# *fx-4500PA*

## *User's Guide*
## *Guía del usuario*

E S

CASIO

# CASIO *fx-4500PA*


Section 1 Configuration and Operation

Section 2 Manual Calculations

Section 3 Integration Calculation

Section 4 Program Calculation

PROGRAM LIBRARY

- 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.
- The manufacturer assumes no responsibility for claims from third parties for loss or damages arising through the use of this calculator or manual.
- The manufacturer assumes no responsibility for any loss or damages arising from loss of data and/or formulas incurred while using this calculator or manual.

# Introduction

Thank you for purchasing the Casio fx-4500PA.

This unit is an advanced programmable scientific calculator which features a 2-tier display capable of displaying both formulas and results at the same time. The upper display features a 12-character dot matrix display capable of alphabetic and numeric display. The fx-4500PA also features a large-volume program memory, and is capable of performing integration calculations. A built-in formula memory is handy when performing repeat calculations, and manual calculations can be easily performed by following written formulas.

Before using this unit, be sure to read these instructions thoroughly. When you're finished reading these instructions, be sure to keep this manual where you can refer to it often.

This manual is composed of four sections:

1. Configuration and Operation
2. Manual Calculations
3. Integration Calculation
4. Program Calculation

Section 1 should be read first to become familiar with the nomenclature, handling and cautions concerning this unit.

Section 2, 3 and 4 can then be read in order to master these types of computations through samples and explanations.

## Section 1 Configuration and Operation


**Key markings** ..... 8
**Modes** ..... 9
Manual calculation modes ..... 9
**Display** ..... 11
Two-tier display ..... 11
Display symbols ..... 12
Exponential display ..... 12
Special display functions ..... 13
**Handling Precautions** ..... 14
**Power and Battery Replacement** ..... 15
Replacing batteries ..... 15
To replace the memory backup battery ..... 15
To replace the main battery ..... 16
Auto Power-OFF function ..... 17
**Nomenclature and Functions** ..... 18
**Before beginning calculations....** ..... 27
Calculation priority sequence ..... 27
Number of stacks ..... 28
Number of input/output digits and calculation digits ..... 28
Overflow and errors ..... 29
Number of input characters ..... 30
Corrections ..... 30


## Section 2 Manual Calculations


**Basic calculations** ..... 34
Arithmetic operations ..... 34
Parenthesis calculations ..... 35
Percentage calculations ..... 36
Specifying the number of decimal places, the number of significant digits and the exponent display ..... 37
**Memory** ..... 39
(1) Variable memory ..... 39
Array-type memories ..... 41
Deleting memory ..... 43
(2) Independent memories ..... 44
**Special functions** ..... 46
Answer function ..... 46
Omitting the multiplication sign (×) ..... 46
Continuous calculation function ..... 47
Replay function ..... 48
Error position display function ..... 49
Multistatement function ..... 50
**Scientific function calculations** ..... 52
Trigonometric functions and inverse trigonometric functions ..... 52
Logarithmic and exponential functions ..... 53
Performing hyperbolic and inverse hyperbolic functions ..... 54
Coordinate transformation ..... 55
Permutation and combination ..... 56
Other functions ($\sqrt{\ }$, $x^2$, $x^{-1}$, $x!$, $\sqrt[3]{\ }$, Ran#) ..... 57
Fractions ..... 58
**Engineering symbol calculations** ..... 59
**Binary, octal, decimal, hexadecimal calculations** ..... 60
Binary and octal block display ..... 61
Binary, octal, decimal, hexadecimal conversions ..... 63
Negative expressions ..... 64
Basic arithmetic operations using binary, octal, decimal and hexadecimal values ..... 65
Logical operations ..... 66
**Statistical calculations** ..... 67
Standard deviation ..... 67
Regression calculation ..... 70
**Formula memory function** ..... 77
Purpose of the formula memory function ..... 77


## Section 3 Integration Calculation


**Input of function $f(x)$ and integration calculation** ..... 84
Examples of operation ..... 85
Notes on integration calculation ..... 88


## Section 4 Program Calculation


**What is a program?** ..... 90
Programming ..... 90
Calculation formula ..... 91
Registering filenames ..... 91
Writing programs ..... 92
Program execution ..... 93
**Program correction, addition and deletion** ..... 95
Inserting lines ..... 95
Inserting a line at the beginning ..... 95

Editing programs ..... 95
Program deletion ..... 100
**Program searches** ..... **101**
Filename sequential search ..... 101
Sequential search of program lines ..... 103
Direct search ..... 104
Scrolling right and left in filenames and program lines ..... 106
**Program execution** ..... **107**
Execution through filename search ..... 107
Execution by pressing [SHIFT] [Prog] ..... 108
Aborting execution ..... 109
Program debugging (correcting errors) ..... 109
**Convenient program commands** ..... **109**
Jump commands ..... 109
Subroutines ..... 114
Pause command ..... 117
Variable input command ..... 118
Fixm ..... 118
**Remaining program capacity** ..... **120**
Determining the number of remaining steps ..... 120
**Using the unit as a data bank** ..... **121**

## PROGRAM LIBRARY

**1. Prime factor analysis** ..... **126**
**2. Greatest common measure** ..... **128**
**3. Δ↔Y transformation** ..... **130**
**4. Minimum loss matching** ..... **132**
**5. Cantilever under concentrated load** ..... **134**
**6. Normal distribution** ..... **136**
**7. Numerical solution of an equation (Newton's law)** ..... **138**
**8. Quadratic equation** ..... **140**
**9. Complex numbers** ..... **144**
**Error messages at-a-glance** ..... **154**
**Input ranges of functions** ..... **155**
**Specifications** ..... **158**

# Section 1 Configuration and Operation

Key markings
Modes
Display
Handling Precautions
Power and Battery Replacement
Nomenclature and Functions
Before beginning calculations....

# Section 1 Configuration and Operation

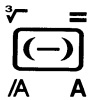
Before using this unit for the first time, be sure to press the ALL RESET button on the back of the unit.

•Flow of Operations (Be sure to read this!)

In contrast to standard electronic calculators, the keys on "scientific" calculators often perform more than one function. The following explains how to use these keys, so it's a good idea to read this section carefully before using your calculator for the first time.

## Key markings

The fx-4500PA uses any given key to perform a number of different functions. For example, the key shown below is used to perform the following 5 functions: ①: (−), ②: $\sqrt[3]{\ }$, ③: =, ④: A, ⑤: /A.




The function of this key differs depending on the operational mode the calculator is set to (see page 9), however if it is pressed directly, it performs the (−) function. If you press the SHIFT key and then press this key, it carries out the second function $\sqrt[3]{\ }$. If you press the 2ndF key and then press the (−) key, it acts as the = key. And if you press it after pressing the ALPHA key, you can input the radical "A". Finally, you can use the (−) key while in the BASE-N (see page 60) "HEX" mode to input the hexadecimal "/A". You'll notice that the key shows marks for each of these functions, which are color coded for easy identification.

Referring back to the (−) key, the function marked in orange is carried out after pressing the SHIFT key. After pressing the 2ndF key, the (−) key operates the function marked in light green. After pressing the ALPHA key, the function marked in red is carried out, and in the BASE-N mode, the (−) key operates the function marked in green.

Next, let's look at the 1 key. Note that the "$\Sigma x^2$" mark is in blue brackets. The fx-4500PA functions marked in these blue brackets are used for standard deviation (SD mode) and regression (LR mode) calculations.

Functions marked in ORANGE — Accessed by first pressing SHIFT key.
Functions marked in LIGHT GREEN — Accessed by first pressing 2ndF key.
Functions marked in RED — Accessed by first pressing ALPHA key.
Functions marked in GREEN — Accessed in BASE-N mode.
Functions marked by BLUE brackets — Accessed in SD mode or LR mode.

## Modes

When using the fx-4500PA, it is necessary to select the proper mode to suit your calculation requirements. This can be done by using the MODE key in combination with the number keys. (Refer to plate below the display window.)

### ■Manual calculation modes

#### • Calculation modes

**MODE 0: COMP mode**
General calculations, including function calculations.

**MODE 1: BASE-N mode**
Binary, octal, decimal, hexadecimal conversion and calculations, as well as logical operations.

**MODE 2: LR mode**
Regression calculation. ("LR" symbol appears in display window when this mode is selected.)

**MODE 3: SD mode**
Standard deviation calculation. ("SD" symbol appears in display window when this mode is selected.)

*Modes 0 ~ 3 are totally independent, and cannot be used together.

**MODE •: Eng mode**
Engineering symbol calculation. ("Eng" symbol appears in display window when this mode is selected.) (Refer to page 59.)

*The calculation mode last selected is retained in memory when the fx-4500PA's power is switched OFF.

#### • Angular measurement modes

**MODE 4: Deg mode**
Specifies measurement in "degrees". (" D " symbol appears in display window when this mode is selected.)

**MODE 5: Rad mode**
Specifies measurement in "radians". (" R " symbol appears in display window when this mode is selected.)

**MODE 6: Gra mode**
Specifies measurement in "grads". (" G " symbol appears in display window when this mode is selected.)

*With the exception of the BASE-N mode, modes 4 ~ 6 can be used in combination with the manual calculation modes.

## • Display modes

**[MODE] [7]: Fix mode**

Specifies number of decimal places. ("Fix" symbol appears in display window when this mode is selected.)

**[MODE] [8]: Sci mode**

Specifies number of significant digits. ("Sci" symbol appears in display window when this mode is selected.)

**[MODE] [9]: Norm mode**

Cancels "Fix" and "Sci" specifications.

This operation also changes the range of the exponent display (see page 12).

*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 fx-4500PA's power is switched OFF.

## • Program calculation mode

**[MODE] [EXP]: WRT mode**

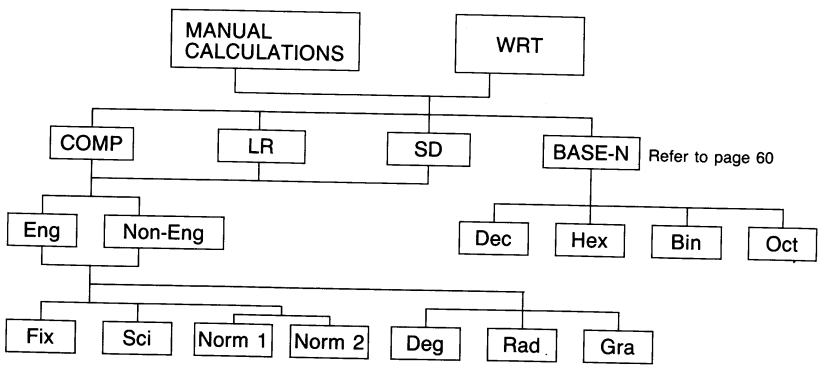
Specified when writing or correcting programs (files). ("WRT" symbol appears in display window when this mode is selected.) Press again to cancel WRT mode.

**[MODE] [Ans]: Defm mode**

Press to expand number of memories. ("Defm" appears in display window when this mode is selected.) After specifying this mode, input a value and press the [EXE] key to specify the number of useable memories (see page 41.)

**Example** [MODE] [Ans] 10 [EXE] — Number of memories expanded by 10.

**Mode Hierarchy Diagram**




## • Abbreviations

| | | | |
|---|---|---|---|
| COMP | Compute | Dec | Decimal number |
| LR | Linear regression | Hex | Hexadecimal number |
| SD | Standard deviation | Bin | Binary number |
| Eng | Engineering | Oct | Octal number |

*To return to standard operation (initialized state) press [MODE] [0] (COMP mode) — [MODE] [4] (Deg mode) — [MODE] [9] (Norm mode)

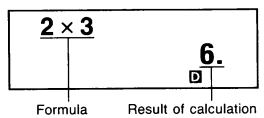
# Display

## ■ Two-tier display

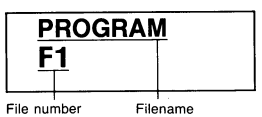
This unit features a two-tier display. The upper tier is a dot display which features an input buffer, for display of up to 12 characters. The lower display is capable of displaying 10 digits for a mantissa, as well as 2 digits for an exponent. When formulas are input, they are displayed on the upper display, and then results are shown on the lower display when the [EXE] is pressed to execute the calculation. This allows both the formula and the result to be displayed simultaneously.

In addition, when filenames are displayed, the file name is shown on the upper display, with the file number shown on the lower display. When programs are displayed, the program data is displayed on the upper display, with the program line number shown on the lower display.

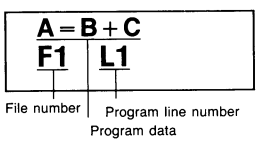
**Example** $2 \times 3 = 6$




**Example** **Filename**

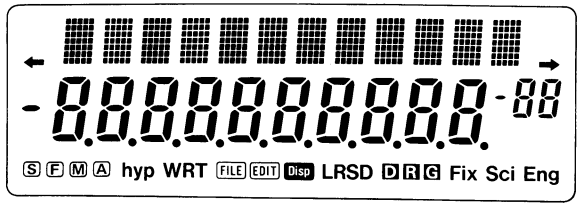



**Example** **Program**



## ■Display symbols

The display window features symbols which light to indicate the present operational status of the fx-4500PA.




S: Indicates SHIFT key has been pressed.
F: Indicates 2ndF key has been pressed.
M: Indicates MODE key has been pressed.
A: Indicates ALPHA key has been pressed.
hyp: Indicates hyp key has been pressed.
WRT: Indicates calculator in the WRT mode.
FILE: Indicates filename or program (file contents) is displayed.
EDIT: Indicates program is being edited in WRT mode.
Disp: Indicates intermediate result is displayed.
LR: Indicates LR mode has been specified.
SD: Indicates SD mode has been specified.
D: Indicates angular measurement in units of "Degrees".
R: Indicates angular measurement in units of "Radians".
G: Indicates angular measurement in units of "Grads".
Fix: Indicates specification of number of decimal places is being executed.
Sci: Indicates specification of number of significant digits is being executed.
Eng: Indicates Eng mode has been specified.
← →: Indicates number of characters exceeds limitation of screen. Non-displayed characters can be viewed by "scrolling" right or left, as indicated by arrow(s).

## ■Exponential display

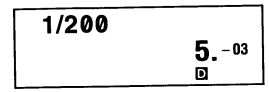
During normal calculation, this unit is capable of displaying up to 10 digits. However if calculation results exceed this limit, they are automatically displayed in exponential format. You can choose from two different types of exponential display formats:

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

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

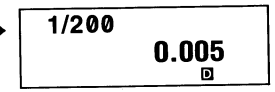
Selection of these modes can be carried out by pressing MODE 9, when no specification has been made for the number of decimal places or significant digits.
The present status is not displayed, so it is necessary to perform the following procedure to specify either display format:

1 ab/c 200 EXE →




(Format A)

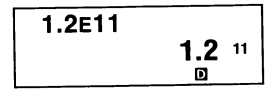
→




(Format B)

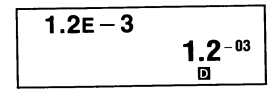
The examples given in this manual shows calculation results in exponential display format "A".

How to view the calculation results in exponential format.




Mantissa Exponent

➡ $1.2 \times 10^{11}$ ➡ 120,000,000,000



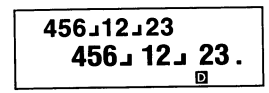

Mantissa Exponent

➡ $1.2 \times 10^{-3}$ ➡ 0.0012

## ■Special display functions

For fractional, hexadecimal and other special calculations, results are shown on the display as follows:

**•Fractions**




[Indicates $456\frac{12}{23}$]

•**Hexadecimal numbers**

ABCDEF12
AbCdEF12 H [Indicates ABCDEF12$_{16}$ (= −1412567278$_{10}$)]

•**Sexagesimal numbers**

12.58244
12°34°56.78
D
[Indicates 12°34'56.78"]

Degrees Minutes Seconds

## Handling Precautions

- This unit is composed of precision electronic components, and should never be disassembled. Do not drop it or otherwise subject it to sudden impacts, or sudden changes in temperature. Be especially careful to avoid storing the unit or leaving it in areas exposed to high temperature, humidity or large amounts of dust. When exposed to low temperatures, the unit will require more time to display answers and may even fail to operate. The display will return to normal once normal temperature is attained.
- The display will appear blank while the unit is performing calculations. At this time most keys will be inoperative. Because of this, keys should normally be used while confirming proper operation by checking the display.
- Batteries should be replaced every 5 years, even if the unit is not used for extended periods. 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 a cloth that has been dipped in a neutral detergent solution and wrung out.
- Note that the manufacturer assumes no responsibility for any loss or claims by third parties which may arise from use of this product.
- Note that the manufacturer assumes no responsibility for any damage incurred as a result of data loss caused by malfunction, repairs or battery replacement. The user should prepare physical records of important data to protect against such data losses.
- If this unit should malfunction, be sure to contact your nearest Casio dealer or service center, explaining the problem in detail.

## Power and Battery Replacement

Power is supplied to this unit by one CR2032 lithium battery, which is used for normal operations, as well as one CR2032 lithium battery used for memory backup.
If both of these batteries are removed at the same time, programs and data will be lost, so avoid replacing them at the same time.

***If both batteries have been removed from the unit for some reason, replace them and press the "All Reset" button after turning the power ON.**

### ■Replacing batteries

If batteries become weak, the "Low Battery" indicator appears on the display:

_Low battery

If you continue to use the unit after this display appears, power will turn OFF automatically and operation will become impossible even if you press the AC ON key. If this occurs, replace the CR2032 battery as soon as possible. Do not leave dead batteries in the unit. Doing so may result in damage or lost memory contents.

### ■To replace the memory backup battery

The battery used for memory backup should be replaced every two years.

①Press the OFF key. Then remove the two screws on the back of the unit and remove the back cover.

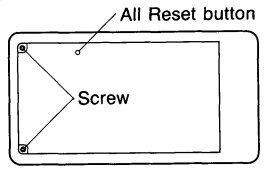
②Remove the screw holding the battery pressure plate (screw "B") and then remove the battery pressure plate.

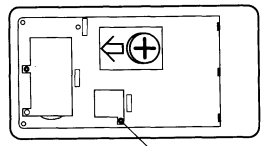
③Remove the old battery from the unit.
(This can be done easily by turning the unit so that the battery compartment is facing downwards, and then lightly tapping the unit.)

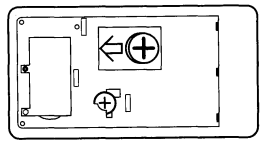
④Wipe the surfaces of the new battery with a soft, dry cloth and load them into the unit, making sure that the positive ⊕ side is facing upwards.

⑤Fasten the battery pressure plate in place using the screw.

⑥Replace the back cover and press the ON key. Memory contents are protected by the



## ■To replace the main battery

①Press the OFF key. Then remove the two screws on the back of the unit and remove the back cover.

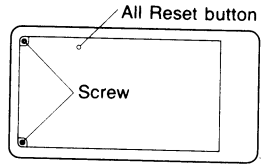
②Remove the screw holding the battery pressure plate (screw "A") and then remove the battery pressure plate.

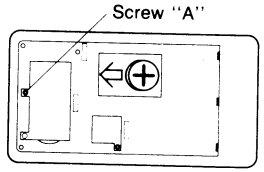
③Remove the old main battery from the unit. (This can be done easily by turning the unit so that the battery compartment is facing downwards, and then lightly tapping the unit.)

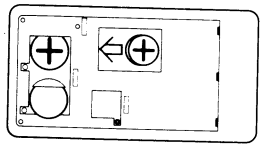
④Wipe the surfaces of the new battery with a soft, dry cloth and load them into the unit, making sure that the positive ⊕ side is facing upwards.

⑤Fasten the battery pressure plate in place using the screw. Then press the ON key.

⑥Replace the back cover. Memory contents are protected by the memory backup battery in this case.



**Avoid changing both batteries at the same time as doing so may result in memory contents being lost.**

**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 the battery are facing in the proper direction.

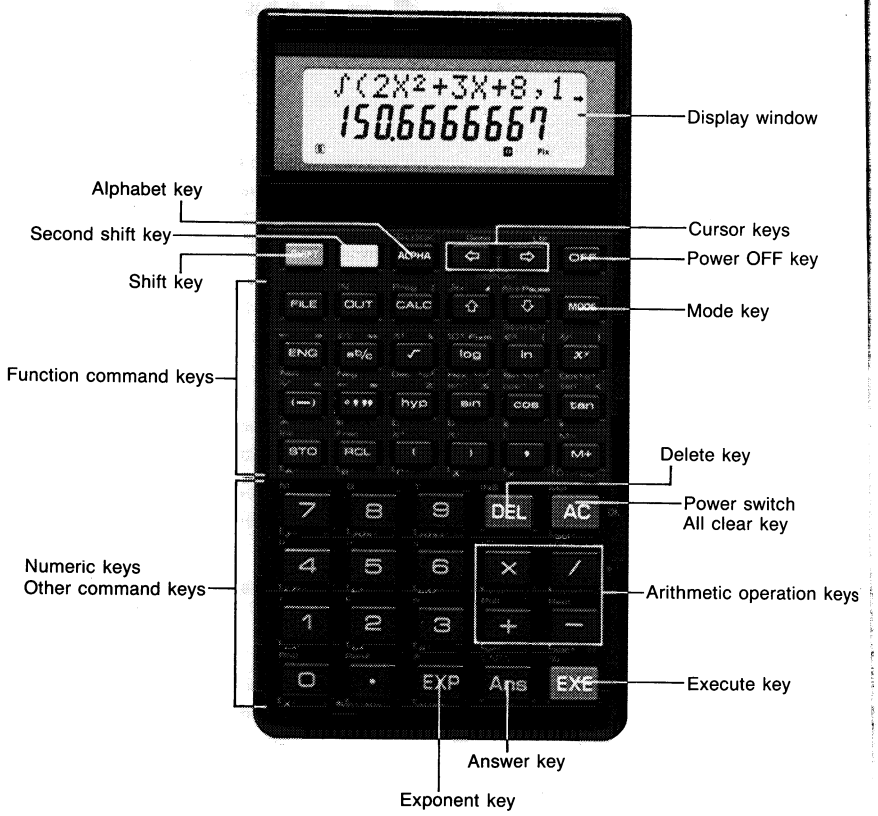
- Never leave a dead battery in the battery compartment.
- Remove the battery if you do not plan to use the unit for long periods.
- Replace the battery at least once every 5 years, no matter how much the unit is used during that period.
- Never try to recharge the battery supplied with the unit.
- Do 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.

## ■Auto Power-OFF function

To preserve battery life, this unit will turn OFF automatically if it is not used for approximately 6 minutes. To restore power, press the AC key. Note that memory contents are protected even when power goes OFF.

## Nomenclature and Functions




### SHIFT Shift key

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

### 2ndF Second shift key

Press when using the function commands and functions marked in light green on the key panel. An "F" will appear on the display to indicate that 2ndF has been pressed. Pressing 2ndF again will cause the "F" to disappear from the display and the unit to return

### MODE Mode key

Press when setting the status of the unit or when specifying the unit of angular measurement. Refer to page 9 for details on modes.

### ALPHA Alphabet key

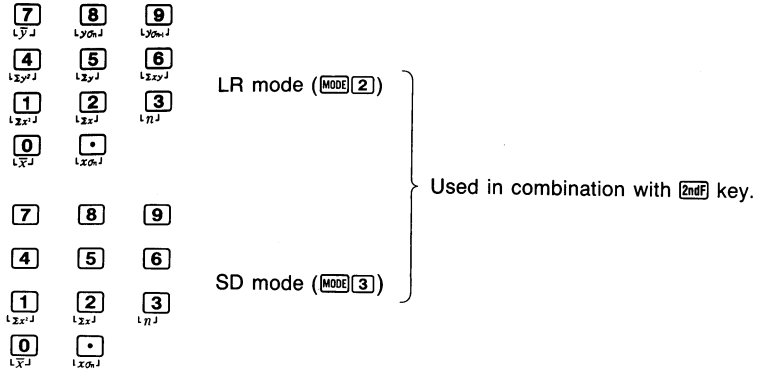
Press to input alphabetic characters or special characters. Pressing ALPHA displays "A" and allows the input of only one character. After the character is input, the unit returns to the status it was in before the ALPHA key was originally pressed. Pressing SHIFT followed by ALPHA will lock the unit in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

| | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| " | | | | [ | ] |
| 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 | SPACE | | |

### 0 ~ 9 , • Numeric keys

- When numeric values or calculation commands are input, they appear on the display window from the left. The • key is used to indicate the decimal point.
- Depending on the selected mode, the SHIFT and 2ndF keys can be used in combination with the number keys to specify the following functions:

| | | | | |
|---|---|---|---|---|
| M 7 | G 8 | T 9 | | |
| μ 4 | m 5 | k 6 | COMP mode (MODE 0) | Used in combination with SHIFT key. |
| f 1 | p 2 | n 3 | LR mode (MODE 2) | |
| Rnd 0 | Ran# • | | SD mode (MODE 3) | |




•When the SHIFT key is pressed in the COMP mode, LR mode or SD mode, the following functions are specified:

SHIFT Rnd ..... Internal rounding
This key operation rounds the internal value (value stored in the Y register) to 10 digits. Note that this also rounds the result that is produced by the Ans function. In the FIX and SCI modes, this key operation changes the internal value to the form specified for value display.

SHIFT Ran# ..... Random number generation
Generates random numbers between 0.000 and 0.999.

For information on other functions, see "Engineering symbol calculations" on page 59.

•The following functions can be specified by pressing 2ndF in the LR mode or SD mode (only some functions are available).

2ndF 0 ($\bar{x}$) ..... $\bar{x}$ (Mean of $x$)
2ndF 1 ($\Sigma x^2$) ..... $\Sigma x^2$ (Sum of squares of $x$)
2ndF 2 ($\Sigma x$) ..... $\Sigma x$ (Sum of $x$)
2ndF 3 ($n$) ..... $n$ (Number of data)
2ndF 4 ($\Sigma y^2$) ..... $\Sigma y^2$ (Sum of squares of $y$)
2ndF 5 ($\Sigma y$) ..... $\Sigma y$ (Sum of $y$)
2ndF 6 ($\Sigma xy$) ..... $\Sigma xy$ (Sum of products of $x$ and $y$)
2ndF 7 ($\bar{y}$) ..... $\bar{y}$ (Mean of $y$)
2ndF 8 ($y\sigma_n$) ..... $y\sigma_n$ (Standard deviation of $y$)
2ndF 9 ($y\sigma_{n-1}$) ..... $y\sigma_{n-1}$ (Sample standard deviation of $y$)
2ndF - ($x\sigma_n$) ..... $x\sigma_n$ (Standard deviation of $x$)

These functions are used in standard deviation and regression calculations. For details, see sections on "Standard deviation" (page 67) and "Regression calculation" (page 70).

### π EXP ($x\sigma_{n-1}$ Z) Exponent/Pi/Standard deviation calculation key

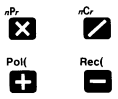
•When using exponents, the EXP key is pressed after the mantissa is input. For example, to input $2.56 \times 10^{34}$, press 2.56 EXP 34. When inputting exponents into a program, the EXP key is pressed after the value is input.
•When pressed following the SHIFT key, the value of pi ($\pi$) is input.
•When pressed following 2ndF key in LR mode or SD mode, the sample standard deviation of $x$ is calculated.

2ndF EXP ($x\sigma_{n-1}$) ..... $x\sigma_{n-1}$ (Sample standard deviation of $x$) is calculated.

