side" and "right side".
The relational operator is a comparison symbol. There are 6 types of relational operators: =, ≠, ≥, ≤, >, <.

Left side = right side (left side equals right side)
Left side ≠ right side (left side does not equal right side)
Left side ≥ right side (left side is greater than or equal to right side)
Left side ≤ right side (left side is less than or equal to 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 the next ":" or "◢" is regarded as one statement.

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

Program

Lbl, 1, :, ?, →, A, :, A, ≥, Ø, ⇒, √‾, 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.

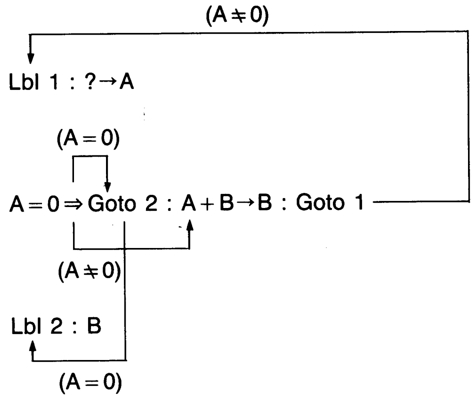
Example **Calculate the sum of input numeric values. If a 0 is input, the total should be displayed.**

Program

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

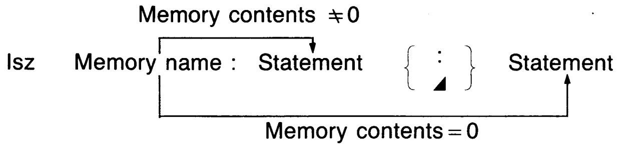
In this program, a 0 is first stored in memory B to clear it for calculation of the sum. Next, the value input by "?→A" is stored in memory A by "A=0⇒" and it is determined whether or not the value stored in memory A equals zero. If A=0, Goto 2 causes execution to jump to Lbl 2. If memory A does not equal 0, Goto 2 will be skipped and the command A+B→B which follows ":" is executed, and 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.

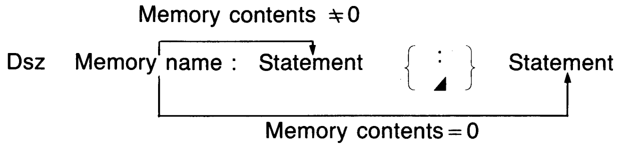
The following illustration shows the flow of the program:



■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 executed. The "Isz" command is used to increase the value in memory by 1 and decide the subsequent execution, 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

Example **Determine the average of 10 input numeric values.**

Program

1, Ø, →, A, :, Ø, →, C, :,
Lbl, 1, :, ?, →, B, :, B, +, C, →, C, :,
Dsz, A, :, Goto, 1, :, C, ÷, 1, Ø 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 a 0. 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 Vm/sec and an angle of S°.**

The formula is expressed as: $h = V\sin\theta t - \frac{1}{2}gt^2$, with $g = 9.8$, with the effects of air resistance being disregarded.

Program

Deg, :, Ø, →, T, :, ?, →, V, :, ?, →, S, :,
Lbl, 1, :, Isz, T, :, V, ×, sin, S, ×, T, —,
9, •, 8, ×, T, x^2, ÷, 2, ◢, Goto, 1 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 respectively.
Lbl 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→T" 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.

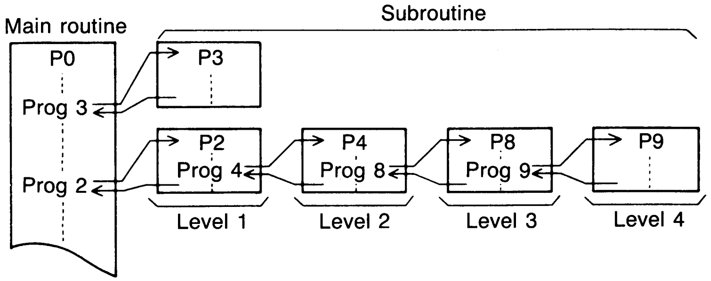
⟨**Summary**⟩

Command	Formula	Operation
Unconditional jump	Lbl *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, 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.

■Subroutines

A program contained in a single program area is called a "main routine". Often used program segments stored in other program areas are called "subroutines".

Subroutines can be used in a variety of ways to help make calculations easier. They can be used to store formulas for repeat calculations as one block to be jumped to each time, or to store often used formulas or operations for call up as required.



The subroutine command is "Prog" followed by a number from 0 through 9 which indicates the program area.

Example **Prog 0 .... Jump to program area 0**
Prog 2 .... Jump to program area 2

After the jump is performed using the Prog command, execution continues from the beginning of the program stored in the specified program area. After execution reaches the end of the subroutine, the program returns to the statement following the Prog *n* command in the original program area. Jumps can be performed from one subroutine to another, and this procedure is known as "nesting". Nesting can be performed to a maximum of 10 levels, and attempts to exceed this limit will cause an error (Ne ERROR) to be generated. Attempting to use Prog to jump to a program area in which there is no program stored will also result in an error (Go ERROR).

**A Goto n contained in a subroutine will jump to the corresponding Lbl n contained in that program area.*

Example **Simultaneously execute the two previously presented programs to calculate the surface areas and volumes of a regular octahedron and tetrahedron.**
Express the result in three decimal places.

This example employs two previously explained programs, and the first step is to input the specified number of decimal places (MODE 7 3).

Now let's review the two original programs.

Regular octahedron

P0 Fix, 3, :, ?, →, A, :, 2, ×, √‾, 3, ×, A, x^2, ◢,
√‾, 2, ÷, 3, ×, A, x^y, 3 23 steps

Regular tetrahedron

P1 Fix, 3, :, ?, →, A, :, √‾, 3, ×, A, x^2, ◢,
√‾, 2, ÷, 1, 2, ×, A, x^y, 3 22 steps
Total: 45 steps

If the two programs are compared, it is evident that the underlined portions are identical. If these portions are incorporated into a common subroutine, the programs are simplified and the number of steps required is decreased.
Furthermore, the portions indicated by the wavy line are not identical as they stand, but if P1 is modified to: √‾, 2, ÷, 3, ×, A, x^y, 3, ÷, 4, the two portions become identical.
Now the portions underlined by the straight line will be stored as an independent routine in P9 and those underlined with the wavy line will be stored in P8.

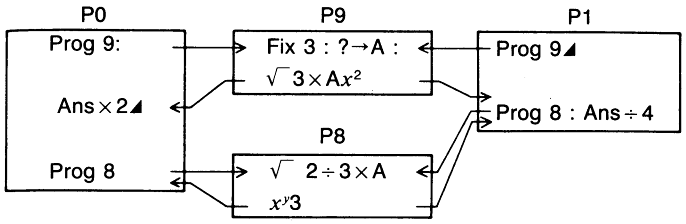
P9 Fix, 3, :, ?, →, A, :, √‾, 3, ×, A, x^2 12 steps
P8 √‾, 2, ÷, 3, ×, A, x^y, 3 8 steps

After the common segments have been removed, the remainder of the regular octahedron formula is stored in P0, and that of the regular tetrahedron is stored in P1. Of course, the "Prog 9" and "Prog 8" must be added to jump to subroutines P9 and P8.

P0 Prog, 9, :, Ans, ×, 2, ◢, Prog, 8 9 steps
P1 Prog, 9, ◢, Prog, 8, :, Ans, ÷, 4 9 steps
Total: 38 steps

With this configuration, execution jumps to program P9 at the beginning of programs P0 and P1, three decimal places are specified, the value for one side is entered, and the surface area of the tetrahedron is calculated. The expression "2×" of the original octahedron formula was omitted in P9, so when execution returns to P0, "Ans×2" is used to obtain the surface of the octahedron. In the case of P1, the result of P9 needs no further modification and so is immediately displayed upon return to P1.

