CASIO®

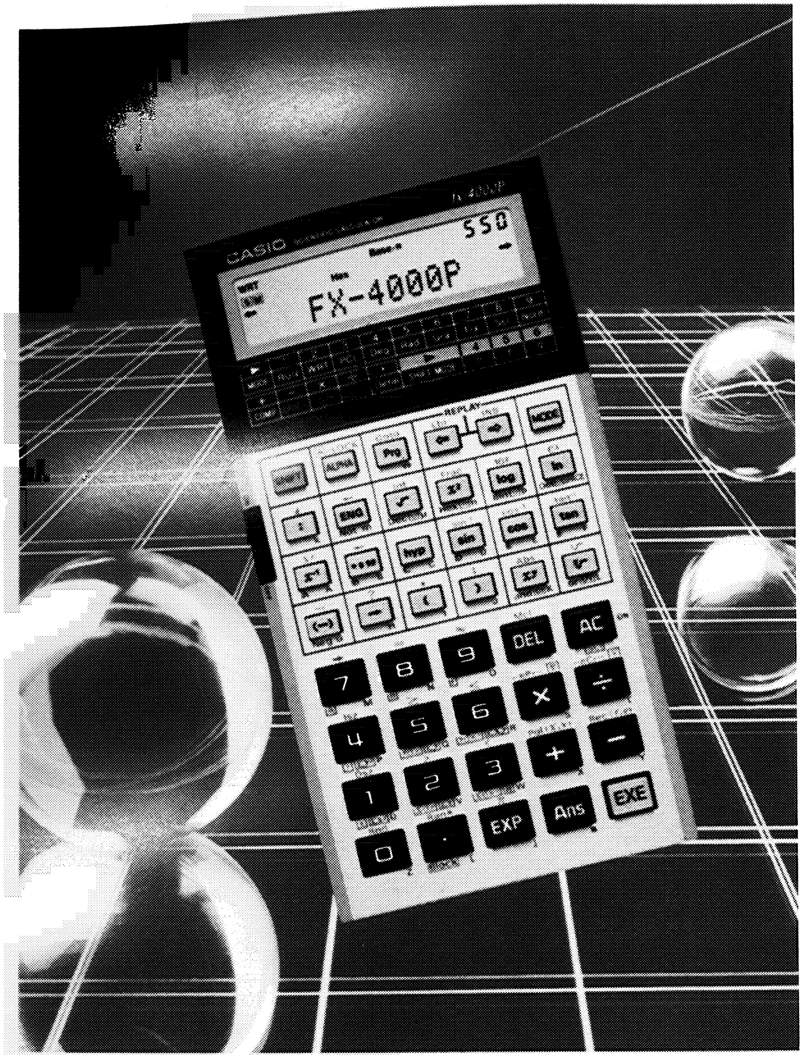
SH1001811A 英 Printed in Japan

fx-4000P OWNER'S MANUAL CASIO®

英

fx-4000P

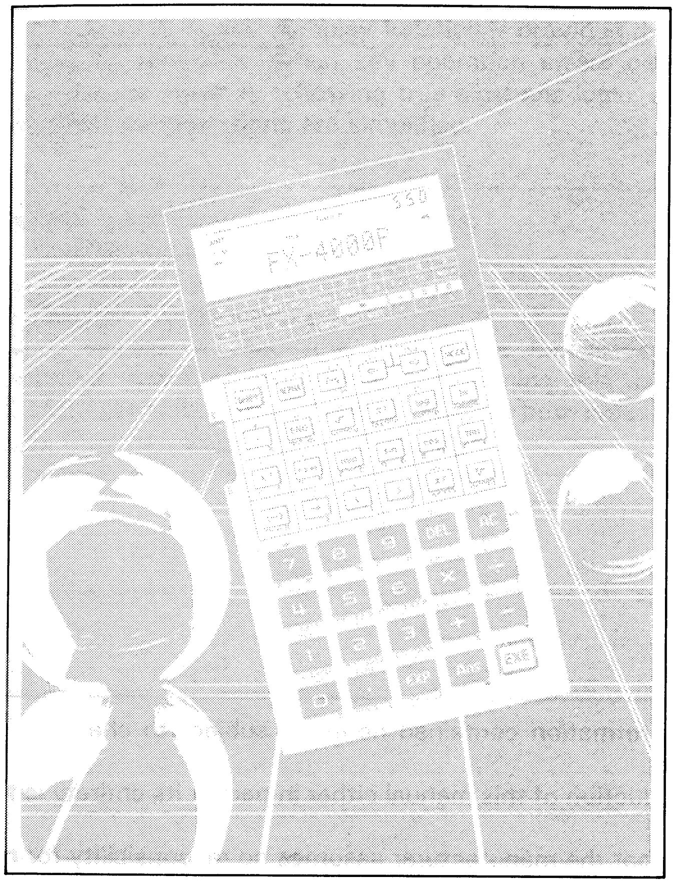
OWNER'S MANUAL



CASIO®

fx-4000P

OWNER'S MANUAL



CASIO®

FOREWORD

Thank you for your purchase of the CASIO fx-4000P.
This unit is an advanced hand held programmable computer capable of alphabetic display. Besides a programming function which is useful for performing repeat or complex computations, 83 scientific functions are also provided.
Manual computations can be easily performed following written formulas (true algebraic logic). A replay function is provided that allows confirmation or correction when key operation errors occur. Programs can also be input by following true algebraic logic, so repeat and/or complex computations are simplified.