### Pol( + (or V) Rec( − (xor W) nPr × (and Q) nCr ÷ (xnor R) Arithmetic operation/Coordinate transformation/Permutation and combination/Logical operation keys

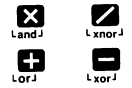
•When carrying out addition, subtraction multiplication and division, enter the calculation as it is written, from left to right.
•+ and − keys can be used to indicate signs.
•SHIFT key combinations for the various modes are as follows:

COMP mode, SD mode or LR mode




SHIFT Pol( + ..... Coordinate transformation; press to transform polar coordinate into rectangular coordinate.
SHIFT Rec( − ..... Coordinate transformation; press to transform rectangular coordinate into polar coordinate.
SHIFT nPr × ..... Permutation; press when making permutation calculations.
SHIFT nCr ÷ ..... Combination; press when making combination calculations.

BASE-N mode




SHIFT + (or) ..... or; press when calculating "or" of logical operation.
SHIFT − (xor) ..... xor; press when calculating "xor" of logical operation.
SHIFT × (and) ..... and; press when calculating "and" of logical operation.
SHIFT ÷ (xnor) ..... xnor; press when calculating "xnor" of logical operation.

*For division, the "/" (slash) key is used.

### % EXE Execute/Percent key

•Press to obtain the result of a calculation. Pressed after data input for a program calculation or to advance to the next execution after a result is obtained.
•Press following SHIFT key for percentage calculations. Note that percentage calculations

### CAPA Ans SPACE Answer/CAPA/Space key

- Press Ans followed by EXE to recall the last calculation result.
- Hold down following SHIFT key to display number of remaining steps in program.
- Press following ALPHA key to input a space.

### INS DEL Delete/Insert key

- Press DEL to delete character where cursor is flashing. Deletes character to left of cursor when cursor is to the right of the last input character.
- Press following SHIFT key to display the insert cursor ( ▯ ). Entering a value while the insert cursor is displayed inserts the value in the position immediately preceding the insert cursor location.

### Mcl AC ON LScl All clear/Memory clear/Statistical data clear/Power On key

- Press to clear all input characters or formulas. Also, press to clear Error Check message on display.
- Pressing SHIFT followed by AC clears all data in unit's memory.
- Pressing 2ndF followed by AC clears contents of statistical calculation memory.
- Press when power is OFF to turn power ON.

### Goto ⇦ ⇨ Lbl REPLAY Cursor/Replay/Jump command keys

- Press to move the cursor to the left or right on the display to correct formulas or numeric values.
  Pressing ⇦ moves the cursor to the left, while pressing ⇨ moves it to the right. Pressing either key and holding it down will cause continuous movement of the cursor in the respective direction.
- Once a formula or numeric value is input and EXE is pressed, these keys become replay keys. Pressing ⇦ displays the formula or numeric value from the end, and pressing ⇨ displays it from the beginning. Pressing EXE again will re-execute (see page 48).
- These keys are also used to input "Jump" commands which alter program execution. Pressing 2ndF followed by ⇦ enters the "Goto" command. Pressing 2ndF followed by ⇨ enters the "Lbl" (Label) command. For details, refer to page 109.

### OFF Power OFF key

Press to turn the unit's power OFF. Note that mode setting and memory contents are protected even when power is turned OFF.

### FILE File key

Press to recall registered files. For details, refer to page 96.

### IN OUT Formula memory key

Used when making calculations using registered formulas. For details, refer to page 77.

### Prog : CALC Formula memory/Program/Multistatement key

- Press CALC to execute formulas in formula memory. For details, refer to page 77.
- Press SHIFT followed by CALC and EXE to execute programs. For details, refer to page 108.
- Press 2ndF followed by CALC to separate formulas or commands in programmed calculations or consecutive calculations. The result of such combinations is known as a "multistatement". For details, refer to page 50.

### ∫dx ⇧ File line scroll up/Integration/Display key

- Press while file contents are displayed to scroll up to the previous file line.
- Press SHIFT followed by ⇧ when making integrations. For details, refer to page 84.
- Press 2ndF followed by ⇧ to display results of program calculations and consecutive calculations.

### Abs Pause ⇩ SEARCH File line scroll down/Absolute value/Pause command/Search key

- Press while file contents are displayed to scroll down to the next file line.
- Press SHIFT followed by ⇩ when making absolute value calculations.
- Press 2ndF followed by ⇩ to input "Pause" command. For details, refer to page 117.
- Press to search contents of file using "Search" function. For details, refer to page 105.

### ⇐ ⇒ ENG Not ,, Engineering/Judge command/Not key

- Each press of this key shifts the decimal of the displayed value three decimal places to the right or left. This in effect results in conversion of the value from one metric unit to another, such as $10^{-3}$ milliseconds, $10^{-6}$ microseconds, $10^{-12}$ picoseconds, or $10^{3}$ kilohertz, $10^{6}$ megahertz, $10^{9}$ gigahertz.

**Example**

| | | |
|---|---|---|
| 12.3456 EXE | 12.3456 | |
| 1st time ENG is entered | 12.3456 | 00 |
| 2nd time ENG is entered | 12345.6 | -03 |
| 3rd time ENG is entered | 12345600. | -06 |
| 4th time ENG is entered | 12345600. | -06 (No change) |

| | | |
|---|---|---|
| 12.3456 EXE | 12.3456 | |
| 1st time SHIFT ENG is entered | 0.0123456 | 03 |
| 2nd time SHIFT ENG is entered | 0.000012345 | 06 |
| 3rd time SHIFT ENG is entered | 0.000000012 | 09 |
| 4th time SHIFT ENG is entered | 0.000000012 | 09 (No change) |

- When pressed after pressing 2ndF key, "judge" symbol "⇒", used in executing Jump command, is input. For details, refer to page 112.
- When pressed in BASE-N mode, this key executes "Not" function used in logical operation.

### [a b/c] Fractions/Judge command/Negative key

Neg

- Used when inputting fractions and mixed numbers.

**Example** To input $\frac{23}{45}$, press 23 [SHIFT] [a b/c] 45: To input $2\frac{3}{4}$, press 2 [a b/c] 3 [a b/c] 4:

- By pressing [SHIFT] [a b/c] in succession, the displayed value will be converted to the improper fraction.
- When pressed after pressing [2ndF] key, "judge" symbol "≠", used in executing Jump command, is input. For details, refer to page 112.
- 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.

### [√] Square root/Square/Judge command/Decimal value key

Dec ⌊d⌋

- Press prior to entering a numeric value to obtain the square root of that value.
- Enter a value, and press this key following [SHIFT] to obtain the square of the value.
- When pressed after pressing [2ndF] key, the "judge" symbol "◺", used in executing Jump command, is input. For details, refer to page 112.
- Press in the BASE-N mode to specify the decimal calculation mode.
- When pressed following [SHIFT] in the BASE-N mode, the subsequently entered value is specified as a decimal value.

### [log] Common logarithm/Exponent of 10/Variable fix command/Hexadecimal key

$10^x$ Fixm Hex ⌊h⌋

- 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.
- When pressed following the [2ndF] key, "Fixm" is entered. For details, refer to page 118.
- Press in the BASE-N mode to specify the hexadecimal calculation mode.
- When pressed following [SHIFT] in the BASE-N mode, the subsequently entered value is specified as a hexadecimal value.

### [ln] Natural logarithm/Exponential/Variable input command/Binary key

$e^x$ { Bin ⌊b⌋ {

- 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 "$e$".
- When pressed following the [2ndF] key, the "{" symbol used in executing the variable input command is input. For details, refer to page 118.
- Press 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.

### [$x^y$] Power/Root/Variable input command/Octal key

Oct ⌊o⌋ }

- Enter $x$, press this key and then enter $y$ (any number) to calculate $x$ to the power of $y$.
- To calculate the $x$th root of $y$, press after pressing [SHIFT].
- When pressed following the [2ndF] key, the "}" symbol used in executing the variable input command is input. For details, refer to page 118.
- Press in the BASE-N mode to specify the octal calculation mode.
- When pressed following the [SHIFT] key in the BASE-N mode, the subsequently entered value is specified as an octal value.

### [(−)] Negative/Cube root/Equal key

/A A

- Press prior to entering a numeric value to make that value negative.

**Example** −123 ➡ [(−)] 123

- Press following the [SHIFT] key to obtain the cube root of a subsequently entered numeric value.
- Press following the [2ndF] key to enter the "=" sign.

### [°'''] Decimal ↔ Sexagesimal/Not equal key

⇐ ≠ B B

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

**Example** 78°45'12'' ➡ 78 [°'''] 45 [°'''] 12 [°''']

- When pressed following the [SHIFT] key, a decimal based value can be displayed in degrees/minutes/seconds.
- Press following the [2ndF] key to input the "≠" sign.

### [hyp] Hyperbolic/Relational operator 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 following the [2ndF] key to enter the "≥" sign.

### [sin] [cos] [tan] Trigonometric function/Inverse trigonometric function/Relational operator keys

$\sin^{-1}$ ≤ D D; $\cos^{-1}$ > E E; $\tan^{-1}$ < 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 [2ndF] and then one of these keys to input the "≦", ">" and "<" signs, respectively.

### Int STO LAJ Store memory/Integer/Constant term key

- Press prior to inputting alphabet character when inputting calculation results to memory.
- Press SHIFT key followed by STO key prior to inputting number in order to obtain integer part of that number.
- Press following 2ndF key in LR mode to calculate constant term "A" in regression formula.

  2ndF STO (A) ..... Calculation of A (Constant term of regression formula)

### Frac RCL LBJ Recall memory/Fraction/Regression coefficient key

- Press prior to inputting alphabet characters to display value input into memory.
- Press SHIFT key followed by RCL key prior to inputting number in order to obtain fraction part of that number.
- Press following 2ndF key in LR mode to calculate regression coefficient "B" in regression formula.

  2ndF RCL (B) ..... Calculation of B (Regression coefficient of regression formula)

### $x^{-1}$ ( LFJ G $x!$ ) LSJ H Parentheses/Reciprocal/Factorial/Correlation coefficient/Estimated value of $x$ key

- Press the open parenthesis key and the closed parenthesis key at the position required in a formula.
- Press SHIFT followed by ( prior to entering a value to obtain the reciprocal of that value.
- Press SHIFT followed by ) after entering a value to obtain the factorial of that value.
- In the LR mode, press following the 2ndF key for coefficient of correlation calculation and estimated value of $x$ in linear regression calculation, respectively.

  2ndF ( r ) ..... Calculation of r (correlation coefficient)

  2ndF ( $\hat{x}$ ) ..... Calculation of $\hat{x}$ (estimate of value of $x$)

### ; , LSJ I Comma/Semicolon/Estimated value of $y$ key

- Press to enter comma in statistical and other formulas.
- Press following SHIFT to enter semicolon.
- In the LR mode, press following the 2ndF key for the estimated value of $y$ in regression calculation.

### M− M+ LDTJ LCLJ Memory plus/Memory minus/Data input/Clear key

- Press to add displayed value to memory. Note that when formula is displayed, results are first derived and then stored into memory.
- Press following SHIFT key to subtract displayed value from memory.
- In the LR and SD modes, press to input data.
- In the LR and SD modes, press following the SHIFT key to clear data which has been input incorrectly.

# Before beginning calculations....

## Calculation priority sequence

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

① Coordinate transformation/integration  Pol $(x, y)$, Rec $(r, \theta)$, $\int dx$

② Type A functions

These functions are those in which the value is entered and then the function key is pressed.

$x^2$, $x^{-1}$, $x!$, °' '', Eng symbols

③ Power/root  $x^y$, $\sqrt[x]{\ }$

④ Fractions  $a\,b/c$

⑤ Abbreviated multiplication format in front of $\pi$, memory or parenthesis $2\pi$, 5A, $\pi$R, etc.

⑥ Type B functions

These functions are those in which 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, $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$, Int, Frac, Abs, (−), (following in BASE-N mode only) d, H, b, o, Neg, Not

⑦ Abbreviated multiplication format in front of Type B functions  $2\sqrt{3}$, A log2, etc.

⑧ Permutation, combination  $n\mathrm{P}r$, $n\mathrm{C}r$

⑨ ×, ÷

⑩ +, −

⑪ and

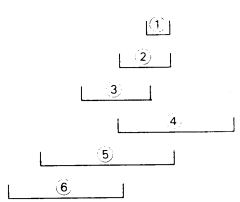
⑫ or, xor, xnor ] In BASE-N mode only.

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

Otherwise, execution is from left to right.

*Everything contained within parentheses receives highest priority.

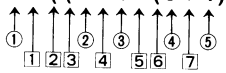
**Example** **$2+3\times(\log \sin 2\pi^2 + 6.8) = 22.07101691$ (in the "Rad" mode)**



## ■Number of stacks

This unit features a memory known as a "stack" for the temporary storage of low priority numeric values and commands (functions, etc.). The numeric value stack has nine levels, while the command stack has 24. If a complex formula is employed that exceeds the stack space available, a stack error (Stk ERROR) message will appear on the display.

**Example** $2 \times ((3 + 4 \times (5 + 4) \div 3) \div 5) + 8 =$




**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 the order of the highest calculation priority first. Once a calculation is executed, it is cleared from the stack.

## ■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 performed internally 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

```
3E5/7
42857.14286
D
```

3 EXP 5 ÷ 7 − 42857 EXE

```
3E5/7-42857
0.1428571
D
```

Once a calculation is completed, the mantissa is rounded off to 10 digits and displayed.

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

3 EXP 5 ÷ 7 EXE

```
3E5/7
42857.14286
D
```

− 42857 EXE

```
42857.14286-
0.14286
D
```

## ■Overflow and errors

If the operational range of the unit is exceeded, or incorrect inputs are made, an error message will appear on the display and subsequent operation will be impossible. This is carried out by 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 function calculations that exceed the input range. (See page 155.)
(3) Improper operation during statistical calculations. (Ex. Attempting to obtain $\bar{x}$ or $x\sigma n$ without data input.)
(4) Illegal argument.
(Ex. Negative value specified for Defm)
(5) The capacity of the numeric value stack or the command stack is exceeded.
(Ex. Entering 23 successive ( 's followed by 2 + 3 × 4)
(6) Input errors are made. (Ex. 5 × × 3 EXE)
(7) Even though memory has not been expanded, a memory such as Z [2] is used. (See page 42 for details on memory.)
(8) When Prog command (see page 114) causes subroutine nesting overflow.
(9) When no Lbl corresponds to Goto command (see page 111), or when no filename corresponds to Prog command (see page 114).

When error messages appear, most keys will become inoperative. In this case, press the AC key to return to normal operation. You can also press the ⇦ key or ⇨ key, causing the cursor to show the position of the error (see "Error position display function" on page 49).

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

(1) ~ (3) Ma ERROR
(4) Arg ERROR
(5) Stk ERROR
(6) Syn ERROR
(7) Mem ERROR
(8) Ne ERROR
(9) Go ERROR

Ne ERROR and Go ERROR messages mainly occur when using programs. (Refer to the Error Message Table on page 154.)

## ■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][√] ([√] key) 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 from the left. Calculation results, however, are displayed from the right.

## ■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_ |
| [←] | 12**2** |
| [3] | 123_ |

**Example** **To change an input of cos60 to sin60:**

| | |
|---|---|
| [cos][6][0] | cos 60_ |
| [←][←][←] | **c**os 60 |
| [sin] | sin **6**0 |

*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. Each press of [SHIFT] [INS] will create a space for input of one command.

**Example** **To correct an input of $2.36^2$ to $\sin 2.36^2$:**

| | |
|---|---|
| [2][•][3][6][SHIFT][$x^2$] | $2.36^2$_ |
| [←][←][←][←][←] | **2**.36$^2$ |
| [SHIFT][INS] | [ ]2.36$^2$ |
| [sin] | sin [ ]2.36$^2$ |

*When [SHIFT][INS] are pressed, the space that is opened is displayed as "[ ]". The function or value assigned to the next key you press will be inserted in the [ ]. To exit from the insertion mode, move the cursors, press [SHIFT][INS], or press [EXE].

Even after the [EXE] key has been pressed to calculate a result, it is possible to use this procedure for correction. Press the [←] key to move the cursor to the place where the correction is to be made.

# Section 2 Manual Calculations

Basic calculations

Memory

Special functions

Scientific function calculations

Engineering symbol calculations

Binary, octal, decimal, hexadecimal calculations

Statistical calculations

Formula memory function

# Section 2 Manual Calculations

## Basic calculations

### Arithmetic operations

- Arithmetic operations are performed by pressing the keys in the same order as noted in the formula.
- For negative values, press (-) before entering the value.

| Example | Operation | Display (Lower) |
|---|---|---|
| $23+4.5-53=-25.5$ | 23 + 4.5 − 53 EXE | −25.5 |
| $56\times(-12)\div(-2.5)=268.8$ | 56 × (-) 12 ÷ (-) 2.5 EXE | 268.8 |
| $12369\times7532\times74103=6.903680613\times10^{12}$ (6903680613000) | 12369 × 7532 × 74103 EXE | 6.903680613 12 |
| $(4.5\times10^{75})\times(-2.3\times10^{-79})=-1.035\times10^{-3}$ (−0.001035) | 4.5 EXP 75 × (-) 2.3 EXP (-) 79 EXE | −1.035 -03 |
| $(2+3)\times10^{2}=500$ | ( 2 + 3 ) × 1 EXP 2 EXE | 500. |
| *The correct answer 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 in 12 digits for a mantissa, and the result is displayed rounded off to 10 digits. Internally, however, the mantissa is calculated to 12 digits. | | |

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

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

### Parenthesis calculations

| Example | Operation | Display (Lower) |
|---|---|---|
| $100-(2+3)\times4=80$ | 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 | 4.5 18 |
| $\frac{6}{4\times5}=0.3$ | 6 ÷ ( 4 × 5 ) EXE | 0.3 |
| *The above is the same as 6 ÷ 4 ÷ 5 EXE. | | |

## ■Percentage calculations

•Premium and discount calculations cannot be performed in the Eng Mode. To perform a premium or discount calculation, first press MODE • to exit the Eng Mode. See page 59 for details.

| Example | Operation | Display (Lower) |
|---|---|---|
| •Percentage<br>26% of $15.00 | 15 × 26 SHIFT % | 3.9 |
| •Premium<br>15% increase from $36.20 | 36.2 × 15 SHIFT % + | 41.63 |
| •Discount<br>4% discount from $47.50 | 47.5 × 4 SHIFT % − | 45.6 |
| •Ratio<br>75 is what % of 250? | 75 ÷ 250 SHIFT % | 30. (%) |
| •Rate of change<br>141 is an increase of what % from 120? | 141 − 120 SHIFT % | 17.5 (%) |
| 240 is a decrease of what % from 300? | 240 − 300 SHIFT % | −20. (%) |

## ■Specifying the number of decimal places, the number of significant digits and the exponent display

•To specify the number of decimal places (Fix), press MODE followed by 7, and then a value indicating the number of places (0 ~ 9). (The "Fix" indicator will appear on the display.)

•To specify the number of significant digits (Sci), press MODE followed by 8, and then a value indicating the number of significant digits (0 ~ 9 to set from 1 to 10 digits with "0" indicating 10 digits). (The "Sci" indicator will appear on the display.)

•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. (Specified values are not cancelled even if power is switched OFF or another mode (besides MODE 9) is specified.)

•MODE 9 cancels Fix and Sci specifications, however the range of the exponent display can be set.

Each time MODE 9 is input, operation switches between Norm1 and Norm2.

Norm 1 : All values less than $10^{-2}$ or greater than $10^{9}$ are automatically expressed as exponents.

Norm 2 : All values less than $10^{-9}$ or greater than $10^{9}$ are automatically expressed as exponents.

•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 to the specified number of decimal places and significant digits, press SHIFT followed by Rnd.

•This operation is invalid in the BASE-N mode. To make this specification in the BASE-N mode, first press MODE followed by 0.

| Example | Operation | Display (Lower) |
|---|---|---|
| 100 ÷ 6 = 16.66666666... | 100 ÷ 6 EXE | 16.66666667 |
| | (Four decimal places specified.) MODE 7 4 | 16.6667 Fix |
| | (Specification cancelled.) MODE 9 | 16.66666667 |
| | (Five significant digits specified.) MODE 8 5 | 1.6667 01 Sci |
| | (Specification cancelled.) MODE 9 | 16.66666667 |
| *Values are displayed rounded off to the place specified. | | |
| 1 ÷ 1000 = 0.001 | (With Norm1 specified.) 1 ÷ 1000 EXE | 1. −03 |
| = 1 × $10^{-3}$ | (Norm2 specified) MODE 9 | 0.001 |

| | | |
|---|---|---|
| $200 \div 7 \times 14 = 400$ | 200 ÷ 7 × 14 EXE | 400. |
| | (Three decimal places specified.) MODE 7 3 | 400.000 Fix |
| | (Calculation continues with 10 digits.) 200 ÷ 7 EXE | 28.571 Fix |
| | × | (Upper) ← 8.57142857 × _ Fix |
| | 14 EXE | 400.000 Fix |
| | If the same calculation is performed with the specified number of digits: | |
| | 200 ÷ 7 EXE | 28.571 Fix |
| | (Value stored internally cut off at specified decimal place.) SHIFT Rnd | 28.571 Fix |
| | × | (Upper) 28.571 × _ Fix |
| | 14 EXE | 399.994 Fix |
| | (Specification cancelled.) MODE 9 | 399.994 |
| $123m \times 456 = 56088m$ | 123 × 456 EXE | 56088. |
| $= 56.088km$ | ENG | 56.088 03 |
| $78g \times 0.96 = 74.88g$ | 78 × 0.96 EXE | 74.88 |
| $= 0.07488kg$ | SHIFT ENG | 0.07488 03 |

# Memory

This unit contains 26 standard memories. There are two basic types of memories, the <Variable> memories, which are accessed by using the STO and RCL keys in combination with the 26 letters of the alphabet, and the <Independent> memories, which are accessed by using the M+, SHIFT M−, RCL and M keys. The variable memory and independent memory utilize the same memory area.
Contents of both of the variable and independent memories are protected even when the power is turned OFF.

## (1) Variable memories

Up to 26 values can be retained in memory at the same time, and can be recalled when desired.

**Example** **Inputting 123 into memory "A":**

| | |
|---|---|
| AC 123 | 123_ |
| STO A | A= 123. |
| AC | _ |
| RCL A | A= 123. |

When formulas are input, the result of the formula's calculation is retained in memory.

**Example** **Inputting the result of 123 × 456 into memory "B":**

| | |
|---|---|
| AC 123 × 456 | 123 × 456_ |
| STO B | B= 56088 |
| AC | _ |

RCL B

B=
56088

If a variable expression is entered, the expression is first calculated according to the values stored in the variable memories used in the expression. The result is then stored in the variable memory specified for the result.

**Example** **Inputting the results of A × B into memory "C":**

AC ALPHA A × ALPHA B

A × B_

STO C

C=
6898824.

AC

_

RCL C

C=
6898824.

*Syn ERROR is generated when an attempt is made to input a substitution formula (such as C = A × B) or multistatements (such as A × B : C × D), and the existing memory contents are retained.

When input is made in a format such as "S = log2", where the variable is equal to the formula, the results of the calculation are input into the specified memory.

**Example** **Executing "S = log2":**

AC ALPHA S 2ndF = log 2

S = log 2_

EXE

S = log 2
0.301029995

AC

_

RCL S

S=
0.301029995

*In the SD mode, variable memories S,T and U are used as statistical memories. In the LR mode, variable memories N, O, P, S, T, and U are used as statistical memories. In addition, G, H, I, J, K, L, and X can be used as integral memories. These variable memories cannot be used simultaneously while making statistical or integral calculations.

## ■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 input by ALPHA (, ALPHA ).

| Standard memory | Array memory |
|---|---|
| A | A[0] |
| B | A[1] |
| C | A[2] |
| D | A[3] |

Proper use of the subscripts shortens programs and makes them easier to use.

## ●Memory expansion

Although there are 26 standard memories (A ~ Z), they can be expanded by changing program storage steps to memory. Memory expansion is performed by converting the 8 steps to one memory.

*See page 90 for information on the number of program steps.

| Number of memories | 26 | 27 | 28 | ..... | 36 | ..... | 100 | ..... | 163 |
|---|---|---|---|---|---|---|---|---|---|
| Number of steps | 1103 | 1095 | 1087 | ..... | 1023 | ..... | 511 | ..... | 7 |

Memory is expanded in units of one. A maximum of 137 memories can be added for a maximum total of 163. Expansion is performed by pressing MODE followed by Ans, 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 Ans 30

Defm 30_

EXE

Defm 30
30.

To check the current number of expanded memories, press [MODE] followed by [Ans] and [EXE].

[MODE][Ans][EXE]

```
Defm
                30.
```

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][Ans]0[EXE]

```
Defm 0
                0.
```

*Though a maximum of 137 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 (Mem ERROR) will be generated and expansion will be impossible.
*The expansion procedure ([MODE][Ans] 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 variable Z[$n$] through variable A[$n+25$], etc., as shown below:

$Z[1]=Y[2]=X[3]=\ \ldots\ =A[26]$ (Defm 1)
$Z[2]=Y[3]=X[4]=\ \ldots\ =A[27]$ (Defm 2)
$\vdots\quad\vdots\quad\vdots\qquad\vdots\qquad\vdots$
$Z[n]=Y[n+1]=X[n+2]=\ \ldots\ =A[n+25]$ (Defm $n$)

*$n$ is the number of expanded memories.

For example, when two memories are added:

$Z[1]=Y[2]=X[3]=\ \ldots\ =A[26]$
$Z[2]=Y[3]=X[4]=\ \ldots\ =A[27]$

These memories are used in the same way as array-type memories, with a subscript being appended to the name.

**Example** **Inputting 123 in Z[2]:**

[MODE][Ans]2[EXE]

```
Defm 2
                2.
```

[ALPHA][Z][ALPHA][(][2][ALPHA][)][2ndF]
[=]123

```
Z[2]=123_
```

[EXE]

```
Z[2]=123
              123.
```

Recall memory data.

[AC]

```
_
```

[ALPHA][Z][ALPHA][(][2][ALPHA][)]

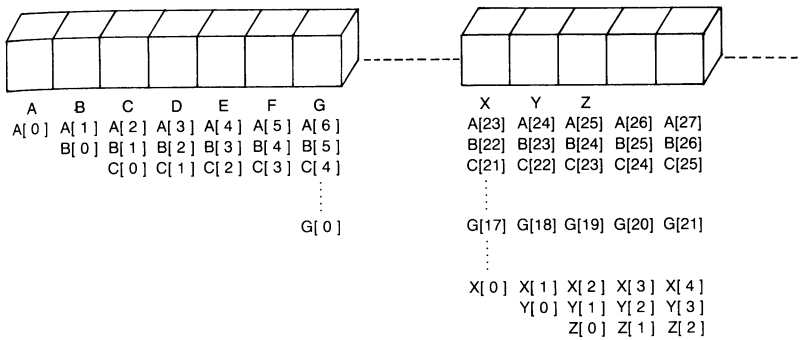
```
Z[2]_
```

[EXE]

```
Z[2]
              123.
```

## • 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 following shows a case in which array-type memories overlap with standard format memories. This situation should always be avoided.
The relation is as follows:




## ■ Deleting memories

To delete all contents of variable memories (including expanded memories), press [SHIFT] followed by [Mcl][EXE].

## (2) Independent memories

Addition and subtraction (to and from sum) results can be stored directly in memory. Results can also be totalized in memory, making it easy to calculate sums.

**Example** **Inputting 123 to independent memory:**

| | |
|---|---|
| AC 123 | 123_ |
| M+ | 123<br>123. |
| Recall memory data. | |
| AC | – |
| RCL M | M =<br>123. |
| Add 25, subtract 12 | |
| 25 M+ 12 SHIFT M− | 12<br>12. |
| (Pressing 25 − 12 M+ provides same result.) | |
| Recall memory data. | |
| AC | – |
| RCL M | M =<br>136. |

*To clear memory contents, press 0 STO M.