Calculation of the volumes is also peformed in a similar manner. After a jump is made to P8 for calculation, execution returns to the main routines. In P0, the program ends after the volume of the octahedron is displayed. In P1, however, the result calculated in P8 is divided by four to obtain the volume of the tetrahedron. By using subroutines in this manner, steps can be shortened and programs become neat and easy to read.
The following illustration shows the flow of the program just presented.



By isolating the common portions of the two original programs and storing them in separate program areas, steps are shortened and programs take on a clear configuration.

4-8 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 subscripts are counted in relation to memory zero as shown above.

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

Using standard memories

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

Using array-type memories

```
Ø, →, Z, :, Lbl, 1, :, Z, +, 1, →, A, [, Z, ], :,
Isz, Z, :, Z, <, 1, Ø, ⇒, 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, Ø, ⇒, J, ◢,
Goto, 1                                        70 steps
```

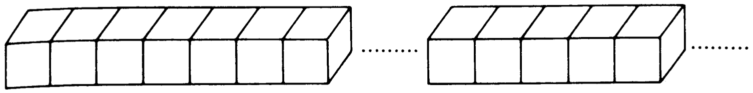
Using array-type memories

```
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:



```
A[ 0 ] A[ 1 ] A[ 2 ] A[ 3 ] A[ 4 ] A[ 5 ] A[ 6 ]      A[23] A[24] A[25] A[26] A[27]
B[-1 ] B[ 0 ] B[ 1 ] B[ 2 ] B[ 3 ] B[ 4 ] B[ 5 ]      B[22] B[23] B[24] B[25] B[26]
C[-2 ] C[-1 ] C[ 0 ] C[ 1 ] C[ 2 ] C[ 3 ] C[ 4 ]      C[21] C[22] C[23] C[24] C[25]
                                              :        :
                                              :        :
G[-6 ] G[-5 ] G[-4 ] G[-3 ] G[-2 ] G[-1 ] G[ 0 ]      G[17] G[18] G[19] G[20] G[21]
                                                       :
                                                       :
                                                      X[ 0 ] X[ 1 ] X[ 2 ] X[ 3 ] X[ 4 ]
                                                      Y[-1 ] Y[ 0 ] Y[ 1 ] Y[ 2 ] Y[ 3 ]
                                                      Z[-2 ] Z[-1 ] Z[ 0 ] Z[ 1 ] Z[ 2 ]
```

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	Ø.
EXE	3.
EXE	4.
EXE	5.

As can be seen, the second displayed value (which should be 2) in A[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 15 pieces 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) through C[30] (memory Z(7)).

```
1, →, A, :, Defm, 7, :,
Lbl, 1, :, ?, →, C, [, A, ], :,
?, →, C, [, A, +, 1, 5, ], :,
Isz, A, :, A, =, 1, 6, ⇒, Goto, 2, :, Goto, 1, :,
Lbl, 2, :, 1, 5, →, A, :, ?, →, B, :,
B, =, Ø, ⇒, Goto, 5, :,
Lbl, 3, :, B, =, C, [, A, ], ⇒, Goto, 4, :,
Dsz, A, :, Goto, 3, :, Goto, 2, :,
Lbl, 4, :, C, [, A, +, 1, 5, ], ◢, Goto, 2, :,
Lbl, 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, =, Ø, ⇒, Goto, 5, :,
Lbl, 3, :, B, =, C, [, A, ], ⇒, Goto, 4, :,
Dsz, A, :, Goto, 3, :, Goto, 2, :,
Lbl, 4, :, R, [, A, ], ◢, Goto, 2, :,
Lbl, 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 be 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, :, ...

4-9 Displaying 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 commands

0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
(,), √ , E, +, −, ×, ÷, ...
sin, cos, tan, log, ln, ...
=, ≠, ≥, ≤, >, <, ...
𝔸, 𝔹, ℂ, 𝔻, 𝔼, 𝔽, d, h, b, o
Neg, Not, and, or, xor
$\bar{x}$, $\bar{y}$, $x\sigma n$, $x\sigma n-1$, ...
° ([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 sorted in P1)

[Prog] [1] [EXE]	X=?
[1] [0] [EXE]	Y=?
⋮	⋮

Messages are also convenient when displaying 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=?
[4] [0] [9] [6] [EXE]	X=
[EXE]	12.
[EXE]	N=?
[3] [1] [2] [4] [EXE]	NO
[EXE]	N=?
[5] [1] [2] [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.

4-10 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 used 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

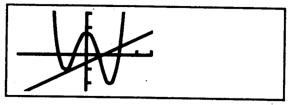
Finally program the equation for the second graph.

Graph, 1, 0, X, −, 3, 0   Total 49 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 procedure 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.

Program Library

Prime factor analysis
Definite integrals using Simpson's rule
Δ↔Y transformation
Minimum loss matching
Cantilever under concentrated load
Normal distribution
Graph variation by parameters
Hysteresis loop
Regression curve
Parade diagram

<Prior to use>

- Always check the number of remaining steps before attempting to store programs.
- The library is divided into two parts: a calculation section and a graph section. The calculation section shows only answers, while the graph section shows whole displays.
- Press the Graph key whenever "Graph" appears within a program ("Graph Y =" indicated).
- If it is necessary to specify a calculation mode (e.g. BASE-N, SD1) in a program, be sure to specify it after pressing MODE 2 (WRT mode).
  Then start programming by pressing EXE.

CASIO PROGRAM SHEET

Program for	Prime factor analysis	No. 1

Description

Prime factors of arbitrary positive integers are produced.

For $1 < m < 10^{10}$

prime numbers are produced from the lowest value first. "END" is displayed at the end of the program.

⟨Overview⟩

m is divided by 2 and by all successive odd numbers ($d = 3, 5, 7, 9, 11, 13, \ldots\ldots$) to check for divisibility.

Where d is a prime factor, $m_i = m_{i-1}/d$ is assumed, and division is repeated until $\sqrt{m_i} + 1 \leqq d$.

Example

⟨1⟩

$119 = 7 \times 17$

⟨2⟩

$1234567890 = 2 \times 3 \times 3 \times 5 \times 3607 \times 3803$

⟨3⟩

$987654321 = 3 \times 3 \times 17 \times 17 \times 379721$

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prog 0 EXE	M?	11	EXE	3803.
2	119 EXE	7.	12	EXE	END
3	EXE	17.	13	EXE	M?
4	EXE	END	14	987654321 EXE	3.
5	EXE	M?	15	EXE	3.
6	1234567890 EXE	2.	16	EXE	17.
7	EXE	3.	17	EXE	17.
8	EXE	3.	18	EXE	(After 15 seconds) 379721.
9	EXE	5.	19	EXE	END
10	EXE	(After 80 seconds) 3607.	20		

No. 1

Line	MODE 2 Program															Notes	Number of steps
1	Mcl	:															2
2	Lbl	0	:	"	M	"	?	→	A	:	Goto	2	:				15
3	Lbl	1	:	2	◢	A	÷	2	→	A	:	A	=	1	⇒		30
4	Goto	9	:														33
5	Lbl	2	:	Frac	(	A	÷	2	)	=	0	⇒	Goto	1	:		48
6	3	→	B	:													52
7	Lbl	3	:	√	A	+	1	→	C	:							62
8	Lbl	4	:	B	≥	C	⇒	Goto	8	:	Frac	(	A	÷	B		77
9	)	=	0	⇒	Goto	6	:										84
10	Lbl	5	:	B	+	2	→	B	:	Goto	4	:					96
11	Lbl	6	:	A	÷	B	×	B	−	A	=	0	⇒	Goto	7		111
12	:	Goto	5	:													115
13	Lbl	7	:	B	◢	A	÷	B	→	A	:	Goto	3	:			129
14	Lbl	8	:	A	◢												134
15	Lbl	9	:	"	E	N	D	"	◢	Goto	0						145
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents								
	A	m_i	H		O		V	
	B	d	I		P		W	
	C	$\sqrt{m_i}+1$	J		Q		X	
	D		K		R		Y	
	E		L		S		Z	
	F		M		T			
	G		N		U			