*Addition/subtraction to or from sum in memory cannot be carried out with M+, SHIFT and M− keys in SD mode and LR mode.

**•Difference between STO M and M+, SHIFT M−.**

Both STO M and M+, SHIFT M− can be used to input results into memory, however when the STO M operation is used, previous memory contents are cleared. When M+, SHIFT M− is used, value is added or subtracted to or from present sum in memory.

**Example** **Inputting 456 into memory "M" using STO M procedure. Memory already contains value of 123:**

| | |
|---|---|
| AC 123 STO M | M =<br>123. |
| AC 456 STO M | M =<br>456. |
| AC | – |
| RCL M | M =<br>456. |

**Example** **Inputting 456 into memory "M" using M+. Memory already contains value of 123:**

| | |
|---|---|
| AC 123 STO M | M =<br>123. |
| AC 456 M+ | 456<br>456. |
| AC | – |
| RCL M | M =<br>579. |

## Special functions

### ■Answer function

This unit has an answer function that stores the result of the most recent calculation. Once a numeric value or numeric expression is entered and [EXE] is pressed, the result (the answer in the case of numeric formulas) is stored by this function.
To recall the stored value, press the [Ans] key. When [Ans] is pressed, "Ans" will appear on the display, and the value can be used in subsequent calculations.
*As the "Ans" function works just like any other memory, it will be referred to as "Ans memory" in subsequent sections of this manual.

**Example** 123 + 456 = 579
789 − 579 = 210

| | |
|---|---|
| [AC] [1] [2] [3] [+] [4] [5] [6] [EXE] | 123+456<br>579. |
| [7] [8] [9] [−] [Ans] | 789−Ans_ |
| [EXE] | 789−Ans<br>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 erased even if the power of the unit is turned OFF. Each time [EXE], [SHIFT] [%], [M+], [SHIFT] [M−] and [STO] $\alpha$ ($\alpha$ = A ~ Z) is pressed, the value in the Ans memory is replaced with the new value produced by the calculation execution. When execution of a calculation results in an error, however, the Ans memory retains its current value.
*Contents of Ans memory are not altered when [RCL] $\alpha$ ($\alpha$ = A ~ Z) is used to recall contents of variable memory. Also, contents of Ans memory are not altered when variables are input when the variable input prompt is displayed.

### ■Omitting the multiplication sign (×)

When inputting a formula as it is written, from left to right, it is possible to omit the multiplication sign (×) in the following cases:

1) Before the following functions:
sin, cos, tan, sin $^{-1}$, cos $^{-1}$, tan $^{-1}$, sinh, cosh, tanh, sinh $^{-1}$, cosh $^{-1}$, tanh $^{-1}$, log, ln, $10^x$, $e^x$, $\sqrt{\ }$, $\sqrt[3]{\ }$, Pol ($x$, $y$), Rec ($r$, $\theta$)
**Ex.** 2sin30, 10log1.2, $2\sqrt{3}$, 2Pol(5, 12), etc.

2) Before fixed numbers, variables and memories:
**Ex.** 2$\pi$, 2AB, 3Ans, etc.

3) Before parentheses:
**Ex.** 3(5 + 6), (A + 1)(B − 1), etc.

### ■Continuous calculation function

Even if calculations are concluded with the [EXE] key, the result obtained can be used for further calculations. In this case, calculations are performed with 10 digits for the mantissa which is displayed.

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

| | |
|---|---|
| [AC] 3 [×] 4 [EXE] | 3×4<br>12. |
| (Continuing) [÷] 3.14 | 12./3.14_ |
| [EXE] | 12./3.14<br>3.821656051 |

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

| | |
|---|---|
| [AC] 1 [÷] 3 [×] 3 [EXE] | 1/3×3<br>1. |
| 1 [÷] 3 [EXE] | 1/3<br>0.333333333 |
| (Continuing) [×] 3 [EXE] | 0.333333333×→<br>0.999999999 |

Calculations performed with 10 digits for mantissa.

This function can be used with Type A functions ($x^2$, $x^{-1}$, $x!$, see page 27), +, −, $x^y$, $\sqrt[x]{\ }$, and °' ".

**Example** **Squaring the result of 78 ÷ 6 = 13:**

| | |
|---|---|
| [AC] 78 [÷] 6 [EXE] | 78/6<br>13. |

(Continuing) [SHIFT] [$x^2$]

13.²_

[EXE]

13.²
169.

## ■Replay function

This function stores formulas that have been executed. After execution is complete, pressing either the [⇨] or [⇦] key will display the formula executed.
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, using the [⇨] and [⇦] to move the cursor, the formula can be checked and numeric values or commands can be changed for subsequent execution.

**Example** [AC] 123 [×] 456 [EXE]

123 × 456
56088.

[⇨]

123 × 456

[EXE]

123 × 456
56088.

[⇦]

123 × 456_

**Example** $4.12 \times 3.58 + \underline{6.4} = 21.1496$
$4.12 \times 3.58 - \underline{7.1} = 7.6496$

[AC] 4.12 [×] 3.58 [+] 6.4 [EXE]

4.12 × 3.58 + 6. →
21.1496

[⇦]

← 12 × 3.58 + 6.4_

[⇦][⇦][⇦][⇦]

← 4.12 × 3.58 + 6. →

[−] 7.1

← 12 × 3.58 − 7.1_

[EXE]

4.12 × 3.58 − 7. →
7.6496

*As with the number of input characters (see page 30), 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] 123 [×] 456 [EXE]

123 × 456
56088.

[AC]

_

[⇦]

123 × 456_

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

## ■Error position display function

When an ERROR message appears during operation execution, the error can be cleared by pressing the [AC] key and the values or formula can be re-entered from the beginning. However, by pressing the [⇨] or [⇦] key, the ERROR message is cancelled and the cursor moves to the point where the error was generated.

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

[AC] 14 [÷] 0 [×] 2.3 [EXE]

Ma ERROR

[⇨] (or [⇦])

14/0 × 2.3

Cursor indicates where error is

[⇦][SHIFT][INS] 1

```
14/10×2.3
```

[EXE]

```
14/10×2.3
        3.22
```

## ■Multistatement function

- The multistatement function (using colons to separate formulas or statements) available in program calculations can also be used for manual calculations.
- The multistatement function allows formulas to be separated by colons ([2ndF][:]) to make consecutive, multiple statement calculations possible.
- When [EXE] is pressed to execute a formula input using the multistatement format, the formula is executed in order from the beginning.
- Inputting "◢" ([2ndF][◢]) in place of the colon ([2ndF][:]) will display the calculation result up to that point during execution.

**Example** 6.9 × 123 = 848.7
123 ÷ 3.2 = 38.4375

[AC] 123 [STO] [A] 6.9 [×] [ALPHA] [A] [2ndF] [◢] [ALPHA] [A] [/] 3.2 [EXE]

```
6.9×A
        848.7
Disp
```

↑ Appears on display when "◢" is used.

[EXE]

```
A/3.2
      38.4375
```

*Even if "◢" is not input at the end of a formula, the final result will be displayed.
*Consecutive calculations containing multistatements cannot be performed.

123 × 456 : × 5
└── Invalid

- Calculations can be performed while an intermediate result is displayed during execution interrupted by "◢".

**Example** [MODE] [4] 5 [×] 6 [2ndF] [◢] 7 [×] 8

```
5×6◢7×8_
        D
```

[EXE]

```
5×6
        30.
Disp    D
```

[sin] [Ans]

```
sin Ans_
Disp    D
```

[EXE]

```
sin Ans
        0.5
Disp    D
```

When interrupt operation is completed, press [EXE] once again to execute.

[EXE]

```
7×8
        56.
        D
```

## Scientific function calculations

### ■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 unit of angular measurement (degrees, radians, grads) is set by pressing [MODE] followed by a value from [4] to [6].

$(90° = \frac{\pi}{2}$ radians $= 100$ grads)

- 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 switched OFF.
- This operation is invalid in the BASE-N mode. When in the BASE-N mode, make setting after pressing [MODE] followed by [0].

| Example | Operation | Display (Lower) |
|---|---|---|
| $\sin 63°52'41'' = 0.897859012$ | [MODE][4] → "D"<br>[sin]63[°'"]52[°'"]41[°'"][EXE] | 0.897859012 |
| $\cos\left(\frac{\pi}{3}\text{ rad}\right) = 0.5$ | [MODE][5] → "R"<br>[cos]([SHIFT][π][÷]3)[EXE] | 0.5 |
| $\tan(-35\text{gra})$<br>$= -0.612800788$ | [MODE][6] → "G"<br>[tan][(−)]35[EXE] | −0.612800788 |
| $2\cdot\sin 45° \times \cos 65°$<br>$= 0.597672477$ | [MODE][4] → "D"<br>2[×][sin]45[×][cos]65[EXE]<br>(Can be omitted.) | 0.597672477 |
| $\sin^{-1}0.5 = 30°$<br>(Determines $x$ for $\sin x = 0.5$) | [SHIFT][sin]0.5[EXE]<br>(Can be entered as .5) | 30. |
| $\cos^{-1}\frac{\sqrt{2}}{2} = 0.785398163$ rad<br>$= \frac{\pi}{4}$ rad | [MODE][5] → "R"<br>[SHIFT][cos]([√]2[÷]2)[EXE]<br>[÷][SHIFT][π][EXE] | 0.785398163<br>0.249999999 |
| $\tan^{-1}0.741 = 36.53844577°$<br>$= 36°32'18.4''$ | [MODE][4] → "D"<br>[SHIFT][tan]0.741[EXE]<br>[SHIFT][°'"] | 36.53844577<br>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.

| | | |
|---|---|---|
| $2.5\times(\sin^{-1}0.8 - \cos^{-1}0.9)$<br>$= 68°13'13.53''$ | 2.5[×]([SHIFT][sin]0.8[−]<br>[SHIFT][cos]0.9)[EXE][SHIFT][°'"] | 68°13°13.53 |

### ■Logarithmic and exponential functions

- The following operation is invalid in the BASE-N mode. When in the BASE-N mode, carry out calculation after pressing [MODE] followed by [0].

| Example | Operation | Display |
|---|---|---|
| $\log 1.23$ ($\log_{10}1.23$) = $8.9905111 \times 10^{-2}$ | [log]1.23[EXE] | 0.089905111 |
| $\ln 90$ ($\log 90$) = 4.49980967 | [ln]90[EXE] | 4.49980967 |
| $\log 456 \div \ln 456$<br>$= 0.434294481$<br>(log/ln ratio = constant M) | [log]456[÷][ln]456[EXE] | 0.434294481 |
| $10^{1.23} = 16.98243652$<br>(To obtain the anti-logarithm of common logarithm 1.23) | [SHIFT][$10^x$]1.23[EXE] | 16.98243652 |
| $e^{4.5} = 90.0171313$<br>(To obtain the anti-logarithm of natural logarithm 4.5) | [SHIFT][$e^x$]4.5[EXE] | 90.0171313 |
| $10^4 \cdot e^{-4} + 1.2 \cdot 10^{2.3}$<br>$= 422.5878667$ | [SHIFT][$10^x$]4[×][SHIFT][$e^x$][(−)]4[+]<br>1.2[×][SHIFT][$10^x$]2.3[EXE] | 422.5878667 |
| $(-3)^4 = (-3)\times(-3)\times(-3)\times(-3)$<br>$= 81$ | [(−)]3[)][$x^2$]4[EXE] | 81. |
| $-3^4 = -(3\times3\times3\times3) = -81$ | [(−)]3[$x^2$]4[EXE] | −81. |
| $5.6^{2.3} = 52.58143837$ | 5.6[$x^2$]2.3[EXE] | 52.58143837 |
| $\sqrt[7]{123}$ ($= 123^{\frac{1}{7}}$)<br>$= 1.988647795$ | 7[SHIFT][√]123[EXE] | 1.988647795 |

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

*$x^y$ and $\sqrt[x]{}$ given calculation priority over × and /.

## ■Performing hyperbolic and inverse hyperbolic functions

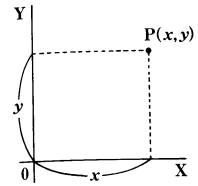
•The following operation is invalid in the BASE-N mode. When in the BASE-N mode, carry out calculation after pressing MODE followed by 0.

| Example | Operation | Display (Lower) |
|---|---|---|
| $\sinh 3.6=18.28545536$ | hyp sin 3.6 EXE | 18.28545536 |
| $\cosh 1.23=1.856761057$ | hyp cos 1.23 EXE | 1.856761057 |
| $\tanh 2.5=0.986614298$ | hyp tan 2.5 EXE | 0.986614298 |
| $\cosh 1.5-\sinh 1.5$ $=0.22313016$ $=e^{-1.5}$ | hyp cos 1.5 − hyp sin 1.5 EXE<br>(Continuing) In Ans EXE | 0.22313016<br>−1.5 |
| (Proof of $\cosh x \pm \sinh x = e^{\pm x}$) | | |
| $\sinh^{-1}30=4.094622224$ | hyp SHIFT sin 30 EXE | 4.094622224 |
| $\cosh^{-1}\left(\frac{20}{15}\right)=0.795365461$ | hyp SHIFT cos ( 20 ÷ 15 ) EXE | 0.795365461 |
| Determine the value of $x$ when $\tanh 4x=0.88$ | | |
| $x=\frac{\tanh^{-1}0.88}{4}$ $=0.343941914$ | hyp SHIFT tan 0.88 ÷ 4 EXE | 0.343941914 |
| $\sinh^{-1}2\times\cosh^{-1}1.5$ $=1.389388923$ | hyp SHIFT sin 2 × hyp SHIFT cos 1.5 EXE | 1.389388923 |
| $\sinh^{-1}\left(\frac{2}{3}\right)+\tanh^{-1}\left(\frac{4}{5}\right)$ $=1.723757406$ | hyp SHIFT sin ( 2 ÷ 3 ) + hyp | 1.723757406 |

## ■Coordinate transformation

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

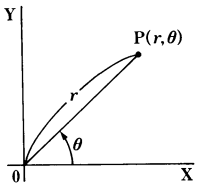
•Rectangular coordinates




Pol

Rec

•Polar coordinates




•Calculation results are stored in variable memory V and variable memory W. Contents of variable memory V are displayed initially. To display contents of memory W, press RCL W.

| | V | W |
|---|---|---|
| Pol | $r$ | $\theta$ |
| Rec | $x$ | $y$ |

•With polar coordinates, $\theta$ can be calculated within a range of $-180° < \theta \leqq 180°$. (Calculation range is the same with radians or grads.)

•The following operation is invalid in the BASE-N mode. When in the BASE-N mode, carry out calculation after pressing MODE followed by 0.

| Example | Operation | Display (Lower) |
|---|---|---|
| If $x=14$ and $y=20.7$, what are $r$ and $\theta$°? | MODE 4 → "D"<br>SHIFT Pol( 14 , 20.7 ) EXE<br>(Continuing) RCL W SHIFT ▶°′″ | 24.98979792 ($r$)<br>55°55°42.2 ($\theta$) |
| If $x=7.5$ and $y=-10$, what are $r$ and $\theta$ rad? | MODE 5 → "R"<br>SHIFT Pol( 7.5 , (−) 10 ) EXE<br>(Continuing) RCL W | 12.5 ($r$)<br>−0.927295218 ($\theta$) |
| If $r=25$ and $\theta=56°$, what are $x$ and $y$? | MODE 4 → "D"<br>SHIFT Rec( 25 , 56 ) EXE<br>(Continuing) RCL W | 13.97982259 ($x$)<br>20.72593931 ($y$) |
| If $r=4.5$ and $\theta=\frac{2}{3}\pi$rad, what are $x$ and $y$? | MODE 5 → "R"<br>SHIFT Rec( 4.5 , ( 2 ÷ 3 ×<br>SHIFT π ) ) EXE<br>(Continuing) RCL W | −2.25 ($x$)<br>3.897114317 ($y$) |

## ■Permutation and combination

**•Total number of permutations**

$$nPr = \frac{n!}{(n-r)!}$$

**•Total number of combinations**

$$nCr = \frac{n!}{r!\,(n-r)!}$$

•The following operation is invalid in the BASE-N mode. When in the BASE-N mode, carry out calculation after pressing MODE followed by 0.

| Example | Operation | Display (Lower) |
|---|---|---|
| Taking any four out of ten items and arranging them in a row, how many different arrangements are possible?<br>$_{10}P_4 = 5040$ | 10 SHIFT nPr 4 EXE | 5040. |
| Using any four numbers from 1 to 7, how many four-digit even numbers can be formed if none of the four digits consist of the same number? ($\frac{3}{7}$ of the total number of permutations will be even.)<br>$_7P_4 \times \frac{3}{7} = 360$ | 7 SHIFT nPr 4 × 3 ÷ 7 EXE | 360. |
| If any four items are removed from a total of 10 items, how many different combinations of four items are possible?<br>$_{10}C_4 = 210$ | 10 SHIFT nCr 4 EXE | 210. |
| If 5 class officers are being selected for a class of 15 boys and 10 girls, how many combinations are possible? At least one girl must be included in each group.<br>$_{25}C_5 - {}_{15}C_5 = 50127$ | 25 SHIFT nCr 5 − 15 SHIFT nCr 5 EXE | 50127. |

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

•The following operation is invalid in the BASE-N mode. When in the BASE-N mode, carry out calculation after pressing MODE followed by 0.

| Example | Operation | Display |
|---|---|---|
| $\sqrt{2}+\sqrt{5} = 3.65028154$ | $\sqrt{\ }$ 2 + $\sqrt{\ }$ 5 EXE | 3.65028154 |
| $2^2+3^2+4^2+5^2 = 54$ | 2 SHIFT $x^3$ + 3 SHIFT $x^3$ + 4 SHIFT $x^2$ + 5 SHIFT $x^2$ EXE | 54. |
| $(-3)^2 = (-3)\times(-3) = 9$ | ( (−) 3 ) SHIFT $x^2$ EXE | 9. |
| $-3^2 = -(3\times3) = -9$ | (−) 3 SHIFT $x^2$ EXE | −9. |
| $\frac{1}{\frac{1}{3}-\frac{1}{4}} = 12$ | ( 3 SHIFT $x^{-1}$ − 4 SHIFT $x^{-1}$ ) SHIFT $x^{-1}$ EXE | 12. |
| $8!(=1\times2\times3\times\ldots\ldots\times8) = 40320$ | 8 SHIFT $x!$ EXE | 40320. |
| $\sqrt[3]{36\times42\times49} = 42$ | 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 |
| $\sqrt{13^2-5^2}+\sqrt{3^2+4^2} = 17$ | $\sqrt{\ }$ ( 13 SHIFT $x^2$ − 5 SHIFT $x^2$ ) + $\sqrt{\ }$ ( 3 SHIFT $x^2$ + 4 SHIFT $x^3$ ) EXE | 17. |
| $\sqrt{1-\sin^2 40°} = 0.766044443$<br>$= \cos 40°$ | MODE 4 → "D"<br>$\sqrt{\ }$ ( 1 − ( sin 40 ) SHIFT $x^2$ ) EXE | 0.766044443 |
| (Proof of $\cos\theta = \sqrt{1-\sin^2\theta}$) | (Continuing) SHIFT cos Ans EXE | 40. |
| $\frac{1}{2!}+\frac{1}{4!}+\frac{1}{6!}+\frac{1}{8!}$<br>$= 0.543080357$ | 2 SHIFT $x!$ SHIFT $x^{-1}$ + 4 SHIFT $x!$ SHIFT $x^{-1}$ + 6 SHIFT $x!$ SHIFT $x^{-1}$ + 8 | 0.543080357 |

## ■ Fractions

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

| Example | Operation | Display (Lower) |
|---|---|---|
| $\frac{2}{5}+3\frac{1}{4}=3\frac{13}{20}$ | 2 a⅞ 5 + 3 a⅞ 1 a⅞ 4 EXE | 3⌟13⌟20. |
| $=3.65$ | (Conversion to decimal) a⅞ | 3.65 |
| *Fractions can be converted to decimals, and then converted back to fractions. | | |
| $3\frac{456}{78}=8\frac{11}{13}$ (Reduced) | 3 a⅞ 456 a⅞ 78 EXE | 8⌟11⌟13. |
| | (Continuing) SHIFT d/c | 115⌟13. |
| *Fractions and improper fractions which can be reduced become reduced fractions when the 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 a⅞ 2578 + 1 a⅞ 4572 EXE | 6.066202547 -04 (Norm 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 a⅞ 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 a⅞ 3 × (-) 4 a⅞ 5 − 5 a⅞ 6 EXE | −1⌟1⌟10. |
| $\frac{1}{2}\times\frac{1}{3}+\frac{1}{4}\times\frac{1}{5}=\frac{13}{60}$ | 1 a⅞ 2 × 1 a⅞ 3 + 1 a⅞ 4 × 1 a⅞ 5 EXE | 13⌟60. |
| $\frac{\frac{1}{2}}{3}=\frac{1}{6}$ | ( 1 a⅞ 2 ) a⅞ 3 EXE | 1⌟6 |
| $\frac{1}{\frac{1}{3}+\frac{1}{4}}=1\frac{5}{7}$ | 1 a⅞ ( 1 a⅞ 3 + 1 a⅞ 4 ) EXE | 1⌟5⌟7 |
| *When parentheses are used in numerators or denominators, it is possible to carry out fractional calculations. | | |

## ■ Engineering symbol calculations

•This unit allows engineering calculations utilizing engineering symbols.

•The Eng mode is specified by pressing MODE . in the COMP mode (MODE 0), LR mode (MODE 2), SD mode (MODE 3) ("Eng" symbol appears on display). To exit from this mode, press MODE . once again.

| Operation | Unit | Unit symbol |
|---|---|---|
| SHIFT k (=6) | $10^{3}$ | k (kilo) |
| SHIFT M (=7) | $10^{6}$ | M (mega) |
| SHIFT G (=8) | $10^{9}$ | G (giga) |
| SHIFT T (=9) | $10^{12}$ | T (tera) |
| SHIFT m (=5) | $10^{-3}$ | m (milli) |
| SHIFT μ (=4) | $10^{-6}$ | μ (micro) |
| SHIFT n (=3) | $10^{-9}$ | n (nano) |
| SHIFT p (=2) | $10^{-12}$ | p (pico) |
| SHIFT f (=1) | $10^{-15}$ | f (femto) |

| Example | Operation | Display (Lower) |
|---|---|---|
| | MODE . → "Eng" | |
| 999k (kilo) + 25k (kilo) = 1.024M (mega) | 999 SHIFT k + 25 SHIFT k EXE | 1.024 M |
| 100m (milli) × 5μ (micro) = 500n (nano) | 100 SHIFT m × 5 SHIFT μ EXE | 500. n |
| 9 ÷ 10 = 0.9 = 900m (milli) | 9 ÷ 10 EXE | 900. m |
| | SHIFT ENG | 0.9 |
| | ENG | 900. m |

## Binary, octal, decimal, hexadecimal calculations

- Binary, octal, decimal and hexadecimal calculations, conversions and logical operations are performed in the BASE-N mode (press [MODE][1]).
- The number system (2, 8, 10, 16) is set by respectively pressing [BIN], [OCT], [DEC] or [HEX]. 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.
- 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 |

- 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 (Upper) |
|---|---|
| [A] (= [(−)]) | A |
| [B] (= [°'"]) | B |
| [C] (= [hyp]) | C |
| [D] (= [sin]) | D |
| [E] (= [cos]) | E |
| [F] (= [tan]) | F |

- Negative numbers in binary, octal and hexadecimal are expressed as two's complements.
- Number of digits displayed in each number system

| Number system | Number of digits displayed |
|---|---|
| Binary | Up to 32 digits (8 digits × 4 blocks) |
| Octal | Up to 11 digits (8 digits + 3 digits) |
| Decimal | Up to 10 digits |
| Hexadecimal | Up to 8 digits |

- Calculation range (in BASE-N mode)

| | | |
|---|---|---|
| Binary | Positive | : $0111111111111111111111111111111 \geqq x \geqq 0$ |
| | Negative | : $1111111111111111111111111111111 \geqq x \geqq 10000000000000000000000000000000$ |
| Octal | Positive | : $17777777777 \geqq x \geqq 0$ |
| | Negative | : $37777777777 \geqq x \geqq 20000000000$ |
| Decimal | Positive | : $2147483647 \geqq x \geqq 0$ |
| | Negative | : $-1 \geqq x \geqq -2147483648$ |
| Hexadecimal | Positive | : $7FFFFFFF \geqq x \geqq 0$ |
| | Negative | : $FFFFFFFF \geqq x \geqq 80000000$ |

## Binary and octal block display

In the binary mode, a maximum of 32 digits are displayed in 4 blocks of 8 digits. In the octal mode, a maximum of 11 digits are displayed in one block of 8 digits, and a second block of 3 digits.

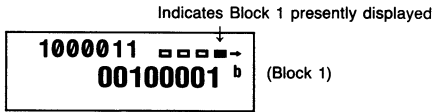
**Example** **In binary mode:**

| Block 4 | Block 3 | Block 2 | Block 1 |
|---|---|---|---|
| 10000111 | 01100101 | 01000011 | 00100001 |
| ←8 digits→ | ←8 digits→ | ←8 digits→ | ←8 digits→ |
| ←32 digits→ | | | |

**In octal mode:**

| Block 2 | Block 1 |
|---|---|
| 012 | 34567012 |
| ←3 digits→ | ←8 digits→ |
| ←11 digits→ | |

- In the binary mode, Block 1 is displayed immediately after calculation. Other blocks are displayed by pressing the [BLOCK] key. The block number increments each time you press the [BLOCK] key. The 4-digit symbol display at the upper right indicates the block presently being displayed. To reverse the order (shift from Block 4 to Block 3, etc.), press [SHIFT][BLOCK].



Indicates Block 1 presently displayed

```
1000011 ▭▭▭■→
   00100001 b
```

(Block 1)

Block 2

[BLOCK] → 1000011 ▫▫■▫→
01000011 b (Block 2)

Block 3

[BLOCK] → 1000011 ▫■▫▫→
01100101 b (Block 3)

Block 4

[BLOCK] → 1000011 ■▫▫▫→
10000111 b (Block 4)

[BLOCK] → 1000011 ▫▫▫■→
00100001 b (Returns to Block 1)

[SHIFT] [BLOCK] → 1000011 ■▫▫▫→
10000111 b (Returns to Block 4)

- In the octal mode, Block 1 is displayed immediately after calculation. Block 2 displayed by pressing the [BLOCK] key. The block display switches between Block 1 and Block 2 each time you press the [BLOCK] key. The 2-digit symbol display at the upper right indicates the block presently being displayed.

012345670 ▫■→
34567012 o (Block 1)

[BLOCK] → 012345670 ■▫→
012 o (Block 2)

[BLOCK] → 012345670 ▫■→
34567012 o (Returns to Block 1)

## ■Binary, octal, decimal, hexadecimal conversions

There are two ways to perform reciprocal binary, octal, decimal and hexadecimal conversions.

### • Conversion using number system specification key

Value from a different number system input when a specific number system mode is being used.

| Example | Operation | Display (Lower) |
|---|---|---|
| What are the decimal values for $2A_{16}$ and $274_8$? | [MODE] [1] | |
| | [Dec] → "d" | d |
| | [SHIFT] [h] 2A [EXE] | 42 d |
| | [SHIFT] [o] 274 [EXE] | 188 d |
| What are the hexadecimal values for $123_{10}$ and $1010_2$? | [Hex] → "H" | |
| | [SHIFT] [d] 123 [EXE] | 0000007b H |
| | [SHIFT] [b] 1010 [EXE] | 0000000A H |
| What are the octal values for $15_{16}$ and $1100_2$? | [Oct] → "o" | |
| | [SHIFT] [h] 15 [EXE] | ▫■ 00000025 o |
| | [SHIFT] [b] 1100 [EXE] | ▫■ 00000014 o |
| What are the binary values for $36_{10}$ and $2C_{16}$? | [Bin] → "b" | |
| | [SHIFT] [d] 36 [EXE] | ▫▫▫■ 00100100 b |
| | [SHIFT] [h] 2C [EXE] | ▫▫▫■ 00101100 b |

### • Conversion using number system mode key

Calculation results can be converted to any specified number system by using the corresponding number system mode key.

| Example | Operation | Display (Lower) |
|---|---|---|
| How is $22_{10}$ expressed in binary, octal and hexadecimal number systems? | [MODE] [1] | |
| | [Dec] → "d" | |
| | 22 [EXE] | 22 d |
| | [Bin] | ▫▫▫■ 00010110 b |
| | [Oct] | ▫■ 00000026 o |
| | [Hex] | 00000016 H |

## ■Negative expressions

| Example | Operation | Display (Lower) |
|---|---|---|
| | [MODE] [1] | |
| How is $110010_2$ expressed as a negative? | [Bin] → "b" | |
| | [Neg] 110010 [EXE] | 11001110 b |
| | [BLOCK] | 11111111 b |
| | [BLOCK] | 11111111 b |
| | [BLOCK] | 11111111 b |
| How is $72_8$ expressed as a negative? | [Oct] → "o" | |
| | [Neg] 72 [EXE] | 77777706 o |
| | [BLOCK] | 377 o |
| How is $3A_{16}$ expressed as a negative? | [Hex] → "H" | |
| | [Neg] 3A [EXE] | FFFFFFC6 H |

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

| Example | Operation | Display (Lower) |
|---|---|---|
| | [MODE] [1] | |
| $10111_2 + 11010_2 = 110001_2$ | [Bin] → "b" | |
| | 10111 [+] 11010 [EXE] | 00110001 b |
| $B47_{16} - DF_{16} = A68_{16}$ | [Hex] → "H" | |
| | B47 [−] DF [EXE] | 00000A68 H |
| $123_8 \times ABC_{16} = 37AF4_{16}$ | [SHIFT] [o] 123 [×] ABC [EXE] | 00037AF4 H |
| $= 228084_{10}$ | [Dec] | 228084 d |
| $1F2D_{16} - 100_{10} = 7881_{10}$ | [SHIFT] [h] 1F2D [−] 100 [EXE] | 7881 d |
| $= 1EC9_{16}$ | [Hex] | 00001EC9 H |
| $7654_8 \div 12_{10} = 334.3333333_{10}$ | [Dec] → "d" | |
| $= 516_8$ | [SHIFT] [o] 7654 [÷] 12 [EXE] | 334 d |
| | [Oct] | 00000516 o |

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

| | | |
|---|---|---|
| $1234_{10} + 1EF_{16} \div 24_8$ | [SHIFT] [d] 1234 [+] [SHIFT] [h] | |
| $= 2352_8$ | 1EF [÷] 24 [EXE] | 00002352 o |
| $= 1258_{10}$ | [Dec] | 1258 d |

*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 (Lower) |
|---|---|---|
| | [MODE] [1] | |
| $19_{16}$ AND $1A_{16} = 18_{16}$ | [Hex] → "H" | |
| | 19 [SHIFT] [and] 1A [EXE] | 00000018 H |
| $1110_2$ AND $36_8 = 1110_2$ | [Bin] → "b" | |
| | 1110 [SHIFT] [and] [SHIFT] [o] 36 [EXE] | 00001110 b |
| $23_8$ OR $61_8 = 63_8$ | [Oct] → "o" | |
| | 23 [SHIFT] [or] 61 [EXE] | 00000063 o |
| $120_{16}$ OR $1101_2 = 12D_{16}$ | [Hex] → "H" | |
| | 120 [SHIFT] [or] [SHIFT] [b] 1101 [EXE] | 0000012d H |
| $1010_2$ AND ($A_{16}$ OR $7_{16}$) | [Bin] → "b" | |
| $= 1010_2$ | 1010 [SHIFT] [and] ( [SHIFT] [h] A | |
| | [SHIFT] [or] [SHIFT] [h] 7 ) [EXE] | 00001010 b |
| $5_{16}$ XOR $3_{16} = 6_{16}$ | [Hex] → "H" | |
| | 5 [SHIFT] [xor] 3 [EXE] | 00000006 H |
| $2A_{16}$ XNOR $5D_{16} = FFFFFF88_{16}$ | [Hex] → "H" | |
| | 2A [SHIFT] [xnor] 5D [EXE] | FFFFFF88 H |
| Negation of $1234_8$ | [Oct] → "o" | |
| | [Not] 1234 [EXE] | 77776543 o |
| Negation of $2FFFED_{16}$ | [Hex] → "H" | |
| | [Not] 2FFFED [EXE] | FFd00012 H |

# ■Statistical calculations

This unit can be used to make statistical calculations including standard deviation in the SD mode, and regression calculation in the LR mode.

## ■Standard deviation

In the SD mode, calculations including 2 types of standard deviation formulas, mean, number of data, sum of data, and sum of squares can be performed.

### • Data input

1. Press [MODE] [3] to specify the SD mode.
2. Press [2ndF] [Scl] [EXE] to clear the statistical memories.
3. Input data, pressing [DT] key (=[M+]) each time a new piece of data is entered. For negative values, press [(−)] followed by [DT].

**Example** **Data: 10, 20, 30**
**Key operation: 10 [DT] 20 [DT] 30 [DT]**

*When multiples of the same data are input, two different entry methods are possible:

**Example 1** **Data: 10, 20, 20, 30**
**Key operation: 10 [DT] 20 [DT] [DT] 30 [DT]**

The previously entered data is entered again each time the [DT] key is pressed without entering data (in this case 20 is re-entered).

**Example 2** **Data: 10, 20, 20, 20, 20, 20, 20, 30**
**Key operation: 10 [DT] 20 [SHIFT] [:] 6 [DT] 30 [DT]**

By pressing [SHIFT] and then entering a semicolon followed by a value that represents the number of items the data is repeated (6, in this case) and the [DT] key, the multiple data entries (for 20, in this case) are made automatically.

### • Deleting input data

There are various ways to delete value data, depending on how and where it was entered.

**Example 1** **40 [DT] 20 [DT] 30 [DT] 50 [DT]**
**To delete 50, press [SHIFT] [CL].**

**Example 2** **40 [DT] 20 [DT] 30 [DT] 50 [DT]**
**To delete 20, press 20 [SHIFT] [CL].**

**Example 3** **30 [DT] 50 [DT] 120 [SHIFT] [:]**
**To delete 120 [SHIFT] [:], press [AC].**

**Example 4** 30 DT 50 DT 120 SHIFT : 31
**To delete 120 SHIFT : 31, press AC.**

**Example 5** 30 DT 50 DT 120 SHIFT : 31 DT
**To delete 120 SHIFT : 31 DT, press SHIFT CL.**

**Example 6** 50 DT 120 SHIFT : 31 DT 40 DT 30 DT
**To delete 120 SHIFT : 31 DT, press 120 SHIFT : 31 SHIFT CL.**

**Example 7** √ 10 DT √ 20 DT √ 30 DT
**To delete √ 20 DT, press √ 20 EXE Ans SHIFT CL.**

**Note:** You need to press EXE Ans to delete a calculation result, such as $\sqrt{20}$.

**Example 8** √ 10 DT √ 20 DT √ 30 DT
**To delete √ 20 DT, press √ 20 SHIFT : (−) 1 DT.**

## •Performing calculations

The following procedures are used to perform the various standard deviation calculations.

| Key operation | Result | |
|---|---|---|
| 2ndF $x\sigma_n$ EXE | Population standard deviation $x\sigma n$ | $x\sigma_n$ = • |
| 2ndF $x\sigma_{n-1}$ EXE | Sample standard deviation $x\sigma n-1$ | $x\sigma_{n-1}$ = EXP |
| 2ndF $\bar{x}$ EXE | Mean | $\bar{x}$ = 0 |
| 2ndF $\Sigma x^2$ EXE | Sum of squares of data | $\Sigma x^2$ = 1 |
| 2ndF $\Sigma x$ EXE | Sum of data | $\Sigma x$ = 2 |
| 2ndF $n$ EXE | Number of data | $n$ = 3 |

Standard deviation and mean calculations are performed as shown below:

## •Standard deviation

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

[Using the entire data of a finite population to estimate the standard deviation for the population.]

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

[Using sample data for a population to estimate the standard deviation for the population.]

## •Mean

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

| Example | Operation | Display (Lower) |
|---|---|---|
| Data 55, 54, 51, 55, 53, 53, 54, 52 | MODE 3 → "SD" | |
| | (Memory cleared) 2ndF Scl EXE | |
| | 55 DT 54 DT 51 DT 55 DT 53 | |
| | DT DT 54 DT 52 DT | 52. |
| | *Results can be obtained in any order desired. | |
| | (Standard deviation $\sigma n$) 2ndF $x\sigma_n$ EXE | 1.316956719 |
| | (Standard deviation $\sigma n-1$) 2ndF $x\sigma_{n-1}$ EXE | 1.407885953 |
| | (Mean $x$) 2ndF $\bar{x}$ EXE | 53.375 |
| | (Number of data $n$) 2ndF $n$ EXE | 8. |
| | (Sum total $\Sigma x$) 2ndF $\Sigma x$ EXE | 427. |
| | (Sum of squares $\Sigma x^2$) 2ndF $\Sigma x^2$ EXE | 22805. |
| What is deviation of the unbiased variance, the difference between each datum, and the mean of the above data? | (Continuing) 2ndF $x\sigma_{n-1}$ SHIFT $x^2$ EXE | 1.982142857 |
| | 55 − 2ndF $\bar{x}$ EXE | 1.625 |
| | 54 − 2ndF $\bar{x}$ EXE | 0.625 |
| | 51 − 2ndF $\bar{x}$ EXE | −2.375 |
| | ⋮ | ⋮ |
| What is $x$ and $\sigma n-1$ for the following table? | 2ndF 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. |
| | 2ndF $n$ EXE | 70. |
| | 2ndF $\bar{x}$ EXE | 137.7142857 |
| | 2ndF $x\sigma_{n-1}$ EXE | 18.42898069 |

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

## ■Regression calculation

In the LR mode, calculations including linear regression, logarithmic regression, exponential regression, and power regression can be performed.

### • Linear regression

Linear regression calculations are carried out using the following formula: $y = A + Bx$.

#### •Data input

1. Press MODE 2 to specify the LR mode.
2. Press 2ndF Scl EXE to clear the statistical memories.
3. Input data in the following format:
   <$x$ data> , <$y$ data> DT

*When multiples of the same data are input, two different entry methods are possible:

**Example 1** **Data: 10/20, 20/30, 20/30, 40/50**
**Key operation: 10 , 20 DT**
**20 , 30 DT**
**DT**
**40 , 50 DT**

The previously entered data is entered again each time the DT key is pressed (in this case 20/30 is re-entered).

**Example 2** **Data: 10/20, 20/30, 20/30, 20/30, 20/30, 20/30, 40/50**
**Key operation: 10 , 20 DT**
**20 , 30 SHIFT : 5 DT**
**40 , 50 DT**

By pressing SHIFT and then entering a semicolon followed by a value that represents the number of times the data is repeated (5, in this case) and the DT key, the multiple data entries (for 20/30, in this case) are made automatically.

#### •Deleting input data

There are various ways to delete value data, depending on how and where it was entered.

**Example 1**
**10 , 40 DT**
**20 , 20 DT**
**30 , 30 DT**
**40 , 50**

To delete 40 , 50, press AC.

**Example 2**
**10 , 40 DT**
**20 , 20 DT**
**30 , 30 DT**
**40 , 50 DT**

To delete 40 , 50 DT, press SHIFT CL.

**Example 3**

To delete 20 , 20 DT, press 20 , 20 SHIFT CL.

**Example 4**
**↑ 10 , 40 DT**
**↑ 20 , 20 DT**
**↑ 30 , 30 DT**
**↑ 40 , 50 DT**

To delete ↑ 20 , 20 DT, press ↑ 20 EXE Ans , 20 SHIFT CL.

**Example 5**

To delete ↑ 20 , 20 DT, press ↑ 20 , 20 SHIFT : (−) 1 DT.

#### •Performing calculations

The following procedures are used to perform the various linear regression calculations.

| Key operation | Result | |
|---|---|---|
| 2ndF A EXE | Constant term of regression A | A = STO |
| 2ndF B EXE | Regression coefficient B | B = RCL |
| 2ndF r EXE | Correlation coefficient $r$ | r = ( |
| 2ndF $\hat{x}$ EXE | Estimated value of $x$ | $\hat{x}$ = ) |
| 2ndF $\hat{y}$ EXE | Estimated value of $y$ | $\hat{y}$ = , |

The regression formula is $y = A + Bx$. The constant term of regression A, regression coefficient B, correlation coefficient $r$, estimated value of $x$, and estimated value of $y$ are calculated as shown below:

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

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

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

| Example | Operation | Display (Lower) |
|---|---|---|
| •Temperature and length of a steel bar<br><br>Temp. / Length<br>10°C / 1003mm<br>15°C / 1005mm<br>20°C / 1010mm<br>25°C / 1011mm<br>30°C / 1014mm<br><br>Using this table, the regression formula and correlation coefficient can be obtained. Based on the coefficient formula, the length of the steel bar at 18°C and the temperature at 1000mm can be estimated. Furthermore, the critical coefficient ($r^2$) and covariance<br>$\left(\frac{\Sigma xy - n \cdot \bar{x} \cdot \bar{y}}{n-1}\right)$<br>can also be calculated. | MODE 2 → "LR"<br>(Memory clear) 2ndF Scl EXE<br>10 , 1003 DT<br>15 , 1005 DT<br>20 , 1010 DT<br>25 , 1011 DT<br>30 , 1014 DT<br><br>(Constant term A) 2ndF A EXE<br>(Regression coefficient B) 2ndF B EXE<br>(Correlation coefficient $r$) 2ndF r EXE<br>(Length at 18°C) 18 2ndF $\hat{y}$ EXE<br>(Temperature at 1000mm) 1000 2ndF $\hat{x}$ EXE<br>(Critical coefficient) 2ndF r SHIFT $x^2$ EXE<br>(Covariance) ( 2ndF $\Sigma xy$ − 2ndF $n$<br>× 2ndF $\bar{x}$ × 2ndF $\bar{y}$ ) ÷ (<br>2ndF $n$ − 1 ) EXE | 10.<br>15.<br>20.<br>25.<br>30.<br><br>997.4<br>0.56<br>0.982607368<br>1007.48<br><br>4.642857143<br>0.965517241<br><br>35. |

## • Logarithmic regression

Logarithmic regression calculations are carried out using the following formula: $y = A + B \cdot \ln x$.

### •Data input

1. Press MODE 2 to specify the LR mode.
2. Press 2ndF Scl EXE to clear the statistical memories.
3. Input data in the following format:
   ln <$x$ data> , <$y$ data> DT

*To make multiple entries of the same data, follow procedures described for linear regression, however always press ln before inputting $x$ data.

### •Deleting input data

To delete input data, follow the procedures described for linear regression, but be sure to use the ln key when specifying $x$ data for deletion.

**Example 1**

To delete ln 10 , 20 DT, press ln 10 EXE Ans , 20 SHIFT CL

**Example 2**

To delete ln 10 , 20 DT, press ln 10 , 20 SHIFT ; (−) 1 DT.

### •Performing calculations

The following procedures are used to perform the various calculations.

| Key operation | Result | |
|---|---|---|
| 2ndF A EXE | Constant term of regression A | A = STO |
| 2ndF B EXE | Regression coefficient B | B = RCL |
| 2ndF r EXE | Correlation coefficient $r$ | r = ( |
| $y$ 2ndF $\hat{x}$ EXE SHIFT $e^x$ Ans EXE | Estimated value of $x$ | $\hat{x}$ = ) |
| ln $x$ 2ndF $\hat{y}$ EXE | Estimated value of $y$ | $\hat{y}$ = , |

If we assume that $\ln x = x$, the logarithmic regression formula $y = A + B \cdot \ln x$ becomes the linear regression formula $y = a + bx$. Therefore, the formulas for constant term A, regression coefficient B and correlation coefficient $r$ are identical for logarithmic and linear regression.

A number of logarithmic regression calculation results differ from those produced by linear regression. Note the following:

| Linear regression | Logarithmic regression |
|---|---|
| $\Sigma x$ | $\Sigma \ln x$ |
| $\Sigma x^2$ | $\Sigma (\ln x)^2$ |
| $\Sigma xy$ | $\Sigma \ln x \cdot y$ |

| Example | Operation | Display (Lower) |
|---|---|---|
| $xi$ / $yi$<br>29 / 1.6<br>50 / 23.5<br>74 / 38.0<br>103 / 46.4<br>118 / 48.9<br><br>Through logarithmic regression of the above data, the regression formula and correlation coefficient are obtained. Furthermore, respective estimated values $\hat{y}$ and $\hat{x}$ can be obtained for $xi = 80$ and $yi = 73$ using the regression formula. | MODE 2 → "LR"<br>2ndF Scl EXE<br>ln 29 , 1.6 DT<br>ln 50 , 23.5 DT<br>ln 74 , 38.0 DT<br>ln 103 , 46.4 DT<br>ln 118 , 48.9 DT<br>(Constant term A) 2ndF A EXE<br>(Regression coefficient B) 2ndF B EXE<br>(Correlation coefficient $r$) 2ndF r EXE<br>($\hat{y}$ when $xi = 80$) ln 80 2ndF $\hat{y}$ EXE<br>($\hat{x}$ when $yi = 73$)<br>73 2ndF $\hat{x}$ EXE SHIFT $e^x$ Ans EXE | 3.36729583<br>3.912023005<br>4.304065093<br>4.634728988<br>4.770684624<br>−111.1283976<br>34.02014749<br>0.994013946<br>37.94879482<br><br>224.1541314 |

## • Exponential regression

Exponential regression calculations are carried out using the following formula: $y = A \cdot e^{B \cdot x}$ $(\ln y = \ln A + Bx)$

### •Data input

1. Press [MODE] [2] to specify the LR mode.
2. Press [2ndF] [Scl] [EXE] to clear the statistical memories.
3. Input data in the following format:
   $<x$ data$>$ [,] [ln] $<y$ data$>$ [DT]

*To make multiple entries of the same data, follow procedures described for linear regression, however always press [ln] before inputting $y$ data.

### •Deleting input data

To delete input data, follow the procedures described for linear regression, but be sure to use the [ln] key when specifying $y$ data for deletion.

**Example 1**

To delete 10 [,] [ln] 20 [DT], press [ln] 20 [EXE] 10 [,] [Ans] [SHIFT] [CL].

**Example 2**

To delete 10 [,] [ln] 20 [DT], press 10 [,] [ln] 20 [SHIFT] [:] [(−)] 1 [DT].

### •Performing calculations

The following procedures are used to perform the various calculations.

| Key operation | Result | |
|---|---|---|
| [SHIFT] [$e^x$] [2ndF] [A] [EXE] | Constant term of regression A | [A] = [STO] |
| [2ndF] [B] [EXE] | Regression coefficient B | [B] = [RCL] |
| [2ndF] [r] [EXE] | Correlation coefficient $r$ | [r] = [(] |
| [ln] $y$ [2ndF] [$\hat{x}$] [EXE] | Estimated value of $x$ | [$\hat{x}$] = [)] |
| $x$ [2ndF] [$\hat{y}$] [EXE] [SHIFT] [$e^x$] [Ans] [EXE] | Estimated value of $y$ | [$\hat{y}$] = [,] |

If we assume that $\ln y = y$ and $\ln A = a'$, the exponential regression formula $y = A \cdot e^{B \cdot x}$ $(\ln y = \ln A + Bx)$ becomes the linear regression formula $y = a + bx$. Therefore, the formulas for constant term A, regression coefficient B and correlation coefficient $r$ are identical for exponential and linear regression.

A number of exponential regression calculation results differ from those produced by linear regression. Note the following:

| Linear regression | Exponential regression |
|---|---|
| $\Sigma y$ | $\Sigma \ln y$ |
| $\Sigma y^2$ | $\Sigma (\ln y)^2$ |
| $\Sigma xy$ | $\Sigma x \cdot \ln y$ |

| Example | Operation | Display (Lower) |
|---|---|---|
| $xi$ \| $yi$<br>6.9 \| 21.4<br>12.9 \| 15.7<br>19.8 \| 12.1<br>26.7 \| 8.5<br>35.1 \| 5.2 | [MODE] [2] → "LR" | |
| | [2ndF] [Scl] [EXE] | |
| | 6.9 [,] [ln] 21.4 [DT] | 6.9 |
| | 12.9 [,] [ln] 15.7 [DT] | 12.9 |
| | 19.8 [,] [ln] 12.1 [DT] | 19.8 |
| | 26.7 [,] [ln] 8.5 [DT] | 26.7 |
| | 35.1 [,] [ln] 5.2 [DT] | 35.1 |
| Through exponential regression of the above data, the regression formula and correlation coefficient are obtained. Furthermore, the regression formula is used to obtain the respective estimated values of $\hat{y}$ and $\hat{x}$, when $xi = 16$ and $yi = 20$. | (Constant term A) [SHIFT] [$e^x$] [2ndF] [A] [EXE] | 30.49758742 |
| | (Regression coefficient B) [2ndF] [B] [EXE] | −0.049203708 |
| | (Correlation coefficient $r$) [2ndF] [r] [EXE] | −0.997247351 |
| | ($\hat{y}$ when $xi = 16$)<br>16 [2ndF] [$\hat{y}$] [EXE] [SHIFT] [$e^x$] [Ans] [EXE] | 13.87915739 |
| | ($\hat{x}$ when $yi = 20$) [ln] 20 [2ndF] [$\hat{x}$] [EXE] | 8.574868046 |

## • Power regression

Power regression calculations are carried out using the following formula: $y = A \cdot x^B$ $(\ln y = \ln A + B \ln x)$

### •Data input

1. Press [MODE] [2] to specify the LR mode.
2. Press [2ndF] [Scl] [EXE] to clear the statistical memories.
3. Input data in the following format:
   [ln] $<x$ data$>$ [,] [ln] $<y$ data$>$ [DT]

*To make multiple entries of the same data, follow procedures described for linear regression, however always press [ln] before inputting $x$ data and $y$ data.

### •Deleting input data

To delete input data, follow the procedures described for linear regression, but be sure to use the [ln] key when specifying $x$ and $y$ data for deletion.

**Example 1**

To delete [ln] 10 [,] [ln] 20 [DT], press [ln] 10 [STO] [A] [ln] 20 [EXE] [ALPHA] [A] [,] [Ans] [SHIFT] [CL].

**Example 2**

To delete [ln] 10 [,] [ln] 20 [DT], press [ln] 10 [,] [ln] 20 [SHIFT] [:] [(−)] 1 [DT].

### •Performing calculations

The following procedures are used to perform the various calculations.

| Key operation | Result | |
|---|---|---|
| SHIFT $e^x$ 2ndF A EXE | Constant term of regression A | A = STO |
| 2ndF B EXE | Regression coefficient B | B = RCL |
| 2ndF r EXE | Correlation coefficient $r$ | r = ( |
| ln $y$ 2ndF $\hat{x}$ EXE SHIFT $e^x$ Ans EXE | Estimated value of $x$ | $\hat{x}$ = ) |
| ln $x$ 2ndF $\hat{y}$ EXE SHIFT $e^x$ Ans EXE | Estimated value of $y$ | $\hat{y}$ = , |

If we assume that $\ln y = y$, $\ln A = a'$ and $\ln x = x$, the power regression formula $y = A \cdot x^B$ ($\ln y = \ln A + B \ln x$) becomes the linear regression formula $y = a + bx$. Therefore, the formulas for constant term A, regression coefficient B and correlation coefficient $r$ are identical for power and linear regression.

A number of power regression calculation results differ from those produced by linear regression. Note the following:

| Linear regression | Power regression |
|---|---|
| $\Sigma x$ | $\Sigma \ln x$ |
| $\Sigma x^2$ | $\Sigma (\ln x)^2$ |
| $\Sigma y$ | $\Sigma \ln y$ |
| $\Sigma y^2$ | $\Sigma (\ln y)^2$ |
| $\Sigma xy$ | $\Sigma \ln x \cdot \ln y$ |

**Example**

| $xi$ | $yi$ |
|---|---|
| 28 | 2410 |
| 30 | 3033 |
| 33 | 3895 |
| 35 | 4491 |
| 38 | 5717 |

Through power regression of the above data, the regression formula and correlation coefficient are obtained. Furthermore, the regression formula is used to obtain the respective estimated values of $\hat{y}$ and $\hat{x}$, when $xi = 40$ and $yi = 1000$.

| Operation | Display (Lower) |
|---|---|
| MODE 2 → "LR" | |
| 2ndF Scl EXE | |
| ln 28 , ln 2410 DT | 3.33220451 |
| ln 30 , ln 3033 DT | 3.401197382 |
| ln 33 , ln 3895 DT | 3.496507561 |
| ln 35 , ln 4491 DT | 3.555348061 |
| ln 38 , ln 5717 DT | 3.63758616 |
| (Constant term A) SHIFT $e^x$ 2ndF A EXE | 0.238801082 |
| (Regression coefficient B) 2ndF B EXE | 2.771866148 |
| (Correlation coefficient $r$) 2ndF r EXE | 0.998906256 |
| ($\hat{y}$ when $xi = 40$) ln 40 2ndF $\hat{y}$ EXE SHIFT $e^x$ Ans EXE | 6587.674743 |
| ($\hat{x}$ when $yi = 1000$) ln 1000 2ndF $\hat{x}$ EXE SHIFT $e^x$ Ans EXE | 20.26225659 |

## Formula memory function

### ■Purpose of the formula memory function

This unit has a built-in formula memory that allows consecutive execution of the same formula, using different variables. The memory function keys (IN, OUT, CALC) are used to operate this memory.

SHIFT IN: Inputs displayed formula into memory.
OUT: Displays formula retained in memory.
CALC: Used to calculate results of formula when variable is input.

There is only a single formula memory (formulas connected as multistatements are counted as a single formula). The maximum formula length is 127 steps. Calculation modes are retained in memory along with the formula, so the mode is recalled along with the formula in memory. Execution of the formula stored in memory is carried out by pressing the CALC key.

**Example 1** **To memorize, display and calculate the following formula:**
$Y = X^2 + 3X - 12$

Formula input

ALPHA Y 2ndF = ALPHA X SHIFT $x^2$ +
3 ALPHA X − 12

```
Y=X²+3X−12_
                D
```

Formula stored into memory

SHIFT IN

```
_
                D
```

Check formula

OUT

```
Y=X²+3X−12_
                D
```

Calculation

CALC

```
X?
              0.
                D
```

Value input for variable

7 [EXE]

Y=X²+3X−12
58.
D

[EXE] *

X?
7.
D

8 [EXE]

Y=X²+3X−12
76.
D

* Press [CALC] key in place of [EXE] key to execute calculation.

When formula is displayed, corrections or alterations can also be made.

**Example 2** **Changing [Y = X² + 3X − 12] to [Y = X² + 5X − 12]:**

Formula displayed

[OUT]

Y=X²+3X−12_
D

Move cursor to position where correction is to be made.

[⇦][⇦][⇦][⇦][⇦]

Y=X²+3X−12
D

Make correction

5

Y=X²+5X−12
D

Input into memory

[SHIFT][IN]

_
D

Check formula

[OUT]

Y=X²+5X−12_
D

To clear the contents of the formula memory, press [AC] followed by [SHIFT][IN]

## • Using formula memory in creating tables

By inserting a " ◢ ", it is possible to write multiple formulas. This makes it easy to create tables, such as the one shown below.