CASIO PROGRAM SHEET

Program for	Definite integrals using Simpson's rule	No.	2

Description

$$I=\int_a^b f(x)\,dx=\frac{h}{3}\{y_0+4(y_1+y_3+\ldots+y_{2m-1})+2(y_2+y_4+\ldots+y_{2m-2})+y_{2m}\}$$

$$h=\frac{b-a}{2m}$$

The right-hand portion of the above equation can be transformed as follows.

$$I=\frac{h}{3}\{y_0+\sum_{i=1}^{m}(4y_{2i-1}+2y_{2i})-y_{2m}\}$$

Let $f(x)=\dfrac{1}{x^2+1}$

Example

⟨1⟩ $a=0,\ b=1,\ 2m=10$

$$I=\int_0^1\frac{1}{x^2+1}d_x=0.7853981537$$

⟨2⟩ $a=2,\ b=5,\ 2m=20$

$$I=\int_2^5\frac{1}{x^2+1}d_x=0.2662526769$$

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prog 0 EXE	A?	11		
2	0 EXE	B?	12		
3	1 EXE	2M?	13		
4	10 EXE	Ø.7853981535	14		
5	EXE	A?	15		
6	2 EXE	B?	16		
7	5 EXE	2M?	17		
8	20 EXE	Ø.2662526768	18		
9			19		
10			20		

No.	2

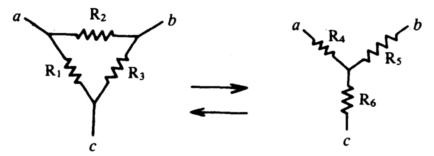
Line	MODE 2	Program														Notes	Number of steps
1	PØ																
2	Lbl	1	:	Mcl	:												5
3	"	A	"	?	→	A	:	"	B	"	?	→	B	:	"		20
4	2	M	"	?	→	M	:										27
5	A	→	G	:	Prog	1	:	P	→	I	:	(	B	−	A		42
6	)	÷	M	→	D	:	M	÷	2	→	O	:					54
7	Lbl	2	:	G	+	D	→	G	:	Prog	1	:	I	+	P		69
8	×	4	→	I	:												74
9	G	+	D	→	G	:	Prog	1	:	I	+	P	×	2	→		89
10	I	:	O	−	1	→	O	:									97
11	O	≠	Ø	⇒	Goto	2	:										104
12	B	→	G	:	Prog	1	:	I	−	P	→	I	:				117
13	D	×	I	÷	3	◢											123
14	Goto	1															125
15																	
16	P1																
17	1	÷	(	G	×	G	+	1	)	→	P						11
18																	
19																Total 136 steps	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents							
A	a	H		O	m (Number of repetitions)	V	
B	b	I	I	P		W	
C		J		Q		X	
D	$h=\frac{b-a}{2m}$	K		R		Y	
E		L		S		Z	
F		M	$2m$	T			
G	x	N		U			

CASIO PROGRAM SHEET

Program for **Δ↔Y transformation**　No. **3**

Description



1) Δ→Y　　　2) Y→Δ

$$R_4=\frac{R_1\cdot R_2}{R_1+R_2+R_3}\qquad R_1=\frac{R_4R_5+R_5R_6+R_6R_4}{R_5}$$

$$R_5=\frac{R_2\cdot R_3}{R_1+R_2+R_3}\qquad R_2=\frac{R_4R_5+R_5R_6+R_6R_4}{R_6}$$

$$R_6=\frac{R_3\cdot R_1}{R_1+R_2+R_3}\qquad R_3=\frac{R_4R_5+R_5R_6+R_6R_4}{R_4}$$

Example

〈1〉
$R_1=12$ (Ω)
$R_2=47$ (Ω)
$R_3=82$ (Ω)

〈2〉
$R_4=100$ (Ω)
$R_5=150$ (Ω)
$R_6=220$ (Ω)

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prog 0 EXE	D→Y:1,Y→D:2?	11	EXE	D→Y:1,Y→D:2?
2	1 EXE	R1=?	12	2 EXE	R4=?
3	12 EXE	R2=?	13	100 EXE	R5=?
4	47 EXE	R3=?	14	150 EXE	R6=?
5	82 EXE	R4=	15	220 EXE	R1=
6	EXE	4.	16	EXE	466.6666667
7	EXE	R5=	17	EXE	R2=
8	EXE	27.33333333	18	EXE	318.1818182
9	EXE	R6=	19	EXE	R3=
10	EXE	6.978723404	20	EXE	700.

No. **3**

Line	MODE 2	Program														Notes	Number of steps
1	Lbl	1	:	"	D	→	Y	:	1	,	Y	→	D	:	2		15
2	"	?	→	N	:												20
3	N	=	2	⇒	Goto	2	:	N	≠	1	⇒	Goto	1	:			34
4	"	R	1	=	"	?	→	A	:								43
5	"	R	2	=	"	?	→	B	:								52
6	"	R	3	=	"	?	→	C	:								61
7	A	+	B	+	C	→	D	:									69
8	"	R	4	=	"	◢	A	×	B	÷	D	◢					81
9	"	R	5	=	"	◢	B	×	C	÷	D	◢					93
10	"	R	6	=	"	◢	A	×	C	÷	D	◢					105
11	Goto	1	:														108
12	Lbl	2	:														111
13	"	R	4	=	"	?	→	E	:								120
14	"	R	5	=	"	?	→	F	:								129
15	"	R	6	=	"	?	→	G	:								138
16	E	×	F	+	F	×	G	+	G	×	E	→	H	:			152
17	"	R	1	=	"	◢	H	÷	F	◢							162
18	"	R	2	=	"	◢	H	÷	G	◢							172
19	"	R	3	=	"	◢	H	÷	E	◢							182
20	Goto	1															184
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

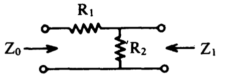
Memory contents	A	R_1	H	$R_4R_5+R_5R_6+R_6R_4$	O		V	
	B	R_2	I		P		W	
	C	R_3	J		Q		X	
	D	$R_1+R_2+R_3$	K		R		Y	
	E	R_4	L		S		Z	
	F	R_5	M		T			
	G	R_6	N	For judgement	U			

CASIO PROGRAM SHEET

Program for **Minimum loss matching** No. **4**

Description

Calculate R_1 and R_2 which match Z_0 and Z_1 with loss minimized. ($Z_0 > Z_1$)



$$R_1 = Z_0\sqrt{1-\frac{Z_1}{Z_0}} \qquad R_2 = \frac{Z_1}{\sqrt{1-\frac{Z_1}{Z_0}}}$$

$$\text{Minimum loss } L_{min} = 20\log\left(\sqrt{\frac{Z_0}{Z_1}} + \sqrt{\frac{Z_0}{Z_1}-1}\right)\text{[dB]}$$

Example

Calculate the values of R_1, R_2 and L_{min} for $Z_0 = 500\Omega$ and $Z_1 = 200\Omega$.