**Example 3** **Complete the following table:**

| A | B | P = A × B | Q = A/B |
|---|---|---|---|
| 4.27 | 1.17 | | |
| 8.17 | 6.48 | | |
| 6.07 | 9.47 | | |
| 2.71 | 4.36 | | |
| 1.98 | 3.62 | | |

**Solution)**

[ALPHA][P][2ndF][=][ALPHA][A][×][ALPHA][B] [2ndF][◢][ALPHA][Q][2ndF][=][ALPHA][A][/] [ALPHA][B]

P=A×B ◢ Q=A/B_
D

[SHIFT][IN]

_
D

[CALC] (Calculation started)

A?
0.
D

4.27 [EXE] (Input A)

B?
0.
D

1.17 [EXE] (Input B)

P = A × B
4.9959
Disp D

[EXE]

Q = A/B
3.64957265
D

[EXE]

A?
4.27
D

8.17 [EXE] (Input A)

B?
1.17
D

6.48 [EXE] (Input B)

P = A × B
52.9416
Disp D

[EXE] — Q = A/B
1.260802469
D

[EXE] — A?
8.17
D

(Continuation omitted)

<NOTES>

(1) A maximum of 127 steps can be input into the formula memory, using the [IN] key.
(2) Memory contents are protected even when power is turned OFF (or when auto power OFF function turns power OFF). However, when a new formula is input into memory, the previous formula is deleted.
(3) Array variables cannot be used in formulas input into memory. If used, it will be impossible to input variables into formula on display.
(4) Variable memories can hold only one variable each.
**Example:** A×BC NO!
A×B×C YES!
(5) Consecutive calculations using [+], [−], [×], [/], and [SHIFT][$x^2$] can be performed utilizing formula calculation answers.

## • Text display

Using double quotation marks, it is possible to assign names to variables in memory.

**Example** **Write formula [A "UNIT PRICE" × B] to memory:**

Formula input [SHIFT][ALPHA][A][,,][U][N][I][T][SPACE][P][R][I][C][E][,,][ALPHA][×][ALPHA][B] — ← IT PRICE"×B_

[SHIFT][IN] — _

[CALC] — UNIT PRICE?
0.

100[EXE] — B?
0.

5[EXE] — A"UNIT PRICE →
500.

Text which is over 12 characters in length is shown from the left and followed by a "?". To view the all of the text, use the [⇦] and [⇨] key to scroll left and right.

**Example** **Write formula [A "SINGLE UNIT PRICE" × B] to memory:**

[SHIFT][ALPHA][A][,,][S][I][N][G][L][E][SPACE][U][N][I][T][SPACE][P][R][I][C][E][,,][ALPHA][×][ALPHA][B] — ← IT PRICE"×B_

[SHIFT][IN] — _

[CALC] — SINGLE UNIT? →
0.

[⇨] — ← INGLE UNIT ? →
0.

[⇨] — ← NGLE UNIT P? →
0.

[⇨] — ← GLE UNIT PR? →
0.

[⇨] — ← LE UNIT PRI? →
0.

[⇨] — ← E UNIT PRIC? →
0.

[⇨] — ← UNIT PRICE?
0.

[⇦] — ← E UNIT PRIC? →
0.

Text can be assigned to a variable memory used in a substitution formula by enclosing the text in double quotation marks. Then when the formula is executed, the text appears on the display.

**Example** **Write [A "ANSWER" = 123 × 456]:**

| | |
|---|---|
| SHIFT ALPHA A ,, A N S W E R ,, 2ndF = 123 × 456 | ← ER"=123×456_ |
| SHIFT IN | _ |
| CALC | ANSWER= 56088. |

Text which is over 12 characters in length is shown from the left and followed by a "=".

**Example** **Write [A "ABCDEFGHIJKLMN" = 123] to memory:**

| | |
|---|---|
| SHIFT ALPHA A ,, A B C D E F G H I J K L M N ,, 2ndF = 123 | ← IJKLMN"=123_ |
| SHIFT IN | _ |
| CALC | ABCDEFGHIJK= → 123. |

*You can use the ◁ and ▷ key to scroll the text left and right while execution is stopped by a "Disp" symbol.

# Section 3 Integration Calculation

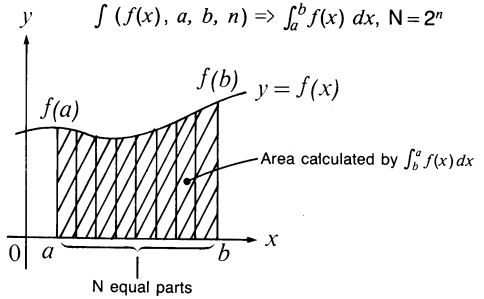
## Input of function $f(x)$ and integration calculation

# Section 3 Integration Calculation

•Integration calculation can be carried out by entering the integral calculus formula in the following format:

[SHIFT] [∫dx] $f(x)$ [,] $a$ [,] $b$ [,] $n$ [)]

- $n$: Value of $n$ when number of divisions $N = 2^n$
- $b$: Ending point
- $a$: Starting point




•Integration calculation is performed using Simpson's rule to determine function $f(x)$. Because of this, partition of the integrated area is necessary, however if the number of divisions is not specified, the unit automatically sets N according to the formula. To specify the number of divisions for $N = 2^n$, $n$ can be an integer from 1 ~ 9.

## Input of function $f(x)$ and integration calculation

(1) Press [SHIFT] [∫dx] to specify integration calculation.

(2) Input the formula for the function $f(x)$, then input integral partitions [$a$, $b$].

*$f(x)$ can use the X variable only. Anything other than X (A ~ W, Y, Z or array variables) is treated as a constant, and its memory contents are applied.

(3) Next input $n$ (number of divisions for $N = 2^{-n}$, $n$ being an integer between 1 and 9) and finish by inputting a parenthesis.

*Input of $n$ (number of divisions for $N = 2^n$) and parenthesis can be omitted. When input is omitted, N is automatically set.

(4) Press [EXE] to execute calculation.

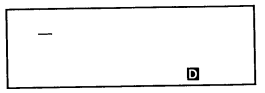
*Results are displayed in a few seconds or a number of minutes (mantissa is number of significant digits). Note that following integration data is input in memories G ~ L:

| Memory | G | H | I | J | K | L |
|---|---|---|---|---|---|---|
| Data | $a$ | $b$ | $2^n$ | $\int_a^b (f(x))dx$ | $f(a)$ | $f(b)$ |

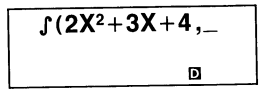
## Examples of operation

**Example 1** Calculate the following: $\int_1^5 (2x^2+3x+4)\,dx$

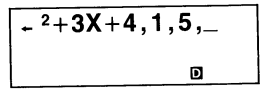
[MODE] [4] (Specify "D")



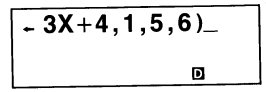

[SHIFT] [∫dx] 2 [ALPHA] [X] [SHIFT] [x²] [+] 3
[ALPHA] [X] [+] 4 [,] ($f(x)$ input)



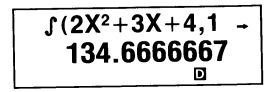

1 [,] 5 [,] ($a$, $b$ input)




6 [)] ($n$ input)

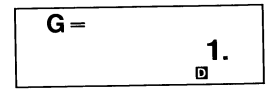



[EXE] (Calculation executed)



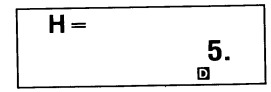

Answer displayed in approximately 15 seconds

[RCL] [G]




$a$

[RCL] [H]

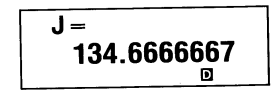



$b$

[RCL] [I]

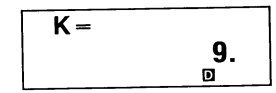
N

[RCL] [J]



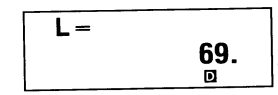

$\int_a^b f(x)dx$

[RCL] [K]




$f(a)$

[RCL] [L]




$f(b)$

**Example 2** **Calculate the following, omitting specification of the number of divisions: $\int_1^3 (\log x)\,dx$**

[MODE] [4] (Specify "D")

```
_
                D
```

[SHIFT] [∫dx] [log] [ALPHA] [X] [,]
($f(x)$ input)

```
∫(log X,
                D
```

1 [,] 3 [)] ($a$, $b$ input)

```
∫(log X,1,3)_
                D
```

Answer displayed in approximately 8 seconds

[EXE] (Calculation executed)

```
∫(log X,1,3)
          0.56277
                D
```

[RCL] [G]

```
G=
             1.
                D
```

$a$

[RCL] [H]

```
H=
             3.
                D
```

$b$

[RCL] [I]

```
I=
            32.
                D
```

N ($n=5$)

[RCL] [J]

```
J=
       0.56277
                D
```

$\int_a^b f(x)dx$

[RCL] [K]

```
K=
             0.
                D
```

$f(a)$

[RCL] [L]

```
L=
   0.477121254
                D
```

$f(b)$

## • Application of integration calculation

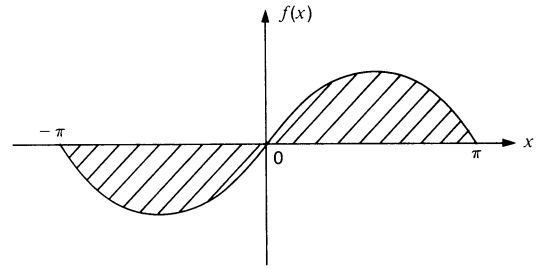
•Integrals or results of integration calculations can be used in arithmetic calculations.

**Example** $\int_a^b f(x)\,dx + \int_c^d g(x)\,dx$, $2 \times \int_a^b f(x)\,dx$, **etc.**

* Results of integration calculation cannot be used in integration calculation formulas.

•When calculating area, Abs (absolute value) should be inserted into formula:
$\int$ (Abs $f(x)$, $a$, $b$, $n$) $\Rightarrow \int_a^b |f(x)|\,dx$

**Example 3**




**Calculate the $[-\pi, \pi]$ areas of $f(x) = \sin x$. Omit input of number of divisions.**

[MODE] [5] (Specify "R")

```
_
                R
```

[SHIFT] [∫dx] [SHIFT] [Abs] [sin] [ALPHA] [X] [,]
($f(x)$ input)

```
∫Abs sin X,_
                R
```

[(−)] [SHIFT] [π] [,] [SHIFT] [π] [)]
($a$, $b$ input)

```
∫sin X,−π,π)_
                R
```

Answer displayed in approximately 20 seconds

[EXE] (Calculation executed)

```
∫(Abs sin X,
             4.
                R
```

[RCL] [G]

```
G=
 −3.141592654
                R
```

$a$

[RCL] [H]

```
H=
  3.141592654
                R
```

$b$

| | | |
|---|---|---|
| RCL I | I= 64. R | N |
| RCL J | J= 4. R | $\int_a^b f(x)dx$ |
| RCL K | K= 0. R | $f(a)$ |
| RCL L | L= 0. R | $f(b)$ |

## ■Notes on integration calculation

- Press AC key during integration calculation (when display is blank) to abort calculation.
- Integration of trigonometric functions carried out in "R" mode (MODE 5).
- This unit utilizes Simpson's rule for integration calculation. As number of significant digits is increased, extended calculation time is required. In some cases, calculation results may be erroneous even after considerable time expires in calculation. In particular, when significant digits are less than 1, an ERROR (Ma ERROR) sometimes occurs.
  In these cases, use the following methods to shorten calculation time and improve accuracy:

1. If integration value varies widely with slight changes in the integration range, divide integration areas to obtain solutions individually.
2. If some periodic functions or integration values are positive and some are negative, divide according to periods or divide positive and negative values and calculate individually.

# Section 4 Program Calculation

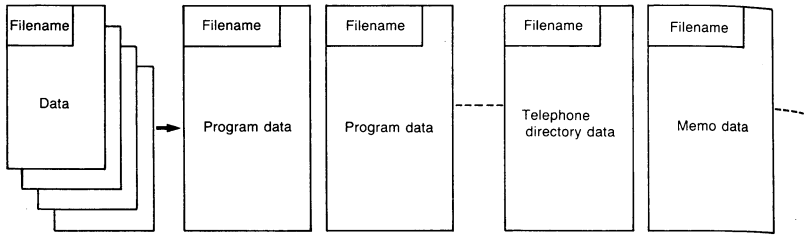

What is a program?
Program correction, addition and deletion
Program searches
Program execution
Convenient program commands
Remaining program capacity
Using the unit as a data bank

# Section 4 Program Calculation

## What is a program?

This unit has a built-in program function that facilitates repeat calculations. As with the "multistatement function", the program function lets you execute series calculations in a series. In addition, the programs entered using this program function are stored in memory as individual files, under filenames. This system lets you search for and edit programs quickly and easily. Any number of programs can be written, provided the total program memory capacity of 1103 steps is not exceeded.

In place of programs, this memory can also be used to store telephone numbers or memos, much in the same way Casio Data Bank entries are made. Each line may contain up to 127 steps.

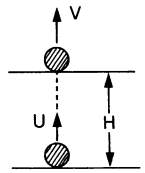



## Programming

The following are practical examples of how the program function can be used.

**Example 1** **Entering formulas**

**(Problem 1)** An object is thrown straight up at an initial velocity of 50 meters per second. How fast will it be travelling after 1 second, and how high will it be? After 3 seconds? After 5 seconds?




| Time (T) | Velocity (V) | Height (H) |
|---|---|---|
| 1 second | (  )m/second | (  )m |
| 3 seconds | (  )m/second | (  )m |
| 5 seconds | (  )m/second | (  )m |

## Calculation formula

The following formulas are used to calculate velocity "V" after "T" seconds, as well as height "H" after "T" seconds, with "U" representing initial velocity, "T" representing time, and "G" representing gravitational acceleration.

$$V = U - GT,\ H = UT - \frac{1}{2}GT^2$$

## Registering filenames

To carry out filename registration, press [MODE][EXP] to specify the WRT mode. The "Filename?" prompt then appears on the display. After the filename is input, press the [EXE] key to register it to memory.

[MODE][EXP]

```
Filename?
F1
  WRT      D
  File No.
```

*Up to 127 steps can be used to store filenames. In addition to the actual number of steps in a name, 2 steps are used each time a filename is registered.

**Example** **Input the filename "GOING UP":**

[MODE][EXP]

```
Filename?
F1
  WRT      D
```

[SHIFT][ALPHA][G][O][I][N][G][SPACE][U][P]

```
GOING UP_
F1
  A WRT    D
```

[EXE]

```
GOING UP
F1
  WRT      D
```

*The calculation mode specified at the time a filename is registered is also held in memory. (Programmed calculation is carried out in mode specified at this time.)

## ■Writing programs

When the filename is displayed in the WRT mode, press the ▼ key (or ▲ key) to display the first line of the program, and begin writing the program.

| | |
|---|---|
| | GOING UP<br>F1<br>WRT D |
| ▼ (or ▲) | _<br>F1 L1<br>WRT D |

Indicates program line number

When you've finished inputting the first line, press the EXE key to register to first line.

**Example** **Input [V = U − GT ◢ ] as the first line in the program:**

| | |
|---|---|
| | _<br>F1 L1<br>WRT D |
| ALPHA V 2ndF = ALPHA U − SHIFT ALPHA G T 2ndF ◢ | V = U − GT ◢ _<br>F1 L1<br>WRT D |
| EXE | V = U − GT ◢<br>F1 L1<br>WRT D |

If you want to input a second line, programming automatically moves to the second line.

**Example** **Input [$H = UT - \frac{1}{2}GT^2$] as the second line in the program:**

| | |
|---|---|
| | V = U − GT ◢<br>F1 L1<br>WRT D |
| ALPHA H | H_<br>F1 L2<br>WRT D |
| 2ndF = SHIFT ALPHA U T ALPHA − ( 1 ÷ 2 ) SHIFT ALPHA G T SHIFT $x^2$ | ← UT − (1/2)GT²_<br>F1 L2<br>WRT D |
| EXE | H = UT − (1/2)GT ←<br>F1 L2<br>WRT D |

*A maximum of 127 steps can be input into a single line. In addition to the actual number

## ■Program execution

Press MODE EXP to cancel the WRT mode, then press the FILE key to call up the filename, and press the EXE key to execute the program.

| | |
|---|---|
| MODE EXP | _<br>D |
| SHIFT Mcl EXE | Mcl<br>0.<br>D |
| FILE | GOING UP<br>F1<br>FILE D |
| EXE | U?<br>0.<br>D |
| 50 EXE | G?<br>0.<br>D |
| 9.8 EXE | T?<br>0.<br>D |
| 1 EXE | V = U − GT<br>40.2<br>Disp D |
| EXE | H = UT − (1/2)GT ←<br>45.1<br>D |
| EXE | U?<br>50.<br>D |
| EXE | G?<br>9.8<br>D |
| EXE | T?<br>1.<br>D |
| 3 EXE | V = U − GT<br>20.6 |

| | |
|---|---|
| [EXE] | H=UT−(1/2)GT → 105.9 D |
| [EXE] | U? 50. D |
| [EXE] | G? 9.8 D |
| [EXE] | T? 3. D |
| 5 [EXE] | V=U−GT 1. Disp D |
| [EXE] | H=UT−(1/2)GT → 127.5 D |

## Program correction, addition and deletion

### Inserting lines

To insert a line between two existing lines, display the line just before the place you want to insert the new line. After inputting the desired program, press the [EXE] key.

### Inserting a line at the beginning

To insert a line at the beginning, display first line by pressing the [⇧] key and then press the [⇧] key once again. After inputting the desired program, press the [EXE] key.

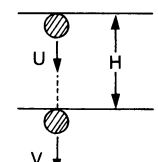
**Example** **Insert [S=GT], a program to determine the displacement velocity in the problem previously presented:**

| | |
|---|---|
| | V=U−GT ◢ F1 L1 WRT D |
| [⇧] | _ F1 L1 WRT D |
| [ALPHA] [S] [2ndF] [=] [SHIFT] [ALPHA] [G] [T] [2ndF] [◢] | S=GT ◢ _ F1 L1 WRT D |
| [EXE] | S=GT ◢ F1 L1 WRT D |
| [⇩] | V=U−GT ◢ F1 L2 WRT D |

### Editing programs

To understand how editing is carried out, work through the following exercise.

**(Problem 2)** An object is dropped at an initial velocity of 50 meters per second. How fast will it be travelling after 1 second, and how far will it have travelled? After 3 seconds? After 5 seconds?




| Time (T) | Velocity (V) | Distance (H) |
|---|---|---|
| 1 second | ( )m/second | ( )m |
| 3 seconds | ( )m/second | ( )m |
| 5 seconds | ( )m/second | ( )m |

## • Registering filename

As the object is dropped in this problem (instead of being thrown up, as in Problem 1 presented previously), input "COMING DOWN" as the filename.

## • Calculation formula

The following formulas are used to calculate the velocity "V" after "T" seconds, as well as the distance travelled "H" after "T" seconds, with "U" representing the initial velocity when the object is dropped, "T" representing time, and "G" representing gravitational acceleration.

$$V = U + GT,\ H = UT + \frac{1}{2}GT^2$$

## • Programming

As with Problem 1 presented previously, programming is carried out in a manner similar to manual calculation:

Speed: [ALPHA][V][2ndF][=][ALPHA][U][+][SHIFT][ALPHA][G][T] ..... V

Distance: [ALPHA][H][2ndF][=][SHIFT][ALPHA][U][T][ALPHA][+][(][1][/][2][)]
[SHIFT][ALPHA][G][T][SHIFT][$x^2$] ..... H

When programmed, this formula is input as follows:

[ALPHA][V][2ndF][=][ALPHA][U][+][SHIFT][ALPHA][G][T][2ndF][◢]
[ALPHA][H][2ndF][=][SHIFT][ALPHA][U][T][ALPHA][+][(][1][/][2][)]
[SHIFT][ALPHA][G][T][SHIFT][ALPHA]

As this program is similar to that used in Problem 1, we can simply "edit" the program already input.

## • Editing filenames

When in the RUN mode, press [MODE][EXP] to specify the WRT mode. Then press the [FILE] key to display the desired filename. Next press the [⇨] key (or [⇦] key) to move the cursor to the beginning (or end) of the filename ([EDIT] symbol appears on display). After editing the filename, press the [EXE] key to register the new (altered) filename.

**Example** **Change "GOING UP" to "COMING DOWN":**

[MODE][EXP]

```
Filename?
F2
    WRT    D
```

[FILE]

```
GOING UP
F1
    WRT    D
```

[⇨]

```
GOING UP
F1
    WRT EDIT   D
```

[SHIFT][ALPHA][C][O][M][I][N][G][SPACE][D][O][W][N]

```
COMING DOWN_
F1
  A WRT EDIT   D
```

[EXE]

```
COMING DOWN
F1
    WRT    D
```

*Note that edited filenames are not registered in memory until [EXE] key is pressed. If [EXE] key has not been pressed, previously registered name will still be held in memory.

## • Program editing

Press [MODE][EXP] to specify the WRT mode. Then press the [FILE] key to display the desired filename. Next press the [⇩] key (or [⇧] key) to move to the beginning (or end) of the line you want to edit. Then use the [⇨] and [⇦] keys to move to the exact point you want to edit. While in the editing mode, the [EDIT] symbol appears on the display. After editing the program, press the [EXE] key to register the edited program.

**Example** **Change the Problem 1 program, which determines speed and height, to the Problem 2 program, which determines speed and distance:**

[MODE][EXP]

```
Filename?
F2
    WRT    D
```

[FILE]

```
COMING DOWN
F1
    WRT    D
```

[⇩][⇩]

```
V=U−GT◢
F1  L2
    WRT    D
```

[⇨]

```
V=U−GT◢
F1  L2
    WRT EDIT   D
```

[⇨][⇨][⇨][+]

```
V=U+GT◢
F1  L2
    WRT EDIT   D
```

[EXE]

```
V=U+GT◢
F1  L2
    WRT    D
```

| | |
|---|---|
| [⇩] | H=UT−(1/2)GT → F1 L3 WRT D |
| [⇨] | H=UT−(1/2)GT → F1 L3 WRT EDIT D |
| [⇨][⇨][⇨][⇨][+] | H=UT+(1/2)GT → F1 L3 WRT D |
| [EXE] | H=UT+(1/2)GT → F1 L3 WRT D |

*Note that edited programs are not registered in memory until [EXE] key is pressed. If [EXE] key has not been pressed, the previously registered program will still be held in memory.

| | |
|---|---|
| [MODE][EXP] | – D |
| [FILE] | COMING DOWN F1 FILE D |
| [EXE] | G? 9.8 D |
| [EXE] | T? 5. D |
| 1 [EXE] | S=GT 9.8 Disp D |
| [EXE] | U? 50. D |
| [EXE] | V=U+GT 59.8 Disp D |
| [EXE] | H=UT+(1/2)GT → 54.9 D |
| [EXE] | G? 9.8 D |
| [EXE] | T? 1. D |
| 3 [EXE] | S=GT 29.4 Disp D |
| [EXE] | U? 50. D |
| [EXE] | V=U+GT 79.4 Disp D |
| [EXE] | H=UT+(1/2)GT → 194.1 D |
| [EXE] | G? 9.8 D |
| [EXE] | T? 3. D |
| 5 [EXE] | S=GT 49. Disp D |
| [EXE] | U? 50. D |
| [EXE] | V=U+GT 99. Disp D |
| [EXE] | H=UT+(1/2)GT → 372.5 D |

## ■Program deletion

Press [MODE][EXP] to specify the WRT mode. As with program editing, display the line containing the program you want to delete and set the unit for editing ([EDIT] symbol appears on display). Next press the [AC] key or [DEL] key and then the [EXE] key. The displayed line will be deleted.

**Example** **Delete the [S=GT] line from the "COMING DOWN" program:**

| | |
|---|---|
| [MODE][EXP] | Filename?<br>F2<br>WRT D |
| [FILE] | COMING DOWN<br>F1<br>WRT D |
| [↓] | S=GT◢<br>F1 L1<br>WRT D |
| [→] | S=GT◢<br>F1 L1<br>WRT EDIT D |
| [AC] | _<br>F1 L1<br>WRT EDIT D |
| [EXE] | V=U+GT◢<br>F1 L1<br>WRT D |

*Note that after a line is deleted, the succeeding line is displayed. When the last line is deleted, however, the line before it is displayed.

### ● Deleting filenames (programs)

Press [MODE][EXP] to specify the WRT mode. Press the [FILE] key to display the filename you want to delete and set the unit for editing ([EDIT] symbol appears on display). Next press the [AC] key or [DEL] key and then the [EXE] key. The displayed filename (and program) will be deleted.

**Example** **Delete the [PROGRAM] file from the list of files to the right:**

| | |
|---|---|
| F1 | GOING UP |
| F2 | COMING DOWN |
| F3 | PROGRAM |
| F4 | FORMULA |

| | |
|---|---|
| [MODE][EXP] | Filename?<br>F5<br>WRT D |
| [FILE][FILE][FILE] | PROGRAM<br>F3<br>WRT D |
| [→] | PROGRAM<br>F3<br>WRT D |
| [AC] | _<br>F3<br>WRT EDIT D |
| [EXE] | FORMULA<br>F3<br>WRT D |

*Note that after a filename is deleted, the succeeding filename is displayed. When the last line is deleted, however, the filename listed before it is displayed.

## ■ Program searches ■

This unit features a program search function which allows you to search for filenames and program lines using either of the following methods:

1. Sequential search (search according to numerical order from beginning)
2. Direct search (search of all filenames or program lines which match input specifications)

## ■Filename sequential search

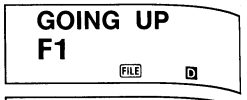
**a. In the RUN mode:**

When in the WRT mode, press [MODE][EXP] to specify the RUN mode. Then press the [FILE] key. File number 1 (F1) will be called up. Each time you press the [FILE] key, the file number will be incremented, and the succeeding filename will be called up. To go back to previous filenames, press [SHIFT][FILE].

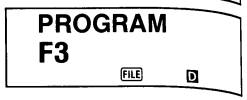
**Example** **Search for the [PROGRAM] file from the filenames listed at the right:**

| | |
|---|---|
| F1 | GOING UP |
| F2 | COMING DOWN |
| F3 | PROGRAM |
| F4 | FORMULA |

AC FILE




FILE FILE



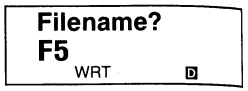

*Note that if you press the FILE key when the last filename is displayed, the last filename remains displayed. Also, if you press SHIFT FILE when the first filename is displayed, it remains displayed.
*You can specify the RUN mode after the display has been cleared (by pressing the AC key), during display of a calculation result, during input of a value for a variable, or while text is displayed.
*If you press the AC key while a filename is displayed, filename display disappears and unit enters "Manual calculations" status.

**b. In the WRT mode:**

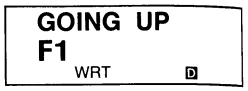
Press MODE EXP to specify the WRT mode. At this time, the "Filename ?" prompt appears on the display. Press the FILE key to display filenames sequentially.
If you press the FILE key when the last filename is displayed, filename input becomes possible, and the display returns to the first filename. Press SHIFT FILE to display names in reverse order.

**Example** **Search for the [PROGRAM] file:**

MODE EXP



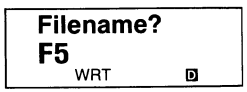

FILE




FILE FILE

*Note that if you press SHIFT FILE when the first filename is displayed, it remains displayed.
*If you press the AC key while a filename is displayed, "Filename?" prompt appears on the display.

AC



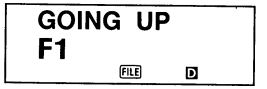

## Sequential search of program lines

**a. In the RUN mode:**

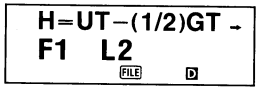
When in the WRT mode, press MODE EXP to specify the RUN mode. Then call up the filename of the program you want to search. Press the ⇩ key to scroll down through program lines. Press the ⇧ to scroll up through program lines.

**Example** **In File 1, we programmed the "GOING UP" program. Search for the 2nd line in this program:**

AC FILE




⇩⇩



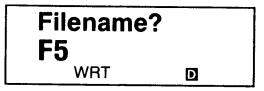

*Note that if you press the ⇩ key when the last program line is displayed, the last program line remains displayed. Also, if you press ⇧ when the first program line is displayed, it remains displayed.
*If AC is pressed when program is displayed, file display disappears and unit enters "Manual calculations" status.

**b. In the WRT mode:**

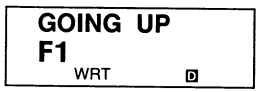
Press MODE EXP to specify the WRT mode. Then call up the filename containing the program you want to search. Press the ⇩ key to scroll down through program lines. If you press the ⇩ key when the last filename is displayed, input of additional lines becomes possible. Press the ⇧ key to scroll up through program lines. If you press the ⇧ key when the first program line is displayed, it becomes possible to insert additional lines at the first of the file.

**Example** **As in the previous example, search for the 2nd line in the "GOING UP" program:**

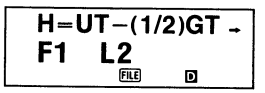
MODE EXP



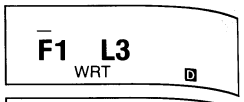

FILE



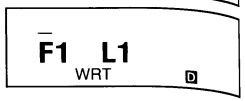

⇩⇩



[⇩]

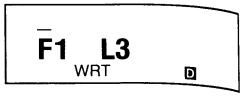



[⇧][⇧][⇧]




*If [AC] is pressed when program is displayed, the last line is automatically displayed and input of new lines becomes possible.

[AC]




## ■Direct search

### • Direct search of filename

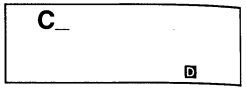
By inputting the first character or characters of a filename (up to 127 steps) and pressing the [FILE] key, it is possible to search for an individual filename directly.

**•In the RUN mode:**

**Example** **Search "COMING DOWN" from the filenames at the right:**

| | |
|---|---|
| F1 | GOING UP |
| F2 | COMING DOWN |
| F3 | PROGRAM |
| F4 | FORMULA |

[AC][ALPHA][C]



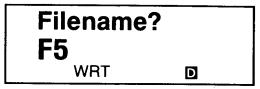

[FILE]

*Character input for search must be carried out in "Manual calculations" status or when filenames are displayed.
*If the filename cannot be found, operation returns to "Manual calculations" status.

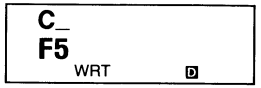
**•In the WRT mode:**

**Example** **Search for "COMING DOWN":**

[MODE][EXP]




[ALPHA][C]




[FILE]

*Character input for search must be carried out while the "Filename?" prompt is displayed or when filenames are displayed.
*If the filename cannot be found, the "Filename?" prompt appears on the display.

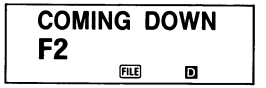
### • Direct search of program lines

By inputting the first character or characters of a line (up to 127 steps) when the corresponding filename is displayed and pressing the [⇩] key (or [⇧] key), it is possible to search for an individual program line directly. The same procedure is used in both the RUN mode and WRT mode.

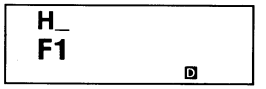
**•In the RUN mode:**

**Example** **In File 2, we programmed the "COMING DOWN" program. Search for the 2nd line in this program:**

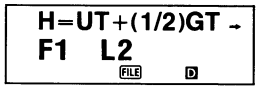
[AC][FILE][FILE]




[ALPHA][H]




[⇩]




*Press the [⇩] key repeatedly to continue direct search. After initial direct search, press the [⇧] key to abort search.
*If the specified line is not found, the last line is automatically displayed and input of new lines becomes possible. If no program has been entered, input becomes possible from the first line.

**Notes on Direct search function**

- The "℃" in $_nC_r$ and "℃" used in the BASE-N mode cannot be searched simultaneously. The same is true for the P of "Pol (" and the letter "P".

## ■Scrolling right and left in filenames and program lines

When filenames or program lines contain more than 12 characters, use the ⇦ and ⇨ keys to scroll to the right or left.

*In the WRT mode, the cursor flashes allowing editing of the filename or program.

**Example** **Check contents of 2nd line of program below:**

| F1 | HELON |
|---|---|
| L1 | L=(A+B+C)/2◢ |
| L2 | S=√(L(L−A) (L−B) (L−C)) |

| | |
|---|---|
| AC FILE | HELON<br>F1 |
| ⇩ | L=(A+B+C)/2◢ ←<br>F1 L1 |
| ⇩ | S=√(L(L−A)(L ←<br>F1 L2 |
| ⇨ | =√(L(L−A)(L− ←<br>F1 L2 |
| ⇨⇨⇨⇨⇨⇨⇨⇨⇨⇨ | ←(L−B)(L−C))<br>F1 L2 |
| ⇦ | ←)(L−B)(L−C)) →<br>F1 L2 |

# Program execution

Programs can be executed in two different ways:

## ■Execution through filename search

After specifying the RUN mode, press the FILE key. The first filename (F1) is displayed. Search the desired filename and press the EXE key to execute the program.

**Example** **Execute the "GOING UP" program:**

| | |
|---|---|
| AC SHIFT Mcl EXE | Mcl<br>0. |
| FILE | GOING UP<br>F1 |
| EXE | U?<br>0. |
| 50 EXE | G?<br>0. |
| 9.8 EXE | T?<br>0. |
| 1 EXE | V=U−GT<br>40.2 |
| EXE | H=UT−(1/2)GT ←<br>45.1 |
| EXE | U?<br>50. |

(Continuation omitted)

## ■Execution by pressing SHIFT Prog

After pressing SHIFT Prog, input the filename and press the EXE key to execute the program.

**Example** **Execute the "COMING DOWN" program:**

| | |
|---|---|
| AC SHIFT Mcl EXE | Mcl<br>0. |
| SHIFT Prog | Prog _ |
| SHIFT ALPHA C O M I N G SPACE<br>D O W N | ← COMING DOWN_ |
| EXE | U?<br>0. |
| 50 EXE | G?<br>0. |
| 9.8 EXE | T?<br>0. |
| 1 EXE | V=U+GT<br>59.8 |
| EXE | 54.9 |
| EXE | U?<br>50. |

(Continuation omitted)

To have the final formula in a program (executed by SHIFT Prog) remain on the upper display, include a "◢" as the last command in the program, after the final formula.

*By inputting "Prog" and then the filename, the filename can be used as a subroutine in the program (see page 114 for details).

## ■Aborting execution

Press the FILE key to abort execution while a program is being executed. The first filename then appears on the display. By pressing the AC key in place of the FILE key, execution is aborted and operation returns to the "Manual calculations" status.

## ■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 the process of correcting them is called "debugging".

When an error message is displayed, press the ◁ or ▷ key to move the cursor to the place where the error is generated (see page 49), and correct the program. For details, see the error message table on page 154.
When an incorrect or unusual result is generated, press MODE EXP to enter the WRT mode, then press the FILE key to display the filename corresponding to the program you want to correct. (See page 97 for details on program editing.)

# ■ Convenient program commands ■

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

## ■Jump commands

Jump commands are used to change the flow of program execution. Programs are executed in the order that they are input 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 two types of jump commands: a simple unconditional jump to a branch destination and a conditional jump that decides the branch destination by determining whether a certain condition is true or not.

### • Unconditional jump

The unconditional jump is composed of "Goto" and "Lbl". When program execution reaches the statement "Goto" and a label name, execution then jumps to the same "Lbl" [label] label name listed with the "Goto" command. 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.

*Label names can contain alphabetic characters, numbers, functional commands (sin, cos, etc.), etc., however they may not contain delimiter codes (:, ◢, ⇒, ⇌, ▵, etc.)
*Label names may be comprised of up to 126 steps.

**Example** **Rewrite the program used in Problem 1 using the "Goto 0" and "Lbl 0" commands to allow repeat calculations:**

The program used in Problem 1 (presented previously) is shown to the right. Add "Goto 0" to the end of the program, and add "Lbl 0" to the beginning of the program which is the branch destination. If this is simply left the way it is, however, the height will not be displayed and only the initial velocity will be displayed. To prevent this, insert the display command "◢" at the end of the formula that calculates height H.

| F1 | GOING UP |
|---|---|
| L1 | V=U−GT◢ |
| L2 | H=UT−(1/2)GT² |

The program is still not complete, because after the first execution, the unit will retain the first value of T that you enter, so repeat calculation using different values for T will not be possible (values for U and G are fixed). Here, we will add the variable input command "{T}" (see page 118) to tell the program to prompt for a new input for variable T each time the program is executed.

| F1 | GOING UP |
|---|---|
| L1 | Lbl 0 |
| L2 | {T} |
| L3 | V=U−GT◢ |
| L4 | H=UT−(1/2)GT²◢ |
| L5 | Goto 0 |

With this, we will execute the program:

AC FILE

```
GOING UP
F1
                FILE  D
```

EXE

```
U?
                    0.
                    D
```

50 EXE

```
G?
                    0.
                    D
```

9.8 EXE

```
T?
                    0.
                    D
```

1 EXE

```
V=U−GT
                   40.2
              Disp  D
```

EXE

```
H=UT−(1/2)GT
                   45.1
              Disp  D
```

EXE

```
T?
                    1
                    D
```

3 EXE

```
V=U−GT
                   20.6
              Disp  D
```

EXE

```
H=UT−(1/2)GT
                  105.9
              Disp  D
```

EXE

```
T?
                    3.
                    D
```

5 EXE

```
V=U−GT
                    1.
              Disp  D
```

EXE

```
H=UT−(1/2)GT
                  127.5
              Disp  D
```

In this way, an unconditional jump is made in accordance with "Goto" and "Lbl", and the flow of program operation is changed. When there is no "Lbl [label name]" to correspond to the "Goto [label name]" command, an error (Go ERROR) appears on the display.

## • Conditional jump

The conditional jump compares a numeric value with a constant or another numeric value in memory. If the condition is true, the statement following "=>" is executed up to the next "≠" or "◺". If the condition is not true, execution skips the statement following "≠" up to the next "◺". In either case, execution continues from the above following the jump end code "◺".

Conditional jumps are formed in the following ways:

1. Left side Relational operator Right side => Statement { : ◢ } ≠ Statement { : ◢ } ◺ Statement

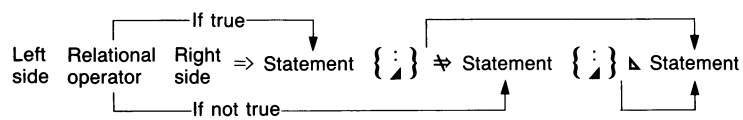
2. Left side Relational operator Right side => Statement { : ◢ } ◺ Statement

One variable (A ~ Z), constant or variable formula (A×2, D−E, etc.) is used for the left side and one for the right side.

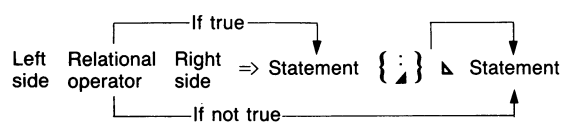
The relational operator is a comparison symbol. There are 6 types of relational operators: =, ≠, ≧, ≦, >, <.

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 greater than or equal to right side)
Left side ≦ right side (left side less than or equal to right side)
Left side > right side (left side greater than right side)
Left side < right side (left side less than right side)

The "=>" is displayed when [2ndF][=>] are pressed. If the condition is true, execution advances to the next statement. The "≠" is displayed when [2ndF][≠>] are pressed. If the condition is not true, execution advances to the next statement. After this, if [2ndF][◺] are pressed, execution continues from the next "◺".




*If statement following "≠" is unnecessary, flow of operations is as follows:




*For statements following "=>" and "≠", multistatements can also be used.

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

Program must be written as follows.
In the following program, A representing input numeric value, B representing calculation result.

```
Lbl 0
{A}
  ┌If true┐
A≧0 => B=√A ◢  ≠  B=A² ◢
  └If not true┘
◺Goto 0
```

Unconditional jump

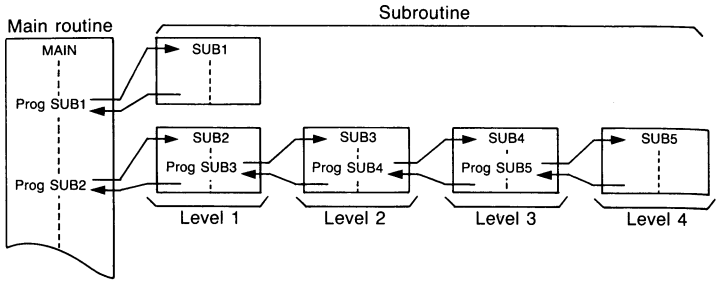
In this formula, a value is input for variable A. If this value is equal to or greater than zero, the statement between "=>" and "◢" is executed. If it is less than zero, the statement between "≠" and "◢" is executed. When "Goto 0" is reached, execution returns to "Lbl 0", for repeated calculation.

Input the filename [VALUE] and execute the program:

| | |
|---|---|
| [AC][ALPHA][V][FILE] | VALUE<br>F5<br>[FILE] [D] |
| [EXE] | A?<br>0.<br>[D] |
| 2[EXE] | B=√A<br>1.414213562<br>[Disp] [D] |
| [EXE] | A?<br>2.<br>[D] |
| 0[EXE] | B=√A<br>0.<br>[Disp] [D] |
| [EXE] | A?<br>0.<br>[D] |
| −2[EXE] | B=A²<br>4.<br>[Disp] [D] |

## ■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 filename which is used to specify a program area.

**Example** **Prog ABC — Jump to program area [ABC]:**

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 name 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 9 levels, and attempts to exceed this level will cause an error (Ne ERROR) to be generated. Attempting to use Prog to jump to a filename in which there is no program stored will also result in an error (Go ERROR).

**Example** **Simultaneously execute the two previously presented programs (Problem 1 and Problem 2) to determine the speed and height of an object which is tossed straight up, and the speed and distance of an object which is dropped:**

This example employs the two programs previously presented:

| F1 | GOING UP |
|---|---|
| L1 | Lbl 0 |
| L2 | {T} |
| L3 | V=U−GT◢ |
| L4 | H=UT−(1/2)GT²◢ |
| L5 | Goto 0 |

| F2 | COMING DOWN |
|---|---|
| L1 | Lbl 0 |
| L2 | {T} |
| L3 | V=U+GT◢ |
| L4 | H=UT+(1/2)GT²◢ |
| L5 | Goto 0 |

When these two programs are compared, it is evident that lines 1, 2 and 5 are identical. If these portions are incorporated into a common routine, the programs are simplified and the number of steps required is decreased. Also, if it is possible to choose which calculation is going to be carried out when this program is executed, the calculation becomes even easier. For this, the conditional jump command is used in the main routine in the following way:

| F3 | MAIN |
|---|---|
| L1 | Lbl 0 |
| L2 | {T, N} |
| L3 | N ''GOING UP:0, COMING DOWN:1'' |
| L4 | N=1⇒Prog COMING DOWN: ⇜N=0⇒Prog GOING UP◣◣ |
| L5 | Goto 0 |

The portions of the program which are not identical are as follows:

| F1 | GOING UP |
|---|---|
| L1 | V=U−GT◢ |
| L2 | H=UT−(1/2)GT²◢ |

| F2 | COMING DOWN |
|---|---|
| L1 | V=U+GT◢ |
| L2 | H=UT+(1/2)GT²◢ |

If the program is written in this way, after the program assigned to the filename "MAIN" is executed, zero is input to jump to the "GOING UP" subroutine to calculate the velocity and height of the object tossed in the air, with an unconditional jump from "Goto 0" to "Lbl 0". If a value of 1 is input for N, execution jumps to the "COMING DOWN" subroutine to calculate the velocity and distance of the object which is dropped.

Actual programming and execution:

AC SHIFT Mcl EXE

```
Mcl
                0.
                D
```

ALPHA M FILE

```
MAIN
F3
          FILE  D
```

EXE

```
GOING UP:0,? ←
                0.
                D
```

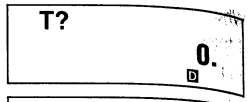
EXE

```
U?
                0.
                D
```

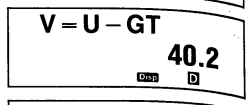
50 EXE

```
G?
                0.
                D
```

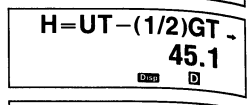
9.8 [EXE]



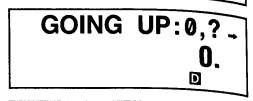

1 [EXE]



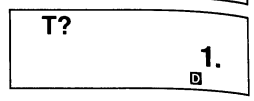

[EXE]



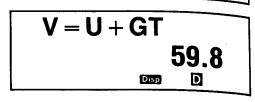

[EXE]



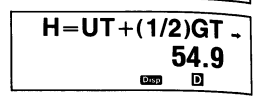

1 [EXE]



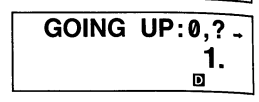

[EXE]




[EXE]




[EXE]




(Continuation omitted)

In this way, subroutines can be used to isolate the common portions of two original programs and store them in separate program areas. Steps are shortened, and programs take on a clearer configuration.

## ■Pause command

By inputting [Pause $n$ ($n=$ an integer between 0 and 9)] in the program, execution can be interrupted (paused) for up to 4.5 seconds.
While in the pause state, the answer from the previous line's calculation and the formula (or text) are displayed.

**Example** **Perform a calculation wherein a value of 1 is added consecutively to variable A. In this case, variable A's initial value is 1:**

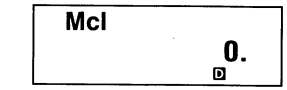
For this case, the program is as follows:

```
Lbl 0
A=A+1
Pause 3 (Displayed for approximately 1.5 seconds)
Goto 0
```

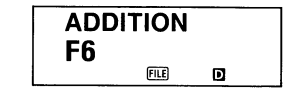
Unconditional jump

When a value is input for variable A, "Pause 3" causes a pause of 1.5 seconds, after which "Goto 0" causes execution to return to "Lbl 0", with the formula [A=A+1] calculated repeatedly.
Here, we will insert the name "ADDITION" and execute the program:

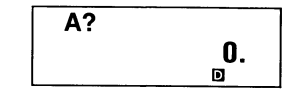
[AC] [SHIFT] [Mcl] [EXE]



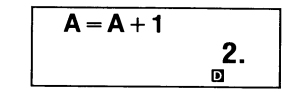

[ALPHA] [A] [FILE]



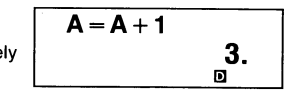

[EXE]




1 [EXE]

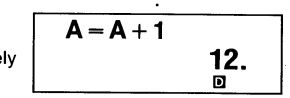



(After approximately 1.5 seconds)

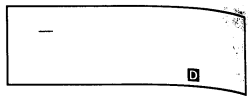



⋮

(After approximately 15 seconds)



AC




The amount of time (approximate) corresponding to "Pause $n$" ("$n$" being an integer between 0 and 9) is as follows:

| $n$ | 0 | 1 | 2 | 3 | 4 | ..... | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| Pause time (seconds) | 0 | 0.5 | 1 | 1.5 | 2 | ..... | 4 | 4.5 |

*"Pause $n$" is treated as a single statement.

## ■Variable input command

When a value is input for a variable in a program, that value is stored in memory as a defined value. If it becomes necessary to input a new value for that variable, the variable input command 2ndF [(], 2ndF [)] can be used to return the variable to its undefined status. This is done by inputting the variable (A~Z) in brackets "{ }".

**Example** {A} .... Variable A returned to undefined status.
{AB} {A, B} {A␣B} .... Variables A and B returned to undefined status.

*"{ }" is treated as a single statement.
*Array variables cannot be used as variables.

## ■Fixm

When 2ndF [Fixm] is input in a program, all values for variables (A~Z) after the command are treated as defined values. When the program is executed, the program does not wait for entry of values for variables, but completes calculation using values which have already been input.

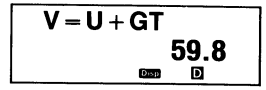
**Example** **Input "Fixm" in the first line of the program written for Problem 2 (see page 95):**

The program written for Problem 2 is as shown at the right. We will assume that the following values have been input: U=50, G=9.8, T=1.

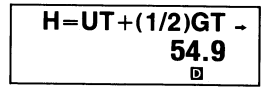
| F2 | COMING DOWN |
|---|---|
| L1 | Fixm |
| L2 | V=U+GT◢ |
| L3 | H=UT+(1/2)GT² |

AC FILE FILE

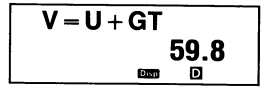
EXE




EXE




EXE



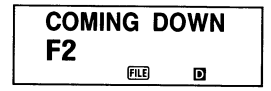

In this case, the calculation was carried out using only the values which had already been entered, so the results did not vary. If the variable input command "{ }" is contained in the same program where "Fixm" is used, the "{ }" command takes priority.

**Example** **Input the "{ } " command into the Problem 2 program which contains "Fixm":**

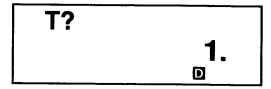
In this program, variables U and G are calculated using the defined values already input. Variable T, however is called up, and a value is input. Here, we will input the value and execute the program:

| F2 | COMING DOWN |
|---|---|
| L1 | Fixm |
| L2 | U=50:G=9.8 |
| L3 | {T} |
| L4 | V=U+GT◢ |
| L5 | H=UT+(1/2)GT² |

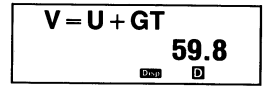
AC FILE FILE



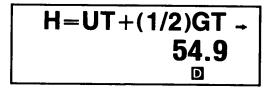

EXE



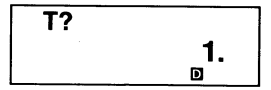

EXE



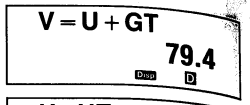

EXE



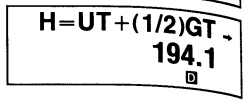

EXE



3 EXE




EXE




(Continuation omitted)

*"Fixm" is treated as a single statement.

## Remaining program capacity

The program capacity of this unit is 1103 steps. The number of steps indicates the amount of storage space available, 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 41.)

*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 "label name", Goto "label name", Prog "filename", etc.

*When the step capacity is exceeded, a "Mem ERROR" is generated.

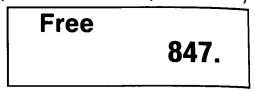
### Determining the number of remaining steps

Hold down SHIFT CAPA to display the current remaining number of steps. Display returns to normal when keys are released.

**Example**

SHIFT CAPA

(Indicates 847 steps available)




## Using the unit as a data bank

In place of data or programs, it is possible to store often used formulas or even telephone numbers in this unit, using it much like a data bank.
Here, we will input a list of telephone numbers.

**Filename: TEL DATA**

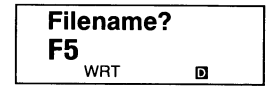
**Data:**

| Listing No. | Numbers |
|---|---|
| 1 | Robert Jones 03-012-3456 |
| 2 | Samuel Stevens 03-023-4567 |
| 3 | John Smith 0425-034-5678 |
| 4 | Henry White 0425-045-6789 |
| 5 | Jane Bell 0473-056-7890 |

**a. Inputting data**

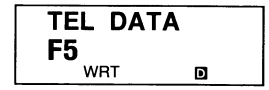
Press MODE EXP to specify WRT mode. Specify this telephone list as "File 5":

MODE EXP



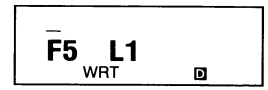

Input the filename:

SHIFT ALPHA T E L SPACE
D A T A EXE




Press ⇩ key to input listing number 1:

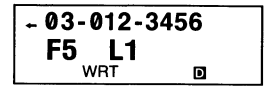
⇩

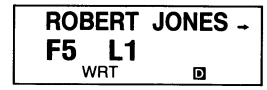



Input data for listing number 1:

SHIFT ALPHA R O B E R T SPACE
J O N E S SPACE ALPHA 0 3 (−)
0 1 2 (−) 3 4 5 6

EXE




Other listings are input in the same way.

**b. Recalling data**

First, call up the filename "TEL DATA". The direct search function can be used as shown below:

[AC] [ALPHA] [T] [FILE]

```
TEL DATA
F5
FILE  D
```

Next, call up the data for "Samuel Stevens":

[ALPHA] [S]

```
S_
F5
D
```

[↓]

```
SAMUEL STEVE→
F5  L2
FILE  D
```

[→]

```
←AMUEL STEVEN→
F5  L2
FILE  D
```

[→]

```
←MUEL STEVENS→
F5  L2
FILE  D
```

Use the [←] and [→] keys to scroll to the left or right:

[→][→][→][→][→][→][→][→][→][→]
[→][→]

```
← 03-023-4567
F5  L2
FILE  D
```

As a space is inserted before entering the telephone number, listings can also be searched according to number:

[AC] [ALPHA] [T] [FILE]

```
TEL DATA
F5
FILE  D
```

Search using the prefix "03":

03[↓]

```
←03-012-3456
F5  L1
FILE  D
```

Press the [↓] key until the listing you're searching for appears:

[↓]

```
←03-023-4567
F5  L2
FILE  D
```

Use the [←] and [→] keys to scroll to the left or right:

[←][←][←][←][←][←][←][←][←][←]
[←][←]

```
UEL STEVENS →
F5  L2
FILE  D
```

# PROGRAM LIBRARY

1. Prime factor analysis
2. Greatest common measure
3. $\Delta \leftrightarrow Y$ transformation
4. Minimum loss matching
5. Cantilever under concentrated load
6. Normal distribution
7. Numerical solution of an equation (Newton's law)
8. Quadratic equation
9. Complex numbers

# 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 ($d=3, 5, 7, 9, 11, 13, .....$) 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〉

$119=7\times17$

〈2〉

$630=2\times3\times3\times5\times7$

〈3〉

$987654321=3\times3\times17\times17\times379721$

## Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below.

| Step | Key operation | Display |
|---|---|---|
| 1 | AC SHIFT Mcl EXE | Mcl 0. |
| 2 | FILE | PRIME FACTOR→ F1 |
| 3 | EXE | M? 0. |
| 4 | 119 EXE | PRIME FACTOR→ 7. |
| 5 | EXE | PRIME FACTOR→ 17. |
| 6 | EXE | END 119. |
| 7 | EXE | M? 17. |
| 8 | 630 EXE | PRIME FACTOR→ 2. |
| 9 | EXE | PRIME FACTOR→ 3. |
| 10 | EXE | PRIME FACTOR→ 3. |
| 11 | EXE | PRIME FACTOR→ 5. |
| 12 | EXE | PRIME FACTOR→ 7. |
| 13 | EXE | END 630. |
| 14 | EXE | M? 7. |
| 15 | 987654321 EXE | PRIME FACTOR→ 3. |
| 16 | EXE | PRIME FACTOR→ 3. |
| 17 | EXE | PRIME FACTOR→ 17. |
| 18 | EXE | PRIME FACTOR→ 17. |
| 19 | EXE (After approx. 1.5 minutes) | PRIME FACTOR→ 379721. |
| 20 | EXE | END 987654321. |

| No. | 1 |
|---|---|

| Line | MODE EXP | Program | | | | | | | | | | | | | | Notes | Number of steps |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | P | R | I | M | E | | F | A | C | T | O | R | | | | | 14 |
| L1 | Lbl | 0 | : | { | A | } | : | A | ” | M | ” | : | N | = | A | | |
| | : | Goto | 2 | ◣ | | | | | | | | | | | | | 34 |
| 2 | Lbl | 1 | : | B | = | 2 | : | ” | P | R | I | M | E | | F | | |
| | A | C | T | O | R | ” | ◢ | A | = | A | / | 2 | : | A | = | | |
| | 1 | ⇒ | Goto | 9 | ◣ | | | | | | | | | | | | 70 |
| 3 | Lbl | 2 | : | Frac | ( | A | / | 2 | ) | = | 0 | ⇒ | Goto | 1 | ◣ | | |
| | B | = | 3 | | | | | | | | | | | | | | 89 |
| 4 | Lbl | 3 | : | C | = | √ | A | + | 1 | | | | | | | | 99 |
| 5 | Lbl | 4 | : | B | ≥ | C | ⇒ | Goto | 8 | ◣ | Frac | ( | A | / | B | | |
| | ) | = | 0 | ⇒ | Goto | 6 | ◣ | | | | | | | | | | 122 |
| 6 | Lbl | 5 | : | B | = | B | + | 2 | : | Goto | 4 | ◣ | | | | | 135 |
| 7 | Lbl | 6 | : | ( | A | / | B | ) | B | − | A | = | 0 | ⇒ | Goto | | |
| | 7 | ◣ | Goto | 5 | | | | | | | | | | | | | 155 |
| 8 | Lbl | 7 | : | B | : | ” | P | R | I | M | E | | F | A | C | | |
| | T | O | R | ” | ◢ | A | = | A | / | B | : | Goto | 3 | ◣ | | | 185 |
| 9 | Lbl | 8 | : | A | : | ” | P | R | I | M | E | | F | A | C | | |
| | T | O | R | ” | ◢ | | | | | | | | | | | | 206 |
| 10 | Lbl | 9 | : | N | : | ” | E | N | D | ” | ◢ | Goto | 0 | | | | 220 |
| 11 | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | |

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 | $m$ | U | | | | |

# CASIO PROGRAM SHEET

| Program for | Greatest common measure | No. | 2 |
|---|---|---|---|

## Description

Euclidean general division is used to determine the greatest common measure for two integers $a$ and $b$.