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prog 0 EXE	Z0=?	11		
2	500 EXE	Z1=?	12		
3	200 EXE	R1=	13		
4	EXE	387.2983346	14		
5	EXE	R2=	15		
6	EXE	258.1988897	16		
7	EXE	LMIN=	17		
8	EXE	8.961393328	18		
9			19		
10			20		

No. **4**

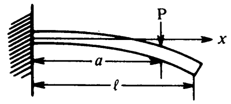
Line	MODE 2							Program									Notes	Number of steps
1	"	Z	0	=	"	?	→	Y	:									9
2	"	Z	1	=	"	?	→	Z	:									18
3	√	(	1	−	Z	÷	Y	)	→	A	:							29
4	Y	×	A	→	R	:	Z	÷	A	→	S	:	Y	÷	Z			44
5	→	B	:	2	0	×	log	(	√	B	+	√	(	B	−			59
6	1	)	)	→	T	:												65
7	"	R	1	=	"	◢	R	◢										73
8	"	R	2	=	"	◢	S	◢										81
9	"	L	M	I	N	=	"	◢	T									90
10																		
11																		
12																		
13																		
14																		
15																		
16																		
17																		
18																		
19																		
20																		
21																		
22																		
23																		
24																		
25																		
26																		
27																		
28																		

Memory contents								
	A	$\sqrt{1-\frac{Z_1}{Z_0}}$	H		O		V	
	B	$\frac{Z_0}{Z_1}$	I		P		W	
	C		J		Q		X	
	D		K		R	R_1	Y	Z_0
	E		L		S	R_2	Z	Z_1
	F		M		T	L_{min}		
	G		N		U			

CASIO PROGRAM SHEET

Program for **Cantilever under concentrated load**	No. **5**

Description



E : Young's modulus [kg/mm²]
I : Geometrical moment of inertia [mm⁴]
a : Distance of concentrated load from support [mm]
P : Load [kg]
x : Distance of point of interest from the support [mm]

Deflection y [mm], Angle of deflection s [°], Bending moment M [kg·mm]

① $\ell > x > a$

$$y = \frac{Pa^3}{6EI} - \frac{Pa^2}{2EI}x$$

$$s = \tan^{-1}\left[-\frac{Pa^2}{2EI}\right]$$

M = 0 (shearing load Ws = 0)

② $x \leqq a$

$$y = \frac{P}{6EI}x^3 - \frac{Pa}{2EI}x^2$$

$$s = \tan^{-1}\left[\frac{Px}{2EI}(x-2a)\right]$$

M = P $(x-a)$ (shearing load Ws = P)

Example

E = 4000 kg/mm²
I = 5 mm⁴
a = 30 mm
P = 2 kg

What are deflection, angle of deflection, bending moment and shearing load at x = 25 mm and x = 32 mm?

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prog 0 EXE	E=?	11	EXE	−10.
2	4000 EXE	I =?	12	EXE	X = ?
3	5 EXE	A=?	13	32 EXE	Y=
4	30 EXE	P=?	14	EXE	−0.99
5	2 EXE	X=?	15	EXE	S=
6	25 EXE	Y=	16	EXE	−2.57657183
7	EXE	−0.677083333	17	EXE	M=
8	EXE	S=	18	EXE	0.
9	EXE	−2.505092867	19	Repeat from step 5.	
10	EXE	M =	20		

No. **5**

Line	MODE 2							Program								Notes	Number of steps
1	Deg	:	"	E	=	"	?	→	E	:	"	I	=	"	?		15
2	→	I	:	"	A	=	"	?	→	A	:	"	P	=	"		30
3	?	→	P	:													34
4	Lbl	1	:	"	X	=	"	?	→	X	:						45
5	X	≤	A	⇒	Goto	2	:										52
6	"	Y	=	"	◢	P	×	A	x^2	÷	(	2	×	E	×		67
7	I	)	×	(	A	÷	3	−	X	)	◢						78
8	"	S	=	"	◢	$\tan^{-1}$	(	(−)	P	×	A	x^2	÷	(	2		93
9	×	E	×	I	)	)	◢	"	M	=	"	◢	0	◢			107
10	Goto	1	:														110
11	Lbl	2	:														113
12	"	Y	=	"	◢	P	×	X	x^2	÷	(	2	×	E	×		129
13	I	)	×	(	X	÷	3	−	A	)	◢						139
14	"	S	=	"	◢	$\tan^{-1}$	(	P	×	X	÷	(	2	×	E		154
15	×	I	)	×	(	X	−	2	×	A	)	)	◢				167
16	"	M	=	"	◢	P	×	(	X	−	A	)	◢				180
17	Goto	1															182
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents							
A	a	H		O		V	
B		I	I	P	P	W	
C		J		Q		X	x
D		K		R		Y	
E	E	L		S		Z	
F		M		T			
G		N		U			

CASIO PROGRAM SHEET

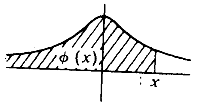
Program for Normal distribution	No. 6

Description

Obtain normal distribution function $\phi\ (x)$ (by Hastings' best approximation).

$$\phi\ (x)=\int_{-\infty}^{t} \phi t dx$$

$$\phi t=\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$



Put $t=\frac{1}{1+\mathrm{P}x}$

$\phi\ (x) \fallingdotseq 1-\phi t\ (c_1 t+c_2 t^2+c_3 t^3+c_4 t^4+c_5 t^5)$

$\mathrm{P}=0.2316419$

$C_1=0.31938153$

$C_2=-0.356563782$

$C_3=1.78147937$

$C_4=-1.821255978$

$C_5=1.330274429$

Example

Calculate the values of $\phi\ (x)$ at $x=1.18$ and $x=0.7$.

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prog 0 EXE	X = ?	11		
2	1.18 EXE	PX =	12		
3	EXE	0.880999696	13		
4	EXE	X = ?	14		
5	0.7 EXE	PX =	15		
6	EXE	0.7580361368	16		
7			17		
8			18		
9			19		
10			20		

No. 6

Line	MODE 2							Program								Notes	Number of steps
1	"	X	=	"	?	→	X	:									8
2	1	÷	(	1	+	0	.	2	3	1	6	4	1	9	×		23
3	X	)	→	T	:	1	÷	√	(	2	×	π	)	×	e^x		38
4	(	(−)	X	x^2	÷	2	)	→	Q	:							48
5	0	.	3	1	9	3	8	1	5	3	→	A	:				61
6	(−)	0	.	3	5	6	5	6	3	7	8	2	→	B	:		76
7	1	.	7	8	1	4	7	9	3	7	→	C	:				89
8	(−)	1	.	8	2	1	2	5	5	9	7	8	→	D	:		104
9	1	.	3	3	0	2	7	4	4	2	9	→	E	:			118
10	"	P	X	=	"	◢	1	−	Q	(	A	T	+	B	T		133
11	x^2	+	C	T	x^y	3	+	D	T	x^y	4	+	E	T	x^y		148
12	5	)															150
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents								
	A		H		O		V	
	B		I		P		W	
	C		J		Q	ϕt	X	x
	D		K		R		Y	
	E		L		S		Z	
	F		M		T	t		
	G		N		U			

CASIO PROGRAM SHEET

Program for	Graph variation by parameters	No. 7

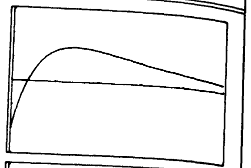
Description

Damped vibration

(i) $\varepsilon > n$ (Overdamping)

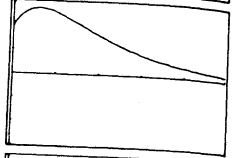
$$P_1 = -\varepsilon + \sqrt{\varepsilon^2 - n^2}, \quad P_2 = -\varepsilon - \sqrt{\varepsilon^2 - n^2}$$

$$x = \frac{v_0 - x_0P_2}{P_1 - P_2} e^{P1t} - \frac{v_0 - x_0P_1}{P_1 - P_2} e^{P2t}$$



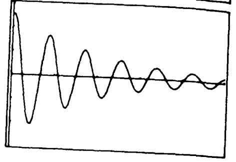
(ii) $\varepsilon = n$ (Critical damping)

$$x = \{x_0 + (v_0 + \varepsilon x_0)t\} e^{-\varepsilon t}$$



(iii) $\varepsilon < n$ (Damping vibration)

$$x = e^{-\varepsilon t} \{x_0 \cos\sqrt{n^2 - \varepsilon^2}\, t + \frac{v_0 + \varepsilon x_0}{\sqrt{n^2 - \varepsilon^2}} \cdot \sin\sqrt{n^2 - \varepsilon^2}\, t\}$$



Example

Draw a graph of the damping vibration that possesses the following parameters:

(1)	(2)	(3)
$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.2$
$n = 1.5$	$n = 0.2$	$n = 0.18$
$x_0 = 2.5$	$x_0 = 2$	$x_0 = -2$
$v_0 = 1$	$v_0 = 0.6$	$v_0 = 1.5$

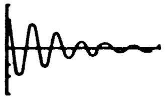
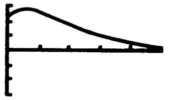
Preparation and operation

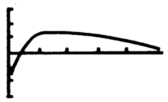
•Store the program written on the next page.

Memory contents								
	A	x_0	H		O		V	
	B	v_0	I		P	$P_1 = -\varepsilon + \sqrt{\varepsilon^2 - n^2}$	W	
	C	$\sqrt{n^2 - \varepsilon^2}$	J		Q	$P_2 = -\varepsilon - \sqrt{\varepsilon^2 - n^2}$	X	t
	D		K		R		Y	x
	E	ε	L		S		Z	
	F		M		T			
	G		N	n	U			

No. 7

Line	MODE 2															Notes	Number of steps
1	Rad	:															2
2	Range	0	,	2	5	,	5	,	(−)	3	,	3	,	1	:		17
3	"	E	P	S	I	L	O	N	=	"	?	→	E	:			31
4	"	N	=	"	?	→	N	:									39
5	"	X	0	=	"	?	→	A	:								48
6	"	V	0	=	"	?	→	B	:								57
7	E	>	N	⇒	Goto	1	:										64
8	E	=	N	⇒	Goto	2	:										71
9	√	(	N	x^2	−	E	x^2	)	→	C	:						82
10	Graph	e^x	(	(−)	E	X	)	(	A	cos	(	C	X	)	+		97
11	(	B	+	E	A	)	C	x^{-1}	sin	(	C	X	)	)	:		112
12	Goto	0	:														115
13	Lbl	1	:														118
14	(−)	E	+	√	(	E	x^2	−	N	x^2	)	→	P	:			132
15	(−)	E	−	√	(	E	x^2	−	N	x^2	)	→	Q	:			146
16	Graph	(	B	−	A	Q	)	(	P	−	Q	)	x^{-1}	e^x	(		161
17	P	X	)	−	(	B	−	A	P	)	(	P	−	Q	)		176
18	x^{-1}	e^x	(	Q	X	)	:										183
19	Goto	0	:														186
20	Lbl	2	:														189
21	Graph	(	A	+	(	B	+	E	A	)	X	)	e^x	(	(−)		204
22	E	X	)	:													208
23	Lbl	0															210
24																	
25																Total 210 steps	
26																	
27																	
28																	
29																	
30																	
31																	
32																	
33																	
34																	
35																	

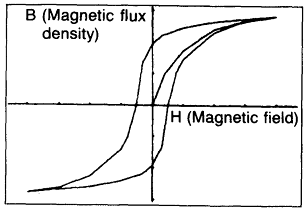
Program for Graph variation by parameters		No. 7
Step	Key operation	Display
1	Prog 0 EXE	EPSILON=?
2	0.1 EXE	N=?
3	1.5 EXE	X0=?
4	2.5 EXE	V0=?
5	1 EXE	
6	EXE	EPSILON=?
7	0.2 EXE	N=?
8	0.2 EXE	X0=?
9	2 EXE	V0=?
10	0.6 EXE	

Program for Graph variation by parameters		No. 7
Step	Key operation	Display
11	EXE	EPSILON=?
12	0.2 EXE	N=?
13	0.18 EXE	X0=?
14	SHIFT (−) 2 EXE	V0=?
15	1.5 EXE	
16		
17		
18		
19		
20		

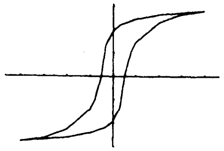
CASIO PROGRAM SHEET

Program for	No.
Hysteresis loop	8

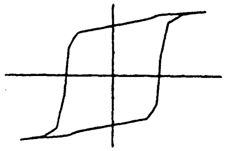
Description



When a ferromagnetic specimen is sustained in a magnetic field, the specimen becomes magnetized. The B-H relationship can be represented by a hysteresis curve.



Soft magnetic substance



Ferromagnetic substance

Example

Hysteresis curve of soft magnetic material

	1	2	3	4	5	6	7	8	9
H	0.4	1.0	2.0	3.0	4.0	2.0	1.0	0.5	0.3
B	0.5	0.86	1.2	1.32	1.4	1.31	1.22	1.13	1.1

- Number of data items: 17
- Number of data items in the main loop: 12

	10	11	12	13	14	15	16	17
H	0	−0.3	−0.5	−0.8	−1.0	−2.0	−3.0	−4.0
B	0.96	0.66	0	−0.53	−0.72	−1.15	−1.33	−1.4

- Within 20 data items.

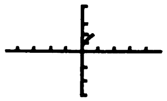
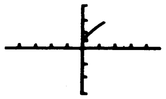
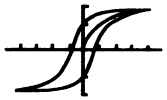
Preparation and operation

- Store the program written on the next page.

Memory contents									
	A	Number of data items	H		O		V		
	B	Number of data items in the main loop	I		P		W		
	C		J		Q		X		
	D		K		R		Y		
	E		L		S		Z		
	F		M		T			Z[1]~Z[20] B	
	G	F[1]~F[20] H	N		U				

		No.	8

Line	MODE 2							Program								Notes	Number of steps
1	Range	(−)	4	.	7	,	4	.	7	,	1	,	(−)	1	.		15
2	5	5	,	1	.	5	5	,	0	.	5	:					27
3	Defm	2	0	:													31
4	"	N	O	.	O	F	SPACE	D	A	T	A	"	?	→	A		46
5	. :																47
6	Lbl	9	:														50
7	"	M	A	I	N	SPACE	L	O	O	P	SPACE	N	"	?	→		65
8	B	:															67
9	B	>	2	0	⇒	Goto	9	:									75
10	1	→	C	:	Plot	0	,	0	:								84
11	Lbl	0	:	"	H	=	"	?	→	F	[	C	]	:			98
12	"	B	=	"	?	→	Z	[	C	]	:						109
13	Plot	F	[	C	]	,	Z	[	C	]	:	Line	◢				122
14	C	+	1	→	C	:											128
15	C	≠	A	+	1	⇒	Goto	0	:								137
16	A	−	B	+	1	→	D	:									145
17	Lbl	1	:	Plot	(−)	F	[	D	]	,	(−)	Z	[	D	]		160
18	:	Line	:														163
19	D	+	1	→	D	:											169
20	D	≠	A	+	1	⇒	Goto	1	:								178
21	"	E	N	D	"												183
22																	
23													Memory 20 × 8 = 160				
24																	
25																Total 343 steps	
26																	
27																	
28																	
29																	
30																	
31																	
32																	
33																	
34																	
35																	

Program for Hysteresis loop		No. 8
Step	Key operation	Display
1	Prog 0 EXE	NO.OF DATA?
2	17 EXE	MAIN LOOP N?
3	12 EXE	H=?
4	0.4 EXE	B=?
5	0.5 EXE	 Disp
6	EXE	H=?
7	1.0 EXE 0.86 EXE	 Disp
8	(Input data in order) ⋮	
9	EXE	END
10	G-T	