For $|a|, |b| < 10^9$, positive values are taken as $< 10^{10}$

〈Overview〉

$n_0 = \max(|a|, |b|)$

$n_1 = \min(|a|, |b|)$

$$n_k = n_{k-2} - \left[\frac{n_{k-2}}{n_{k-1}}\right] n_{k-1}$$

$k = 2, 3 \ldots\ldots$

If $n_k = 0$, then the greatest common measure ($c$) will be $n_{k-1}$.

## Example

| When | 〈1〉 | 〈2〉 | 〈3〉 |
|---|---|---|---|
| | $a = 238$ | $a = 23345$ | $a = 522952$ |
| | $b = 374$ | $b = 9135$ | $b = 3208137866$ |
| | ↓ | ↓ | ↓ |
| | $c = 34$ | $c = 1015$ | $c = 998$ |

## Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below.

| Step | Key operation | Display |
|---|---|---|
| 1 | AC SHIFT Mcl EXE | Mcl 0. |
| 2 | FILE | COMMON MEASU → F1 |
| 3 | EXE | A? 0. |
| 4 | 238 EXE | B? 0. |
| 5 | 374 EXE | C 34. |
| 6 | EXE | A? 102. |
| 7 | 23345 EXE | B? 34. |
| 8 | 9135 EXE | C 1015. |
| 9 | EXE | A? 4060. |
| 10 | 522952 EXE | B? 1015. |
| 11 | 3208137866 EXE | C 998. |

No. 2

| Line | Program | Notes | Number of steps |
|---|---|---|---|
| F1 | C O M M O N M E A S U R E | | 16 |
| L1 | Lbl 1 | | 19 |
| 2 | { A , B } | | 25 |
| 3 | A = Abs A : B = Abs B | | 35 |
| 4 | B > A ⇒ C = A : A = B : B = C | | |
| | ◣ | | 52 |
| 5 | Lbl 2 | | 55 |
| 6 | C = − ( Int ( A / B ) × B − A ) | | 71 |
| 7 | C ≠ 0 ⇒ A = B : B = C : Goto 2 ◣ | | 87 |
| 8 | B : " " C " " ◢ Goto 1 | | 96 |
| 9 | | | |
| 10 | | | |
| 11 | | | |
| 12 | | | |
| 13 | | | |
| 14 | | | |
| 15 | | | |
| 16 | | | |
| 17 | | | |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |
| 22 | | | |
| 23 | | | |
| 24 | | | |
| 25 | | | |
| 26 | | | |

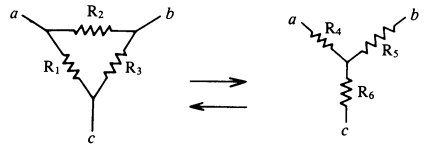
Memory contents

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | $a, n_0$ | H | | O | | V | |
| B | $b, n_1$ | I | | P | | W | |
| C | $n_k$ | J | | Q | | X | |
| D | | K | | R | | Y | |
| E | | L | | S | | Z | |
| F | | M | | T | | | |
| G | | N | | U | | | |

# CASIO PROGRAM SHEET

| Program for | No. |
|---|---|
| Δ↔Y transformation | 3 |

**Description**




1) Δ→Y

$$R_4=\frac{R_1\cdot R_2}{R_1+R_2+R_3}$$
$$R_5=\frac{R_2\cdot R_3}{R_1+R_2+R_3}$$
$$R_6=\frac{R_3\cdot R_1}{R_1+R_2+R_3}$$

2) Y→Δ

$$R_1=\frac{R_4R_5+R_5R_6+R_6R_4}{R_5}$$
$$R_2=\frac{R_4R_5+R_5R_6+R_6R_4}{R_6}$$
$$R_3=\frac{R_4R_5+R_5R_6+R_6R_4}{R_4}$$

**Example**

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

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

**Preparation and operation**

- Store the program written on the next page.
- Execute the program as shown below.

| Step | Key operation | Display | Step | Key operation | Display |
|---|---|---|---|---|---|
| 1 | AC SHIFT Mcl EXE | Mcl 0. | 11 | 2 EXE | R4? 4. |
| 2 | FILE | TRANSFORMATI→ F1 | 12 | 100 EXE | R5? 27.33333333 |
| 3 | EXE | D⇒Y:1,Y⇒D:2? 0. | 13 | 150 EXE | R6? 6.978723404 |
| 4 | 1 EXE | R1? 0. | 14 | 220 EXE | R1= 466.6666667 |
| 5 | 12 EXE | R2? 0. | 15 | EXE | R2= 318.1818182 |
| 6 | 47 EXE | R3? 0. | 16 | EXE | R3= 700. |
| 7 | 82 EXE | R4= 4. | | | |
| 8 | EXE | R5= 27.33333333 | | | |
| 9 | EXE | R6= 6.978723404 | | | |
| 10 | EXE | D⇒Y:1,Y⇒D:2? 1. | | | |

No. 3

| Line | Program | Notes | Number of steps |
|---|---|---|---|
| F1 | T R A N S F O R M A T I O N | | 16 |
| L1 | Lbl 1 | | 19 |
| 2 | { N } : N " D ⇒ Y : 1 , Y ⇒ D | | |
| | : 2 " | | 38 |
| 3 | N = 2 ⇒ Goto 2 : ⇜ N ≠ 1 ⇒ Goto 1 ◣ | | 54 |
| 4 | { A } : A " R 1 " : { B } : B | | |
| | " R 2 " : { C } : C " R 3 " | | 84 |
| 5 | D = A + B + C | | 92 |
| 6 | E " R 4 " = A B ∕ D ◢ | | 104 |
| 7 | F " R 5 " = B C ∕ D ◢ | | 116 |
| 8 | G " R 6 " = C A ∕ D ◢ | | 128 |
| 9 | Goto 1 | | 131 |
| 10 | Lbl 2 | | 134 |
| 11 | { E } : E " R 4 " : { F } : F | | |
| | " R 5 " : { G } : G " R 6 " | | 164 |
| 12 | H = E F + F G + G E | | 175 |
| 13 | A " R 1 " = H ∕ F ◢ | | 186 |
| 14 | B " R 2 " = H ∕ G ◢ | | 197 |
| 15 | C " R 3 " = H ∕ E ◢ | | 208 |
| 16 | Goto 1 | | 211 |
| 17 | | | |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |
| 22 | | | |
| 23 | | | |
| 24 | | | |

Memory contents

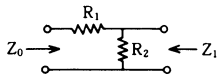
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | $R_1$ | H | $R_4R_5+R_5R_6+R_6R_4$ | O | | V | |
| B | $R_2$ | I | | P | | W | |
| C | $R_3$ | J | | Q | | X | |
| D | $R_1+R_2+R_3$ | K | | R | | Y | |
| E | $R_4$ | L | | S | | Z | |
| F | $R_5$ | M | | T | | | |
| G | | N | For judgement |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

| Step | Key operation | Display | Step | Key operation | Display |
|---|---|---|---|---|---|
| 1 | AC SHIFT Mcl EXE | Mcl<br>0. | | | |
| 2 | FILE | LOSS MATCHIN →<br>F1 | | | |
| 3 | EXE | Z0 ?<br>0. | | | |
| 4 | 500 EXE | Z1?<br>0. | | | |
| 5 | 200 EXE | R1 =<br>387.2983346 | | | |
| 6 | EXE | R2 =<br>258.1988897 | | | |
| 7 | EXE | LMIN =<br>8.961393328 | | | |
| | | | | | |
| | | | | | |
| | | | | | |

No. 4

| Line | Program | Notes | Number of steps |
|---|---|---|---|
| F1 | LOSS" Z0" :Z" Z1" | | 15 |
| L1 | Y" Z0/Z1" :B=Y/ | | 27 |
| 2 | A=√(1−Z/Y):B=Y/ | | |
| | Z | | 44 |
| 3 | R" R1" =YA◢ | | 54 |
| 4 | S" R2" =ZA◢ | | 65 |
| 5 | T" LMIN" =20log(√B+ | | |
| | √(B−1))◢ | | 89 |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |
| 11 | | | |
| 12 | | | |
| 13 | | | |
| 14 | | | |
| 15 | | | |
| 16 | | | |
| 17 | | | |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |
| 22 | | | |
| 23 | | | |
| 24 | | | |
| 25 | | | |

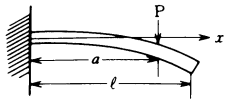
Memory contents

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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/cm²]
I : Geometrical moment of inertia [cm⁴]
$a$ : Distance of concentrated load from support [cm]
P : Load [kg]
$x$ : Distance of point of interest from the support [cm]

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

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 = 2,100,000 kg/cm²
I = 700 cm⁴
$a$ = 50 cm
P = 100 kg

What are deflection, angle of deflection, bending moment and shearing load at $x$ = 40 cm and $x$ = 60 cm?

## Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below.

| Step | Key operation | Display |
|---|---|---|
| 1 | AC SHIFT Mcl EXE | Mcl<br>0. |
| 2 | FILE | CANTILEVER →<br>F1 |
| 3 | EXE | E?<br>0. |
| 4 | 2100000 EXE | I?<br>0. |
| 5 | 700 EXE | A?<br>0. |
| 6 | 50 EXE | P?<br>0. |
| 7 | 100 EXE | X?<br>0. |
| 8 | 40 EXE | Y=PX²/2EI×(X →<br>−0.001995464 |
| 9 | EXE | S=tan⁻¹(PX/2 →<br>−0.004677206 |
| 10 | EXE | M=P(X−A)<br>−1000. |
| 11 | EXE | X?<br>40. |
| 12 | 60 EXE | Y=PA²/2EI×(A →<br>−0.003684807 |
| 13 | EXE | S=tan⁻¹(−PA² →<br>−0.00487209 |
| 14 | EXE | M=∅<br>0. |

| No. | 5 |
|---|---|

| Line | Program | Notes | Number of steps |
|---|---|---|---|
| F1 | C A N T I L E V E R | | 12 |
| L1 | Deg | | 14 |
| 2 | Lbl 1 | | 17 |
| 3 | E : I : A : P : { X } | | 29 |
| 4 | X ≦ A ⇒ Goto 2 ◣ | | 37 |
| 5 | Y = P A $x^2$ / 2 E I × ( A / 3 − X ) ◢ | | 56 |
| 6 | S = $\tan^{-1}$ ( − P A $x^2$ / 2 E I ) ◢ | | 71 |
| 7 | M = ∅ ◢ | | 76 |
| 8 | Goto 1 | | 79 |
| 9 | Lbl 2 | | 82 |
| 10 | Y = P X $x^2$ / 2 E I × ( X / 3 − A ) ◢ | | 101 |
| 11 | S = $\tan^{-1}$ ( P X / 2 E I × ( X − 2 A ) ◢ | | 121 |
| 12 | M = P ( X − A ) ◢ | | 131 |
| 13 | Goto 1 | | 134 |
| 14 | | | |
| 15 | | | |
| 16 | | | |
| 17 | | | |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |
| 22 | | | |
| 23 | | | |
| 24 | | | |

Memory contents

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

# CASIO PROGRAM SHEET

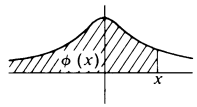
| Program for | Normal distribution | No. | 6 |
|---|---|---|---|

## Description

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

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

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




$$\text{Put } t = \frac{1}{1+Px}$$

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

$P = 0.2316419$

$C_1 = 0.31938153$

$C_2 = -0.356563782$

$C_3 = 1.78147937$

$C_4 = -1.821255978$

$C_5 = 1.330274429$

## 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.

| Step | Key operation | Display | Step | Key operation | Display |
|---|---|---|---|---|---|
| 1 | AC SHIFT Mcl EXE | Mcl<br>0. | | | |
| 2 | FILE | DISTRIBUTION ➡<br>F1 | | | |
| 3 | EXE | X?<br>0. | | | |
| 4 | 1.18 EXE | PX=<br>0.880999696 | | | |
| 5 | EXE | X?<br>1.18 | | | |
| 6 | 0.7 EXE | PX=<br>0.758036136 | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

No. 6

| Line | MODE EXP Program | | | | | | | | | | | | | | Notes | Number of steps |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | D | I | S | T | R | I | B | U | T | I | O | N | | | | 14 |
| L1 | Lbl | 0 | | | | | | | | | | | | | | 17 |
| 2 | { | X | } | | | | | | | | | | | | | 21 |
| 3 | T | = | 1 | ⁄ | ( | 1 | + | 0 | . | 2 | 3 | 1 | 6 | 4 | 1 | |
| | 9 | X | ) | | | | | | | | | | | | | 40 |
| 4 | Q | = | 1 | ⁄ | √ | 2 | π | × | $e^x$ | ( | − | X | $x^2$ | ⁄ | 2 | |
| | ) | | | | | | | | | | | | | | | 57 |
| 5 | A | = | 0 | . | 3 | 1 | 9 | 3 | 8 | 1 | 5 | 3 | | | | 70 |
| 6 | B | = | (−) | 0 | . | 3 | 5 | 6 | 5 | 6 | 3 | 7 | 8 | 2 | | 85 |
| 7 | C | = | 1 | . | 7 | 8 | 1 | 4 | 7 | 9 | 3 | 7 | | | | 98 |
| 8 | D | = | (−) | 1 | . | 8 | 2 | 1 | 2 | 5 | 5 | 9 | 7 | 8 | | 113 |
| 9 | E | = | 1 | . | 3 | 3 | 0 | 2 | 7 | 4 | 4 | 2 | 9 | | | 127 |
| 10 | P | ,, | P | X | ,, | = | 1 | − | Q | ( | A | T | + | B | T | |
| | $x^2$ | + | C | T | $x^y$ | 3 | + | D | T | $x^y$ | 4 | + | E | T | $x^y$ | |
| | 5 | ) | ◢ | | | | | | | | | | | | | |
| 11 | Goto | 0 | | | | | | | | | | | | | | 161 |
| 12 | | | | | | | | | | | | | | | | 164 |
| 13 | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | | |

Memory contents

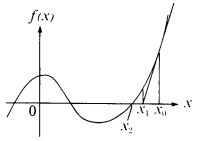
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | $C_1$ | H | | O | | V | |
| B | $C_2$ | I | | P | $\phi(x)$ | W | |
| C | $C_3$ | J | | Q | $\phi t$ | X | $x$ |
| D | $C_4$ | K | | R | | Y | |
| E | $C_5$ | L | | S | | Z | |
| F | | M | | T | $t$ | | |
| G | | N | | U | | | |

# CASIO PROGRAM SHEET

| Program for | No. |
|---|---|
| **Numerical solution of an equation (Newton's law)** | **7** |

## Description

Using Newton's law to calculate $x$ so that $f(x)=0$ in $y=f(x)$




Parameters:

$x_0$ ... Initial value

$h$ ... Micro interval in direction of $x$ axis when numeric differential is performed for points $(x, f(x))$

$\varepsilon$ ... Convergent condition of solution ("$\varepsilon$" used to continue calculation until inequality in $\varepsilon > |x_{n+1} - x_n|$ is developed).

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad f'(x) = \frac{f(x+h)-f(x)}{h}$$

$x_n$ is root for $|x_{n+1} - x_n| < \varepsilon$

## Example

Write a program for the following: $f(x) = ax^3 + bx^2 + cx + d$

$f(x) = 2x^3 + 3x^2 - x - 5$

$x_0 = 1,\ \varepsilon = 1 \times 10^{-5},\ h = 0.001$

**Note:** If solution does not appear after an extended period, no root exists. In this case, press AC to abort execution and re-execute with a different value for $x_0$.

## Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below.

| Step | Key operation | Display |
|---|---|---|
| 1 | AC SHIFT Mcl EXE | Mcl<br>0. |
| 2 | FILE | NEWTON<br>F1 |
| 3 | EXE | AX$x^y$3+BX²+CX→<br>0. |
| 4 | (After approx. 1 second) | A?<br>0. |
| 5 | 2 EXE | B?<br>0. |
| 6 | 3 EXE | C?<br>0. |
| 7 | (−) 1 EXE | D?<br>0. |
| 8 | (−) 5 EXE | EPSILON?<br>0. |
| 9 | 1 EXP (−) 5 EXE | X0 ?<br>0. |
| 10 | 1 EXE | H?<br>0. |

| Step | Key operation | Display |
|---|---|---|
| 11 | 0.001 EXE<br>(After approx. 9 seconds) | CALCULATING |
| 12 | | ANSWER=<br>1.084900341 |

| No. | 7 |
|---|---|

| Line | Program | Notes | Number of steps |
|---|---|---|---|
| F1 | MODE EXP N E W T O N | | 8 |
| L1 | " A X $x^y$ 3 + B X $x^2$ + C X + D = | | |
| | 0 " : Pause 2 | | 29 |
| 2 | A : B : C : D | | 37 |
| 3 | E " E P S I L O N " : P " X 0 | | |
| | " : H | | 56 |
| 4 | " C A L C U L A T I N G " | | 70 |
| 5 | Lbl 1 | | 73 |
| 6 | S = P : N = 2 | | 81 |
| 7 | Lbl 2 | | 84 |
| 8 | Y = A P $x^2$ P + B P $x^2$ + C P + D | | 100 |
| 9 | P = P + H : N = N − 1 | | 112 |
| 10 | N ≠ 0 ⇒ Z = Y : Goto 2 ◣ | | 124 |
| 11 | Y = ( Y − Z ) ⁄ H | | 134 |
| 12 | Z = S − Z ⁄ Y | | 142 |
| 13 | Abs ( Z − S ) ≥ E ⇒ P = Z : Goto 1 | | |
| | ◣ | | 159 |
| 14 | S : " A N S W E R = " ◢ | | 172 |
| 15 | | | |
| 16 | | | |
| 17 | | | |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |
| 22 | | | |
| 23 | | | |
| 24 | | | |

Memory contents

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | $a$ | H | $h$ | O | | V | |
| B | $b$ | I | | P | $x_0$ | W | |
| C | $c$ | J | | Q | | X | |
| D | $d$ | K | | R | | Y | $ax^3 + bx^2 + cx + d$ |
| E | $\varepsilon$ | L | | S | $x_n$ | Z | $x_{n+1}$ |
| F | | M | | T | | | |
| G | | N | $n$ | U | | | |

# CASIO PROGRAM SHEET

| Program for | Quadratic equation | No. | 8 |
|---|---|---|---|

## Description

$ax^2+bx+c=0$ {(Condition) $a \neq 0$ Accuracy to 6 significant digits.

By inputting coefficients $a$, $b$, and $c$ in the above formula, solutions for $\alpha$ and $\beta$ can be determined.

The root formula is used as shown below:

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

When it is considered that $d=b^2-4ac$:

i) When $d>0$, real roots of $\alpha$ and $\beta$ exist $\alpha=\frac{-b+\sqrt{d}}{2a}$, $\beta=\frac{-b-\sqrt{d}}{2a}$

ii) When $d=0$, the real root of $\alpha$ exits $\alpha=\frac{-b}{2a}$

iii) When $d<0$, imaginary roots of $\alpha$ and $\beta$ exist $\alpha=\frac{-b}{2a}+\frac{\sqrt{-d}}{2a}i$

$$\beta=\frac{-b}{2a}-\frac{\sqrt{-d}}{2a}i$$

## Example

Determine solutions for the following quadratic equations:

1) $2x^2-x-15=0$
2) $4x^2-12x+9=0$
3) $x^2+x+1=0$

## Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below.

| Step | Key operation | Display |
|---|---|---|
| 1 | AC SHIFT Mcl EXE | Mcl<br>0. |
| 2 | FILE | QUADRATIC<br>F1 |
| 3 | EXE | AX²+BX+C=0<br>0. |
| 4 | (After approx. 1 second) | A?<br>0. |
| 5 | 2 EXE | B?<br>0. |
| 6 | (−) 1 EXE | C?<br>0. |
| 7 | (−) 15 EXE | X1=<br>3. |
| 8 | EXE | X2=<br>−2.5 |
| 9 | EXE | AX²+BX+C=0<br>−2.5 |
| 10 | (After approx. 1 second) | A?<br>2. |
| 11 | 4 EXE | B?<br>−1. |

| Step | Key operation | Display |
|---|---|---|
| 12 | (−) 12 EXE | C?<br>−15. |
| 13 | 9 EXE | X=<br>1.5 |
| 14 | EXE | AX²+BX+C=0<br>1.5 |
| 15 | (After approx. 1 second) | A?<br>4. |
| 16 | 1 EXE | B?<br>−12. |
| 17 | 1 EXE | C?<br>9. |
| 18 | 1 EXE | X1:REAL P.<br>−0.5 |
| 19 | EXE | IMAGINARY P.<br>0.866025 |
| 20 | EXE | X2:REAL P.<br>−0.5 |
| 21 | | IMAGINARY P.<br>−0.866025 |

| No. | 8 |
|---|---|

| Line | MODE EXP | Program | | | | | | | | | | | | | | Notes | Number of steps |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | Q | U | A | D | R | A | T | I | C | | | | | | | | 11 |
| L1 | Lbl | R | T | N | | | | | | | | | | | | | 16 |
| 2 | " | A | X | $x^2$ | + | B | X | + | C | = | 0 | " | : | Pause | 2 | | 32 |
| 3 | Lbl | 0 | : | { | A | B | C | } | | | | | | | | | 41 |
| 4 | A | = | 0 | ⇒ | " | A | ≠ | 0 | " | : | Pause | 1 | : | Goto | 0 | | |
| | ◣ | | | | | | | | | | | | | | | | 58 |
| 5 | B | : | C | | | | | | | | | | | | | | 62 |
| 6 | D | = | B | $x^2$ | − | 4 | A | C | | | | | | | | | 71 |
| 7 | D | > | 0 | ⇒ | Prog | S | U | B | 1 | : | Goto | R | T | N | ◣ | | 87 |
| 8 | D | = | 0 | ⇒ | Prog | S | U | B | 2 | : | Goto | R | T | N | ◣ | | 103 |
| 9 | Prog | S | U | B | 3 | | | | | | | | | | | | 109 |
| 10 | Goto | R | T | N | | | | | | | | | | | | | 114 |
| | | | | | | | | | | | | | | | | | |
| F2 | S | U | B | 1 | | | | | | | | | | | | | 6 |
| L1 | ( | (−) | B | + | √ | D | ) | / | 2 | A | | | | | | | 17 |
| 2 | Prog | R | N | D | | | | | | | | | | | | | 22 |
| 3 | P | " | X | 1 | " | = | Ans | ◢ | | | | | | | | | 31 |
| 4 | ( | (−) | B | − | √ | D | ) | / | 2 | A | | | | | | | 42 |
| 5 | Prog | R | N | D | | | | | | | | | | | | | 47 |
| 6 | Q | " | X | 2 | " | = | Ans | ◢ | | | | | | | | | 56 |
| | | | | | | | | | | | | | | | | | |
| F3 | S | U | B | 2 | | | | | | | | | | | | | 6 |
| L1 | (−) | B | / | 2 | A | | | | | | | | | | | | 12 |
| 2 | Prog | R | N | D | | | | | | | | | | | | | 17 |
| 3 | P | " | X | " | = | Ans | ◢ | | | | | | | | | | 25 |
| | | | | | | | | | | | | | | | | | |
| F4 | S | U | B | 3 | | | | | | | | | | | | | 6 |
| L1 | (−) | B | / | 2 | A | | | | | | | | | | | | 12 |

Memory contents

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | $a$ | H | | O | | V | |
| B | $b$ | I | | P | $\frac{-b+\sqrt{d}}{2a}$, $\frac{-b}{2a}$ | W | |
| C | $c$ | J | | Q | $\frac{-b-\sqrt{d}}{2a}$, $\frac{\sqrt{-d}}{2a}$ | X | |
| D | $b^2-4ac$ | K | | R | | Y | |
| E | | L | | S | | Z | |
| F | | M | | T | | | |
| G | | N | | U | | | |

# CASIO PROGRAM SHEET

| Program for | Quadratic equation | No. | 8 |
|---|---|---|---|

| Step | Key operation | Display | Step | Key operation | Display |
|---|---|---|---|---|---|
| 1 | | | 11 | | |
| 2 | | | 12 | | |
| 3 | | | 13 | | |
| 4 | | | 14 | | |
| 5 | | | 15 | | |
| 6 | | | 16 | | |
| 7 | | | 17 | | |
| 8 | | | 18 | | |
| 9 | | | 19 | | |
| 10 | | | 20 | | |

No. 8

| Line | MODE EXP | Program | | | | | | | | | | | | | Notes | Number of steps |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | Prog | R | N | D | | | | | | | | | | | | 17 |
| 3 | P | = | Ans | | | | | | | | | | | | | 21 |
| 4 | √ | (−) | D | / | 2 | A | | | | | | | | | | 28 |
| 5 | Prog | R | N | D | | | | | | | | | | | | 33 |
| 6 | Q | = | Ans | | | | | | | | | | | | | 37 |
| 7 | P | : | ” | X | 1 | : | R | E | A | L | | P | | ” | ◢ | 53 |
| 8 | Q | : | ” | I | M | A | G | I | N | A | R | Y | | P | | |
| | ” | ◢ | | | | | | | | | | | | | | 71 |
| 9 | P | : | ” | X | 2 | : | R | E | A | L | | P | | ” | ◢ | 87 |
| 10 | (−) | Q | : | ” | I | M | A | G | I | N | A | R | Y | | P | |
| | | ” | ◢ | | | | | | | | | | | | | 106 |
| | | | | | | | | | | | | | | | | |
| F5 | R | N | D | | | | | | | | | | | | | 5 |
| L1 | Sci | 6 | : | Rnd | : | Norm | | | | | | | | | | 12 |
| 2 | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | |

Memory contents

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | $a$ | H | | O | | V | |
| B | $b$ | I | | P | $\frac{-b+\sqrt{d}}{2a}, \frac{-b}{2a}$ | W | |
| C | $c$ | J | | Q | $\frac{-b-\sqrt{d}}{2a}, \frac{\sqrt{-d}}{2a}$ | X | |
| D | $b^2-4ac$ | K | | R | | Y | |
| E | | L | | S | | Z | |
| F | | M | | T | | | |
| G | | N | | U | | | |

# CASIO PROGRAM SHEET

| Program for | Complex numbers | No. | 9 |
|---|---|---|---|

**Description**

$Z_1 = x_1 + iy_1 \qquad r_1 = \sqrt{x_1^2 + y_1^2} \quad , \quad \theta_1 = \tan^{-1}\frac{y_1}{x_1}$

$Z_2 = x_2 + iy_2 \qquad r_2 = \sqrt{x_2^2 + y_2^2} \quad , \quad \theta_2 = \tan^{-1}\frac{y_2}{x_2}$

**Example**

$Z_1 = 2 + \sqrt{3}i$

$Z_2 = 4 - i$

- Sum, difference $Z_1 \pm Z_2 = (x_1 \pm x_2) + i(y_1 \pm y_2)$
- Product $Z_1 \times Z_2 = (x_1 x_2 - y_1 y_2) + i(x_1 y_2 + x_2 y_1)$
- Quotient $\dfrac{Z_1}{Z_2} = \dfrac{(x_1 x_2 + y_1 y_2) + i(x_2 y_1 - x_1 y_2)}{x_2^2 + y_2^2}$ (Condition) $Z_2 \neq 0$ / $x_2 \neq 0,\ y_2 \neq 0$
- $n$-th power $Z_1^n = r^n \cdot e^{in\theta} = (r^n \cos n\theta) + i(r^n \sin n\theta)$
- $x, y \rightarrow r, \theta$
- $r, \theta \rightarrow x, y$

**Preparation and operation**

- Store the program written on the next page.
- Execute the program as shown below.

Note: • If "Mem ERROR" appears on the display while executing the program, press [MODE] [Ans] 0 [EXE].

• When the message "1:+ 2:− 3:×?➡" appears on the display, you can input any of the following values to specify the type of operation to be performed.

1 ..... $Z_1 + Z_2$ 2 ..... $Z_1 - Z_2$ 3 ..... $Z_1 \times Z_2$ 4 ..... $Z_1 \div Z_2$ 5 ..... $Z_1 n$
6 ..... $x, y \rightarrow r, \theta$ 7 ..... $r, \theta \rightarrow x, y$ 8 ..... $Z_1, Z_2$ input 9 ..... Quit

You should input the full message as follows:
1:+ 2:− 3:× 4:/ 5:$Zx^y$N 6:➡POL 7:➡REC 8:INPUT 9:QUIT

While this message is displayed, use the [◁] and [▷] keys to scroll left and right to view the parts of the message that don't fit on the display.

| Step | Key operation | Display |
|---|---|---|
| 1 | [AC] [SHIFT] [Mcl] [EXE] | Mcl<br>0. |
| 2 | [FILE] | COMPLEX<br>F1 |
| 3 | [EXE] | 1:+ 2:− 3:×?➡<br>0. |
| 4 | (Input data of complex numbers) 8 [EXE] | X1(Z1)?<br>0. |
| 5 | 2 [EXE] | Y1(Z1)?<br>0. |
| 6 | [√] 3 [EXE] | X2(Z2)?<br>0. |
| 7 | 4 [EXE] | Y2(Z2)?<br>0. |
| 8 | [(−)] 1 [EXE] | 1:+ 2:− 3:×?➡<br>8. |
| 9 | (Calculate sum) 1 [EXE] | REAL=<br>6. |
| 10 | [EXE] | IMAGE=<br>0.732050807 |
| 11 | [EXE] | 1:+ 2:− 3:×?➡<br>1. |
| 12 | (Calculate product) 3 [EXE] | REAL=<br>9.732050808 |
| 13 | [EXE] | IMAGE=<br>4.92820323 |
| 14 | [EXE] | 1:+ 2:− 3:×?➡<br>3. |
| 15 | 9 [EXE] | END OF JOB<br>0. |

| No. | 9 |
|---|---|

| Line | Program | | | | | | | | | | | | | | | | Notes | Number of steps |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | C | O | M | P | L | E | X | | | | | | | | | | | 9 |
| L1 | Rad | | | | | | | | | | | | | | | | | 11 |
| 2 | Defm | : | N | = | Ans | | | | | | | | | | | | | 17 |
| 3 | Defm | N | + | 6 | | | | | | | | | | | | | | 22 |
| 4 | Lbl | M | E | N | U | | | | | | | | | | | | | 28 |
| 5 | { | O | } | : | O | " | 1 | : | + | | 2 | : | − | | 3 | | | |
| | : | × | | 4 | : | / | | 5 | : | Z | $x^y$ | N | | 6 | : | | | |
| | − | > | P | O | L | | 7 | : | − | > | R | E | C | | 8 | | | |
| | : | I | N | P | U | T | | 9 | : | Q | U | I | T | " | | | | 88 |
| 6 | O | = | 1 | ⇒ | Prog | + | | | | | | | | | | | | 95 |
| 7 | ⇜ | O | = | 2 | ⇒ | Prog | − | | | | | | | | | | | 103 |
| 8 | ⇜ | O | = | 3 | ⇒ | Prog | × | | | | | | | | | | | 111 |
| 9 | ⇜ | O | = | 4 | ⇒ | Prog | / | | | | | | | | | | | 119 |
| 10 | ⇜ | O | = | 5 | ⇒ | Prog | N | | | | | | | | | | | 127 |
| 11 | ⇜ | O | = | 6 | ⇒ | Prog | P | O | L | | | | | | | | | 137 |
| 12 | ⇜ | O | = | 7 | ⇒ | Prog | R | E | C | | | | | | | | | 147 |
| 13 | ⇜ | O | = | 8 | ⇒ | Prog | I | N | P | U | T | | | | | | | 159 |
| 14 | ⇜ | O | = | 9 | ⇒ | Goto | E | N | D | ◣ | ◣ | ◣ | ◣ | ◣ | ◣ | | | |
| | ◣ | ◣ | ◣ | | | | | | | | | | | | | | | 178 |
| 15 | Goto | M | E | N | U | | | | | | | | | | | | | 184 |
| 16 | Lbl | E | N | D | | | | | | | | | | | | | | 189 |
| 17 | Defm | N | | | | | | | | | | | | | | | | 192 |
| 18 | " | E | N | D | | O | F | | J | O | B | " | | | | | | 205 |
| | | | | | | | | | | | | | | | | | | |
| F2 | + | | | | | | | | | | | | | | | | | 3 |

Memory contents

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A | | H | | O | For judgement | V | $r, x$ | | |
| B | | I | | P | $x^2 + y^2$ | W | $\theta, y$ | | |
| C | | J | | Q | | X | | | |
| D | | K | | R | $n$ | Y | | | |
| E | | L | | S | | Z | | | |
| F | | M | | T | | | | | |
| G | | N | Expanded memories | U | | | | | |

Expanded memories

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Z[N+1] | $x_1$ | Z[N+3] | $x_2$ | Z[N+5] | $x$ | | |
| Z[N+2] | $y_1$ | Z[N+4] | $y_2$ | Z[N+6] | $y$ | | |

No. 9

| Line | MODE EXP Program | | | | | | | | | | | | | | | Notes | Number of steps |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L1 | Z | [ | N | + | 5 | ] | " | R | E | A | L | " | = | Z | [ | | |
| | N | + | 1 | ] | + | Z | [ | N | + | 3 | ] | ◢ | | | | | 31 |
| 2 | Z | [ | N | + | 6 | ] | " | I | M | A | G | E | " | = | Z | | |
| | [ | N | + | 2 | ] | + | Z | [ | N | + | 4 | ] | ◢ | | | | 60 |
| | | | | | | | | | | | | | | | | | |
| F3 | − | | | | | | | | | | | | | | | | 3 |
| L1 | Z | [ | N | + | 5 | ] | " | R | E | A | L | " | = | Z | [ | | |
| | N | + | 1 | ] | − | Z | [ | N | + | 3 | ] | ◢ | | | | | 30 |
| 2 | Z | [ | N | + | 6 | ] | " | I | M | A | G | E | " | = | Z | | |
| | [ | N | + | 2 | ] | + | Z | [ | N | + | 4 | ] | ◢ | | | | 59 |
| | | | | | | | | | | | | | | | | | |
| F4 | × | | | | | | | | | | | | | | | | 3 |
| L1 | Z | [ | N | + | 5 | ] | " | R | E | A | L | " | = | Z | [ | | |
| | N | + | 1 | ] | Z | [ | N | + | 3 | ] | − | Z | [ | N | + | | |
| | 2 | ] | Z | [ | N | + | 4 | ] | ◢ | | | | | | | | 43 |
| 2 | Z | [ | N | + | 6 | ] | " | I | M | A | G | E | " | = | Z | | |
| | [ | N | + | 1 | ] | Z | [ | N | + | 4 | ] | + | Z | [ | N | | |
| | + | 2 | ] | Z | [ | N | + | 3 | ] | ◢ | | | | | | | 84 |
| | | | | | | | | | | | | | | | | | |
| F5 | / | | | | | | | | | | | | | | | | 3 |
| L1 | P | = | Z | [ | N | + | 3 | ] | $x^2$ | + | Z | [ | N | + | 4 | | |
| | ] | $x^2$ | | | | | | | | | | | | | | | 21 |
| 2 | Z | [ | N | + | 5 | ] | " | R | E | A | L | " | = | ( | Z | | |
| | [ | N | + | 1 | ] | Z | [ | N | + | 3 | ] | + | Z | [ | N | | |
| | + | 2 | ] | Z | [ | N | + | 4 | ] | ) | / | P | ◢ | | | | 65 |

| Memory contents | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | | H | | O | For judgement | V | $r, x$ | |
| B | | I | | P | $x^2+y^2$ | W | $\theta, y$ | |
| C | | J | | Q | | X | | |
| D | | K | | R | $n$ | Y | | |
| E | | L | | S | | Z | | |
| F | | M | | T | | | | |
| G | | N | Expanded memories | U | | | | |

| Expanded memories | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Z[N+1] | $x_1$ | Z[N+3] | $x_2$ | Z[N+5] | $x$ | | |
| Z[N+2] | $y_1$ | Z[N+4] | $y_2$ | Z[N+6] | $y$ | | |

No. 9

| Line | MODE EXP Program | | | | | | | | | | | | | | | Notes | Number of steps |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | Z | [ | N | + | 6 | ] | " | I | M | A | G | E | " | = | ( | | |
| | Z | [ | N | + | 3 | ] | Z | [ | N | + | 2 | ] | − | Z | [ | | |
| | N | + | 1 | ] | Z | [ | N | + | 4 | ] | ) | / | P | ◢ | | | 110 |
| | | | | | | | | | | | | | | | | | |
| F6 | N | | | | | | | | | | | | | | | | 3 |
| L1 | Lbl | Ø | | | | | | | | | | | | | | | 6 |
| 2 | Prog | S | E | L | | | | | | | | | | | | | 11 |
| 3 | Q | = | 9 | ⇒ | Goto | E | N | D | ◣ | | | | | | | | 21 |
| 4 | Pol( | Z | [ | N | + | Q | ] | , | Z | [ | N | + | Q | + | 1 | | |
| | ] | ) | | | | | | | | | | | | | | | 39 |
| 5 | { | R | } | | | | | | | | | | | | | | 43 |
| 6 | Z | [ | N | + | 5 | ] | " | R | E | A | L | " | = | V | $x^y$ | | |
| | R | " | N | " | cos | R | W | ◢ | | | | | | | | | 67 |
| 7 | Z | [ | N | + | 6 | ] | " | I | M | A | G | E | " | = | V | | |
| | $x^y$ | R | sin | R | W | ◢ | | | | | | | | | | | 89 |
| 8 | Goto | Ø | | | | | | | | | | | | | | | 92 |
| 9 | Lbl | E | N | D | | | | | | | | | | | | | 97 |
| | | | | | | | | | | | | | | | | | |
| F7 | P | O | L | | | | | | | | | | | | | | 5 |
| L1 | Lbl | Ø | | | | | | | | | | | | | | | 8 |
| 2 | Prog | S | E | L | | | | | | | | | | | | | 13 |
| 3 | Q | ≠ | 9 | ⇒ | Pol( | Z | [ | N | + | Q | ] | , | Z | [ | N | | |
| | + | Q | + | 1 | ] | ) | | | | | | | | | | | 35 |
| 4 | V | : | " | R | = | " | ◢ | | | | | | | | | | 43 |
| 5 | W | : | " | T | H | E | T | A | = | " | ◢ | | | | | | 55 |

| Memory contents | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | | H | | O | For judgement | V | $r, x$ | |
| B | | I | | P | $x^2+y^2$ | W | $\theta, y$ | |
| C | | J | | Q | | X | | |
| D | | K | | R | $n$ | Y | | |
| E | | L | | S | | Z | | |
| F | | M | | T | | | | |
| G | | N | Expanded memories | U | | | | |

| Expanded memories | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Z[N+1] | $x_1$ | Z[N+3] | $x_2$ | Z[N+5] | $x$ | | |
| Z[N+2] | $y_1$ | Z[N+4] | $y_2$ | Z[N+6] | $y$ | | |

No. 9

| Line | Program | Notes | Number of steps |
|---|---|---|---|
| 6 | Goto 0 ◣ | | 59 |
| | | | |
| F8 | I N P U T | | 7 |
| L1 | S = Z [ N + 1 ] : { S } : Z [ | | |
| | N + 1 ] = S ” X 1 ( Z 1 ) ” | | 37 |
| 2 | S = Z [ N + 2 ] : { S } : Z [ | | |
| | N + 2 ] = S ” Y 1 ( Z 1 ) ” | | 67 |
| 3 | S = Z [ N + 3 ] : { S } : Z [ | | |
| | N + 3 ] = S ” X 2 ( Z 2 ) ” | | 97 |
| 4 | S = Z [ N + 4 ] : { S } : Z [ | | |
| | N + 4 ] = S ” Y 2 ( Z 2 ) ” | | 127 |
| | | | |
| F9 | S E L | | 5 |
| L1 | Lbl 0 | | 8 |
| 2 | { Q } : Q ” 1 : Z 1 2 : Z 2 | | |
| | 9 : Q U I T ” | | 32 |
| 3 | Q ≠ 1 ⇒ Q ≠ 2 ⇒ Q ≠ 9 ⇒ Goto 0 ◣ | | |
| | ◣ ◣ | | 50 |
| 4 | Q = 2 ⇒ Q = 3 ◣ | | 59 |
| | | | |
| F10 | R E C | | 5 |
| L1 | Lbl 0 | | 8 |
| 2 | Prog S E L | | 13 |
| 3 | Q ≠ 9 ⇒ Rec( Z [ N + Q ] , Z [ N | | |
| | + Q + 1 ] ) | | 35 |

Memory contents

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | | H | | O | For judgement | V | $r$, $x$ |
| B | | I | | P | $x^2+y^2$ | W | $\theta$, $y$ |
| C | | J | | Q | | X | |
| D | | K | | R | $n$ | Y | |
| E | | L | | S | | Z | |
| F | | M | | T | | | |
| G | | N | Expanded memories | U | | | |

Expanded memories

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Z[N+1] | $x_1$ | Z[N+3] | $x_2$ | Z[N+5] | $x$ | | |
| Z[N+2] | $y_1$ | Z[N+4] | $y_2$ | Z[N+6] | $y$ | | |

No. 9

| Line | Program | Notes | Number of steps |
|---|---|---|---|
| 4 | V : ” X = ” ◢ | | 43 |
| 5 | W : ” Y = ” ◢ | | 51 |
| 6 | Goto 0 ◣ | | 55 |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |
| 11 | | | |
| 12 | | | |
| 13 | | | |
| 14 | | | |
| 15 | | | |
| 16 | | | |
| 17 | | | |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |
| 22 | | | |
| 23 | | | |
| 24 | | | |
| 25 | | | |
| 26 | | | |
| 27 | | | |
| 28 | | | |

Memory contents

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | | H | | O | For judgement | V | $r$, $x$ |
| B | | I | | P | $x^2+y^2$ | W | $\theta$, $y$ |
| C | | J | | Q | | X | |
| D | | K | | R | $n$ | Y | |
| E | | L | | S | | Z | |
| F | | M | | T | | | |
| G | | N | Expanded memories | U | | | |

Expanded memories

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Z[N+1] | $x_1$ | Z[N+3] | $x_2$ | Z[N+5] | $x$ | | |
| Z[N+2] | $y_1$ | Z[N+4] | $y_2$ | Z[N+6] | $y$ | | |

# CASIO PROGRAM SHEET

| Program for | No. |
|---|---|
| | |

| Step | Key operation | Display | Step | Key operation | Display |
|---|---|---|---|---|---|
| 1 | | | 11 | | |
| 2 | | | 12 | | |
| 3 | | | 13 | | |
| 4 | | | 14 | | |
| 5 | | | 15 | | |
| 6 | | | 16 | | |
| 7 | | | 17 | | |
| 8 | | | 18 | | |
| 9 | | | 19 | | |
| 10 | | | 20 | | |

No.

| Line | MODE EXP Program | Notes | Number of steps |
|---|---|---|---|
| | | | |

| 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 | | | | |

# CASIO PROGRAM SHEET

| Program for | No. |
|---|---|
| | |

| Step | Key operation | Display | Step | Key operation | Display |
|---|---|---|---|---|---|
| 1 | | | 11 | | |
| 2 | | | 12 | | |
| 3 | | | 13 | | |
| 4 | | | 14 | | |
| 5 | | | 15 | | |
| 6 | | | 16 | | |
| 7 | | | 17 | | |
| 8 | | | 18 | | |
| 9 | | | 19 | | |
| 10 | | | 20 | | |

No.

| Line | MODE EXP Program | Notes | Number of steps |
|---|---|---|---|
| | | | |

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 | | | |

Expanded memories

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |

## ■Error messages at-a-glance

| Message | Meaning | Countermeasure |
|---|---|---|
| Ma ERROR | ①Calculation exceeds range of operation.<br>②Calculation is performed outside the input range of a function.<br>③Illogical operation (division by zero, etc.) | ① ② ③<br>Check the input numeric value and correct it. When using memories, check that the numeric values stored in memories are correct. |
| Arg ERROR | •Argument input incorrectly. Ex. Negative value input for Defm, value other than 1 ~ 9 input for $n$ in integration calculation, etc. | •Re-enter argument 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 9 levels for numeric values and 24 levels for calculations.<br>•Divide the formula into two or more parts. |
| Syn ERROR | ①Calculation formula contains an error.<br>②Formula in a program contains an error. | ① ②<br>Use cursor keys to display the point where the error was generated and correct it. |
| Mem ERROR | •Memory expansion exceeds level remaining in program.<br>•Attempt to use a memory such as Z[5] when no memory has been expanded.<br>•Program written although no memory remains for program. | •Press MODE Ans (Defm) to expand memory to necessary level.<br>•Use memories within the current number of memories.<br>•Simplify program to fit within current available memory or delete unnecessary programs. |
| Ne ERROR | •Nesting of subroutines by execution exceeds 10-level limit. | •Ensure that Prog (filename) is not used to return from subroutines to main routine. If used, delete any unnecessary Prog (filename).<br>•Trace subroutine jump destinations and ensure that no jumps are made back to the original program area. Ensure that returns are made correctly. |
| Go ERROR | ①No corresponding Lbl (label name) to Goto (label name).<br>②No program stored in specified Prog (filename). | ①Correctly input an Lbl corresponding to Goto.<br>②Store a program in program area Prog (filename) or delete the Prog (filename) if unnecessary. |

## ■Input ranges of functions

| Function | Input range | Internal digits | Accuracy | Notes |
|---|---|---|---|---|
| sin<br>cos<br>tan | (Deg) $\lvert x\rvert < 9\times10^{9}$°<br>(Rad) $\lvert x\rvert < 5\times10^{7}\pi$rad<br>(Gra) $\lvert x\rvert < 1\times10^{10}$grad | 12 digits | As a rule, accuracy is ±1 at the 10th digit. | However, for tan $x$:<br>$\lvert x\rvert \neq 90(2n+1)$: Deg<br>$\lvert x\rvert \neq \pi/2(2n+1)$: Rad<br>$\lvert x\rvert \neq 100(2n+1)$: Gra |
| $\sin^{-1}$<br>$\cos^{-1}$ | $\lvert x\rvert \leqq 1$ | ,, | ,, | |
| $\tan^{-1}$ | $\lvert x\rvert < 1\times10^{100}$ | | | |
| sinh<br>cosh | $\lvert x\rvert \leqq 230.2585092$ | ,, | ,, | Note: For sinh and tanh, when $x=0$, errors are cumulative and accuracy is affected at a certain point. |
| tanh | $\lvert x\rvert < 1\times10^{100}$ | | | |
| $\sinh^{-1}$ | $\lvert x\rvert < 5\times10^{99}$ | ,, | ,, | |
| $\cosh^{-1}$ | $1 \leqq x < 5\times10^{99}$ | | | |
| $\tanh^{-1}$ | $\lvert x\rvert < 1$ | | | |
| log<br>ln | $1\times10^{-99} \leqq x < 1\times10^{100}$ | ,, | ,, | |
| $10^{x}$ | $-1\times10^{100} < x < 100$ | ,, | ,, | |
| $e^{x}$ | $-1\times10^{100} < x \leqq 230.2585092$ | | | |
| $\sqrt{}$ | $0 \leqq x < 1\times10^{100}$ | ,, | ,, | |
| $x^2$ | $\lvert x\rvert < 1\times10^{50}$ | | | |
| $x^{-1}$ | $\lvert x\rvert < 1\times10^{100}$, $x \neq 0$ | ,, | ,, | |
| $\sqrt[3]{}$ | $\lvert x\rvert < 1\times10^{100}$ | | | |
| $x!$ | $0 \leqq x \leqq 69$<br>($x$ is an integer) | ,, | ,, | |
| $n$Pr<br>$n$Cr | Result $< 1\times10^{100}$<br>$n$, $r$ ($n$ and $r$ are integers)<br>$0 \leqq r \leqq n$,<br>$n < 1\times10^{10}$ | ,, | ,, | |
| Pol $(x,y)$ | $\sqrt{x^2+y^2} < 1\times10^{100}$ | ,, | ,, | |
| Rec $(r,\theta)$ | $0 \leqq r < 1\times10^{100}$<br>(Deg) $\lvert\theta\rvert < 9\times10^{9}$°<br>(Rad) $\lvert\theta\rvert < 5\times10^{7}\pi$rad | ,, | ,, | However, for tan $x$:<br>$\lvert x\rvert \neq 90(2n+1)$: Deg<br>$\lvert x\rvert \neq \pi/2(2n+1)$: Rad |

| Function | Input range | Internal digits | Accuracy | Notes |
|---|---|---|---|---|
| °' '' | $\|a\|, b, c < 1 \times 10^{100}$<br>$0 \leqq b, c$ | 12 digits | As a rule, accuracy is ±1 at the 10th digit. | |
| $\overleftarrow{\text{°' ''}}$ | $\|x\| < 1 \times 10^{100}$<br>Hexadecimal display:<br>$\|x\| \leqq 27777.77777$ | | | |
| $x^y$ | $x > 0$:<br>$-1 \times 10^{100} < y \log x < 100$<br>$x = 0$: $y > 0$<br>$x < 0$:<br>$y = n, \frac{1}{2n+1}$<br>($n$ is an integer)<br>However;<br>$-1 \times 10^{100} < \frac{1}{y} \log \|x\| < 100$ | ,, | ,, | |
| $\sqrt[x]{y}$ | $x > 0$: $y \neq 0$<br>$-1 \times 10^{100} < \frac{1}{y} \log x < 100$<br>$x = 0$: $y > 0$<br>$x < 0$: $y = 2n+1, \frac{1}{n}$<br>($n \neq 0$, $n$ is an integer)<br>However;<br>$-1 \times 10^{100} < \frac{1}{y} \log \|x\| < 100$ | ,, | ,, | |
| $a^{b/c}$ | •Results<br>Total of integer, numerator and denominator must be within 10 digits (includes division marks).<br>•Input<br>Result displayed as a fraction for integer when integer, numerator and denominator are less than $1 \times 10^{10}$. | ,, | ,, | |
| SD<br>(LR) | $\|x\| < 1 \times 10^{50}$<br>$\|y\| < 1 \times 10^{50}$<br>$\|n\| < 1 \times 10^{100}$<br>$x\sigma_n$, $y\sigma_n$, $\bar{x}$, $\bar{y}$, A, B, $r$: $n \neq 0$<br>$x\sigma_{n-1}$, $y\sigma_{n-1}$: $n \neq 0, 1$ | ,, | ,, | |

| Function | Input range |
|---|---|
| BASE-N | Values after variable within following range:<br>Dec: $-2147483648 \leqq x \leqq -1$ (negative)<br>$0 \leqq x \leqq 2147483647$ (0, positive)<br>Bin: $10000000000000000000000000000000 \leqq x \leqq 11111111111111111111111111111111$ (negative)<br>$0 \leqq x \leqq 01111111111111111111111111111111$ (0, positive)<br>Oct: $20000000000 \leqq x \leqq 37777777777$ (negative)<br>$0 \leqq x \leqq 17777777777$ (0, positive)<br>Hex: $80000000 \leqq x \leqq$ FFFFFFFF (negative)<br>$0 \leqq x \leqq$ 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-4500PA

### Calculations

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

**Built-in functions:**
Trigonometric/inverse trigonometric functions (units or 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 conversions/calculations, coordinate transformations, permutations/combinations, $\pi$, random numbers, absolute values, integers, fractions.

**Statistical calculation functions:**
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 $y$.

**Integration calculation:** using Simpson's rule.

**Memories:**
26 standard (163 maximum)

**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 — $10^{-9} > |x|, |x| \geqq 10^{10}$

### Programs

**Number of steps:** 1,103 maximum

**Jump function:**
Unconditional jump (Goto, Lbl)
Conditional jump ($=$, $\neq$, $>$, $<$, $\geqq$, $\leqq$)

**Subroutines Prog:** 10 routines

**Number of stored programs:** As many files as total memory capacity allows

**Check function:** Program checking, debugging, deletion, addition, etc.

### Common section

**Display system and contents:**
Liquid crystal display, dot 12 digits, 10-digit mantissa and 2-digit exponent, binary, octal, hexadecimal display, sexagesimal display, conditional displays (WRT, FILE, EDIT, Disp, LR, SD, D, R, G, S, F, M, A, hyp, d, H, b, o, Fix, Sci, Eng, ←, →)

**Character display function:**
Function commands, program commands, alphabet characters (12 maximum)

**Error check function:**
Checks for values exceeding $10^{100}$, illogical calculations and illogical jumps, error messages displayed.

**Power supply:**
1 lithium battery for normal operation (CR2032); 1 lithium battery for memory protection (CR2032)

**Power consumption:** 0.001W

**Battery life:** Approximately 5,000 hours on type CR2032 battery.

**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$^9/_{16}$″D)

**Weight:** 85 g (3.2 oz) including batteries