Program for		No.
Step	Key operation	Display

CASIO PROGRAM SHEET

Program for	Regression curve	No.	9

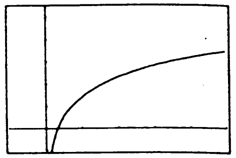
Description

i Logarithmic regression curve

Regression formula: $y = \mathrm{A} + \mathrm{B}\ln x$

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

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

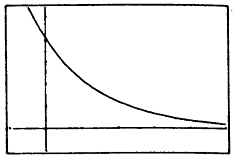


ii Exponential regression curve

Regression formula: $y = \mathrm{A}\cdot e^{\mathrm{B}x}$

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

$$\mathrm{A} = e\left(\frac{\Sigma\ln y - \mathrm{B}\cdot\Sigma x}{n}\right)$$

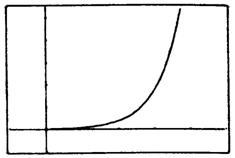


iii Power regression curve

Regression formula: $y = \mathrm{A}\cdot x^{\mathrm{B}}$

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

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



*See page 162 for an example.

Preparation and operation

- Store the program written on the next page.

Memory contents	A	A or lnA	H	$\Sigma(\ln x)^2$	O		V	Σx
	B	B	I		P	Σy^2	W	n
	C	$\Sigma\ln x$	J		Q	Σy	X	x data
	D	$\Sigma\ln y$	K		R	Σxy	Y	y data
	E	$\mathrm{X}\Sigma\ln y$	L		S	For selection of 1~3	Z	
	F	$\mathrm{Y}\Sigma\ln x$	M		T			
	G	$\Sigma(\ln x\cdot\ln y)$	N		U	Σx^2		

No.	9

Line	MODE 2	Program														Notes	Number of steps
1	P0	SHIFT	MODE	÷	→	LR	2										
2	Scl	:	Cls	:	0	→	C	~	H	:							10
3	"	Range	O	K	?	"	◢										17
4	"	A	C	→	Prog	1	SPACE	X	:	?	"	:					29
5	Lbl	1	:														32
6	"	X	:	"	?	→	X	:									40
7	"	Y	:	"	?	→	Y	:									48
8	ln	X	+	C	→	C	:	ln	Y	+	D	→	D	:	X		63
9	ln	Y	+	E	→	E	:	Y	ln	X	+	F	→	F	:		78
10	ln	X	×	ln	Y	+	G	→	G	:	(	ln	X	)	x^2		93
11	+	H	→	H	:												98
12	X	,	Y	DT	◢												103
13	Goto	1															105
14																	
15	P1	MODE	+	→	COMP												
16	"	L	→	1	SPACE	E	→	2	SPACE	P	→	3	:	"	?		15
17	→	S	:														18
18	S	=	1	⇒	Prog	7	:										25
19	S	=	2	⇒	Prog	8	:										32
20	S	=	3	⇒	Prog	9	:										39
21	"	E	N	D	"												44
22																	
23	P7	SHIFT	MODE	÷	→	LR	2										
24	(	W	F	−	C	Q	)	(	W	H	−	C	x^2	)	x^{-1}		15
25	→	B	:	(	Q	−	B	C	)	W	x^{-1}	→	A	:			29
26	Graph	A	+	B	ln	X	◢										36
27	"	A	:	"	◢	A	◢										43
28	"	B	:	"	◢	B	◢										50
29																	
30																	
31																	
32																	
33																	
34																	
35																	

CASIO PROGRAM SHEET

Program for	No.
Regression curve	9

Example

Perform exponential regression of the following data:

xi	2.2	5.6	9.5	13.8	18.0	23.2	29.9	37.8
yi	35.6	28.1	23.0	17.9	12.9	10.2	6.2	4.0

Draw an exponential regression curve, and use the trace function to estimate the value for y when $x = 20$. Also, obtain the values of A and B of the regression formula.

Range values:

X min : −10	Y min : −10
X max : 50	Y max : 55
X scl : 10	Y scl : 10

Preparation and operation

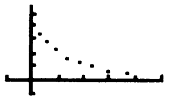
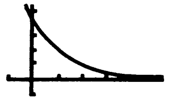
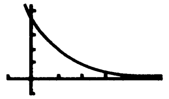
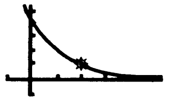
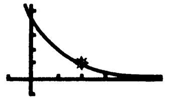
- Store the program written on the next page.

Memory contents								
	A		H		O		V	
	B		I		P		W	
	C		J		Q		X	
	D		K		R		Y	
	E		L		S		Z	
	F		M		T			
	G		N		U			

No.	9

Line	MODE 2							Program								Notes	Number of steps
1	P8	SHIFT	MODE	÷	→	LR	2										
2	(	W	E	−	V	D	)	(	W	U	−	V	x^2	)	x^{-1}		15
3	→	B	:	(	D	−	B	V	)	W	x^{-1}	→	A	:			29
4	Graph	e^x	A	×	e^x	B	X	◢									37
5	"	A	:	"	◢	e^x	A	◢									45
6	"	B	:	"	◢	B	◢										52
7																	
8	P9	SHIFT	MODE	÷	→	LR	2										
9	(	W	G	−	C	D	)	(	W	H	−	C	x^2	)	x^{-1}		15
10	→	B	:	(	D	−	B	C	)	W	x^{-1}	→	A	:			29
11	Graph	e^x	A	×	X	x^y	B	◢									37
12	"	A	:	"	◢	e^x	A	◢									45
13	"	B	:	"	◢	B	◢										52
14																	
15																Total 303 steps	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
29																	
30																	
31																	
32																	
33																	
34																	
35																	

Program for **Regression curve** No. **9**

Step	Key operation	Display
1	[Prog] 0 [EXE] (Range setting check)	LR2 Disp Range OK?
2	Set range values [AC] [Range]	LR2 Xmin? −3.8
	[SHIFT] [(−)] 10 [EXE] 50 [EXE] 10 [EXE] [SHIFT] [(−)] 10 [EXE] 55 [EXE] 10	
3	[EXE] [Prog] 0 [EXE]	LR2 Range OK?
4	[EXE]	LR2 X : ?
5	2.2 [EXE]	LR2 Y : ?
6	35.6 [EXE]	LR2 Disp 2.2
7	[EXE]	LR2 X : ?
8	(Input data in order) ⋮	
9	4.0 [EXE]	LR2 Disp 37.8
10	[G↔T]	 Disp
11	[AC] [Prog] 1 [EXE]	L→1 E→2 P→3?
12	2 [EXE] (Select exponential regression).	 Disp
13	[Trace]	 X= Disp −5.26315
14	[⇨] ~ Move pointer to X=20	 X= Disp 20.
15	[SHIFT] [X↔Y]	 Y= Disp 11.86149
16	[EXE]	LR2 Disp A :
17	[EXE]	LR2 Disp 40.68214076
18	[EXE]	LR2 Disp B :
19	[EXE]	LR2 Disp −0.061624605

Program for Regression curve	No. 9

Step	Key operation	Display
20	EXE	LR2 END
21		
22		
23		
24		
25		
26		
27		
28		
29		

Program for	No.

Step	Key operation	Display

CASIO PROGRAM SHEET

Program for	No.
Parade diagram	**10**

Description

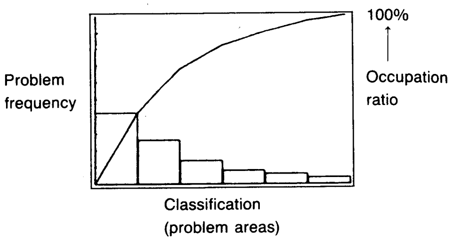
One example of a parade diagram application is problem solving in QC activities. The problem is quantitatively analyzed based on actual data concerning its extent, and the main points demanding attention are determined.

Horizontal axis: Problem classification

(Item 6 in this example)

Vertical axis: (Right) Occupation ratio

(Left) Problem extent in each classification



Example

Create a parade diagram using the data on the right.

Problem areas	Frequency
A	105
B	65
C	35
D	20
E	15
Others	10

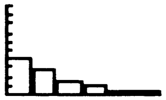
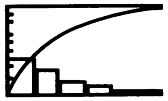
Preparation and operation

- •Store the program written on the next page.

Memory contents	A	Input data	H		O		V	
	B		I		P		W	n
	C		J		Q		X	Count of data
	D		K		R		Y	
	E		L		S	Display count	Z	Sum of data
	F		M		T			Z[1]~Z[6]
	G		N		U			

No. **10**

Line	MODE 2						Program									Notes	Number of steps
1	P0	SHIFT	MODE	☒	→	SD2											
2	Scl	:	Mcl	:	Defm	6	:										7
3	Range	0	,	6	,	1	,	0	,	2	0	,	2	:			21
4	Lbl	1	:														24
5	"	D	A	T	A	"	?	→	A	:							34
6	X	;	A	DT	:												39
7	X	+	1	→	X	:	X	≤	5	⇒	Goto	1	:				52
8	Range	,	,	,	,	W	,	W	÷	1	0	:	·				64
9	Graph	◢															66
10	Plot	0	,	0	:												71
11	1	→	S	:													75
12	Lbl	2	:														78
13	Z	[	S	]	+	Z	→	Z	:								87
14	Plot	S	,	Z	:	Line	:										94
15	S	+	1	→	S	:	S	≤	6	⇒	Goto	2	:				107
16	Graph	W															109
17																	
18																	
19																	
20																Memory 6×8=48	
21																Total 157 steps	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
29																	
30																	
31																	
32																	
33																	
34																	
35																	

Program for	Parade diagram	No. 10
Step	**Key operation**	**Display**
1	[Prog] 0 [EXE]	SD2 DATA?
2	105 [EXE]	SD2 DATA?
3	65 [EXE]	SD2 DATA?
4	(Input data in order.) ⋮	
5	10 [EXE] (Bar graph display)	 Disp
6	[EXE] (Parade diagram display)	
7		
8		
9		
10		

Appendix

Function Reference

■Manual Calculations

Mode specification	COMP Mode (MODE +)	Four arithmetic and function calculations.
	BASE-N Mode (MODE −)	Binary, octal, decimal, hexadecimal conversions and calculations, logical operations.
	SD1 Mode (MODE ×)	Standard deviation calculations (1-variable statistical).
	LR1 Mode (MODE ÷)	Regression calculations (paired variable statistical).
	SD2 Mode (SHIFT MODE ×)	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 functions	Function command input immediately after numeric value. [x^2, x^{-1}, $x!$, °, ^{r}, ^{g}, ° ' "]
	Type B functions	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, ln, e^x, 10^x, $\sqrt{\ }$, $\sqrt[3]{\ }$, Abs, Int, Frac, etc.
	Paired variable functions	Function command input between two numeric values. Numeric value enclosed in parentheses input immediately after function command. 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, $\overleftarrow{\text{ENG}}$, $\overleftarrow{\text{° ' "}}$]

Binary, octal, decimal, hexadecimal calculations MODE −	Setting number system	Decimal .... Dec EXE Hexadecimal ...... Hex EXE Binary .... 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 Hexadecimal ...... SHIFT h Binary .... SHIFT b Octal .... SHIFT o
	Logical operations	Input numeric values are converted to binary and each bit is tested. Result is converted back to number system used for input, and then displayed. Not .... Reverse of each bit and .... Logical product of each bit or .... Logical sum of each bit xor .... Exclusive logical sum of each bit xnor .... Exclusive negative logical sum of each bit
Standard deviation calculations MODE ×	Data clear	SHIFT Scl EXE
	Data input	Data [;frequency] DT **Frequency can be omitted.*
	Data deletion	Data [;frequency] CL **Frequency can be omitted.*
	Result display	Number of data (n) .... ALPHA 3 (n) EXE Sum (Σx) .... ALPHA 2 (Σx) EXE Sum of squares (Σx^2) .... ALPHA 1 (Σx^2) EXE Mean ($\bar{x}$) .... SHIFT $\bar{x}$ EXE Population standard deviation ($x\sigma_n$) .... SHIFT $x\sigma_n$ EXE Sample standard deviation ($x\sigma_{n-1}$) .... SHIFT $x\sigma_{n-1}$ EXE

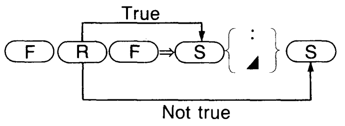
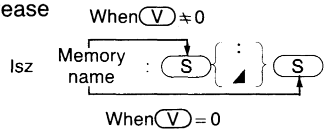
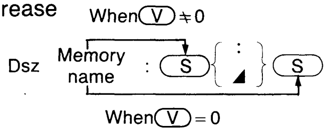
Regression calculations MODE ÷	Data clear	SHIFT Scl EXE
	Data input	x data, y data [;frequency] DT **Frequency can be omitted.*
	Data deletion	x data, y data [;frequency] CL **Frequency can be omitted.*
	Result display	Number of data (n) ... ALPHA 3 (n) EXE Sum of x (Σx) .... ALPHA 2 (Σx) EXE Sum of y (Σy) .... ALPHA 5 (Σy) EXE Sum of squares of x (Σx^2) .... ALPHA 1 (Σx^2) EXE Sum of squares of y (Σy^2) .... ALPHA 4 (Σy^2) EXE Sum of products of x and y (Σxy) .... ALPHA 6 (Σxy) EXE Mean of x ($\bar{x}$) .... SHIFT $\bar{x}$ EXE Mean of y ($\bar{y}$) .... SHIFT $\bar{y}$ EXE Population standard deviation of x ($x\sigma_n$) .... SHIFT $x\sigma_n$ EXE Population standard deviation of y ($y\sigma_n$) .... SHIFT $y\sigma_n$ EXE Sample standard deviation of x ($x\sigma_{n-1}$) .... SHIFT $x\sigma_{n-1}$ EXE Sample standard deviation of y ($y\sigma_{n-1}$) .... SHIFT $y\sigma_{n-1}$ 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 ($\hat{x}$) .... SHIFT $\hat{x}$ EXE Estimated value of y ($\hat{y}$) .... SHIFT $\hat{y}$ EXE

Special functions	Ans	The latest result obtained in manual or program calculations is stored in memory. It is recalled by pressing Ans. **Mantissa of numeric value is 10 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.
	Multistatement	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 26. Memories can be expanded in units of one up to 50 (for a total of 76). Eight steps are 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	Zoom × f .... Zooms in on the graph in accordance with the zoom factors Zoom × $^1/_f$ ... Zooms out on the graph in accordance with the inverse of the zoom factors Zoom Org .. Returns zoomed graph to original dimensions
	Scroll	Scrolls screen to view parts of graphs that are off the display.

■Program Calculations

Program input	Input mode	WRT Mode (MODE 2)
	Calculation mode	Mode that conforms with program specified by: MODE +, MODE −, MODE ×, MODE ÷.
	Program area specification	Cursor is moved to the desired program area name (P0 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
Program editing	Input mode	WRT Mode (MODE 2)
	Program area specification	Cursor is moved to the desired program area name (P0 through P9) using ⇨ or ⇦, 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 (P0 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, execution 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 "◢". Increase  When(V)≠0 Isz Memory name : S : ◢ S When(V)=0 Decrease  When(V)≠0 Dsz Memory name : S : ◢ S When(V)=0 (S): Statement (V): Value in memory
	Subroutines	Program execution jumps from main routine to subroutine indicated by Prog *n* (*n* = 0 through 9). After execution of the subroutine, execution returns to the point following Prog *n* in the original program area.

Error Message Table

Message	Meaning	Countermeasure
Syn ERROR	①Calculation formula contains an error. ②Formula in a program contains an error.	①Use ⇦ or ⇨ to display the point where the error was generated and correct it. ②Use ⇦ or ⇨ to display the point where the error was generated, press AC and then correct the program in the WRT Mode.
Ma ERROR	①Calculation result exceeds calculation range. ②Calculation is performed outside the input range of a function. ③Illogical operation (division by zero, etc.)	①②③ Check the input numeric value and correct it. When using memories, check that the numeric values stored in memories are correct.
Go ERROR	①No corresponding Lbl *n* for Goto *n*. ②No program stored in program area P *n* which corresponds to Prog *n*.	①Correctly input a Lbl *n* to correspond to the Goto *n*, or delete the Goto *n* if not required. ②Store a program in program area P *n* to correspond to Prog *n*, or delete the Prog *n* if not required.
Ne 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 Prog *n*. •Trace the subroutine jump destinations and ensure that no jumps are made back to the original program area. Ensure that returns are made correctly.

Stk 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 value input for Defm, value other than 1 ~ 9 input for n, etc.	•Re-enter argument correctly.

Input Ranges of Functions

Function	Input range	Internal digits	Accuracy	Notes
sin$x$ cos$x$ tan$x$	(Deg) $\|x\| < 9 \times 10^{9}$° (Rad) $\|x\| < 5 \times 10^{7}\pi$ rad (Gra) $\|x\| < 1 \times 10^{10}$ grad	12 digits	As a rule, accuracy is ±1 at the 10th digit.	However, for tan x: $\|x\| \neq 90(2n+1)$: Deg $\|x\| \neq \pi/2(2n+1)$: Rad $\|x\| \neq 100(2n+1)$: Gra
$\sin^{-1}x$ $\cos^{-1}x$	$\|x\| \leqq 1$	,,	,,	
$\tan^{-1}x$	$\|x\| < 1 \times 10^{100}$			
sinh$x$ cosh$x$	$\|x\| \leqq 230.2585092$	,,	,,	Note: For sinh and tanh, when $x=0$, errors are cumulative and accuracy is affected at a certain point.
tanh$x$	$\|x\| < 1 \times 10^{100}$			
$\sinh^{-1}x$	$\|x\| < 5 \times 10^{99}$	,,	,,	
$\cosh^{-1}x$	$1 \leqq x < 5 \times 10^{99}$			
$\tanh^{-1}x$	$\|x\| < 1$			
log$x$ ln$x$	$1 \times 10^{-99} \leqq x < 1 \times 10^{100}$	,,	,,	
10^{x}	$-1 \times 10^{100} < x < 100$	,,	,,	
e^{x}	$-1 \times 10^{100} < x \leqq 230.2585092$			
$\sqrt{x}$	$0 \leqq x < 1 \times 10^{100}$	,,	,,	
x^2	$\|x\| < 1 \times 10^{50}$			
$1/x$	$\|x\| < 1 \times 10^{100}$, $x \neq 0$	,,	,,	
$\sqrt[3]{x}$	$\|x\| < 1 \times 10^{100}$			
$x!$	$0 \leqq x \leqq 69$ (x is an integer)	,,	,,	

Function	Input range	Internal digits	Accuracy	Notes
Pol (x,y)	$\sqrt{x^2+y^2}<1\times10^{100}$	12 digits	As a rule, accuracy is ±1 at the 10th digit.	
Rec (r,θ)	$0\leqq r<1\times10^{100}$ (Deg) $\lvert\theta\rvert<9\times10^{9}$° (Rad) $\lvert\theta\rvert<5\times10^{7}\pi$rad (Gra) $\lvert\theta\rvert<1\times10^{10}$ grad	”	”	However, for tan θ: $\lvert\theta\rvert\fallingdotseq 90(2n+1)$: Deg $\lvert\theta\rvert\fallingdotseq\pi/2(2n+1)$: Rad $\lvert\theta\rvert\fallingdotseq 100(2n+1)$: Gra
°’ ”	$\lvert a\rvert$, b, c$<1\times10^{100}$ $0\leqq$ b, c	”	”	
$\overleftarrow{°’ ”}$	$\lvert x\rvert<2.777777777\times10^{96}$ Hexadecimal display: $\lvert x\rvert\leqq 2777777.777$			
x^y	$x>0$: $-1\times10^{100}<y\log x<100$ $x=0$: $y>0$ $x<0$: $y=n$, $\frac{1}{2n+1}$ (n is an integer) However; $-1\times10^{100}<\frac{1}{y}\log\lvert x\rvert$ <100	”	”	
$\sqrt[x]{y}$	$y>0$: $x\fallingdotseq 0$ $-1\times10^{100}<\frac{1}{x}\log y<100$ $y=0$: $x>0$ $y<0$: $x=2n+1$, $\frac{1}{n}$ ($n\fallingdotseq 0$, n is an integer) However; $-1\times10^{100}<\frac{1}{x}\log\lvert y\rvert$ <100	”	”	

Function	Input range	Internal digits	Accuracy
$a^b/_c$	•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 digit.
SD (LR)	$\lvert x\rvert<1\times10^{50}$ $\lvert y\rvert<1\times10^{50}$ $\lvert n\rvert<1\times10^{100}$ $x\sigma_n$, $y\sigma_n$, $\bar{x}$, $\bar{y}$, A, B, r: $n\fallingdotseq 0$ $x\sigma_{n-1}$, $y\sigma_{n-1}$: $n\fallingdotseq 0, 1$	”	”

Function	Input range
BASE-N	Values after variable within following range: Dec: $-2147483648\leqq x\leqq 2147483647$ Bin: $10000000000\leqq x$ $\leqq 1111111111$ (negative) $0\leqq x\leqq 0111111111$ (0, positive) Oct: $2000000000\leqq x\leqq 3777777777$ (negative) $0\leqq x\leqq 1777777777$ (0, positive) Hex: $80000000\leqq x\leqq$ FFFFFFFF (negative) $0\leqq x\leqq$ 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.

Specifications

Model: fx-6300G

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, ln, 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, Overdraw, Trace, Zoom (×f, ×$1/f$, 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 functions: 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, π, random numbers, absolute values, integers, fractions.

Statistics: Standard deviation — number of data, sum, sum of squares, mean, 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, estimated value of x, estimated value of y.

Special functions: Insert, delete, replay functions, substitution (=), multistatement (: and ◢).

Memories: 26 standard (maximum 76), Ans memory

Calculation range: $\pm 1 \times 10^{-99} \sim \pm 9.999999999 \times 10^{99}$ 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: Norm 1 — $10^{-2} > |x|$, $|x| \geqq 10^{10}$
Norm 2 — $10^{-9} > |x|$, $|x| \geqq 10^{10}$

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 programs: 10 maximum (P0 to P9)

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

General

Power supply:	Two lithium batteries (CR2032)
Power consumption:	0.009 W
Battery life:	Approximately 350 hours on CR2032
Auto power off:	Power is automatically switched off approximately 6 minutes after last operation.
Ambient temperature range:	0°C ~ 40°C (32°F ~ 104°F)
Dimensions:	9.9mmH × 73mmW × 141.5mmD ($^3/_8$"H × 2$^7/_8$"W × 5$^1/_2$"D)
Weight:	84g (2.9oz) including batteries

Index

Index

MEMO

MEMO