This manual is composed of three sections:

1. Configuration and Operation
2. Manual Computations
3. Program Computations

Section 1 should be read first to become familiar with the nomenclature, handling and cautions concerning this unit. Sections 2 and 3 can then be read in order to master both types of computations through samples and explanations.

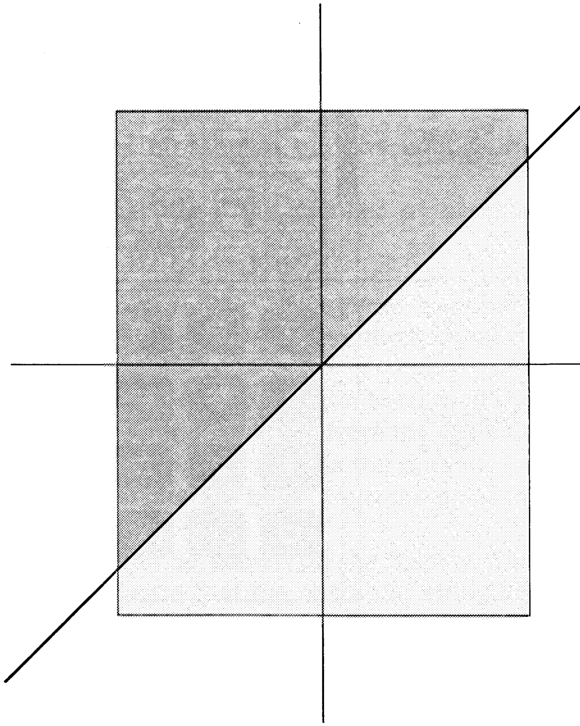
CONTENTS

HANDLING PRECAUTIONS

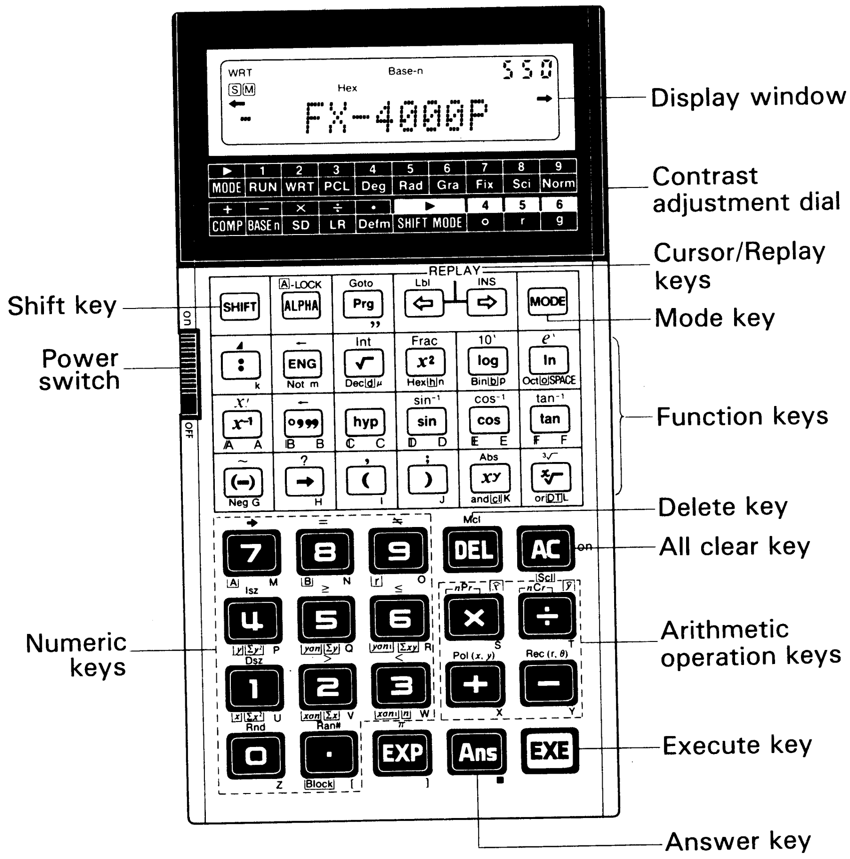
- This unit is composed of precision electronic components, and should never be disassembled. Do not drop it or otherwise subject it to sudden impacts, or sudden temperature changes. Be especially careful to avoid storing the unit or leaving it in areas exposed to high temperature, humidity or large 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.
- A"–"will be displayed when the unit is performing computations. At this time most keys will be inoperative. Therefore, keys should normally be used while confirming proper operation by checking the display.
- Batteries should be replaced every 2 years even if the unit is not used for extended periods. Never leave dead batteries in the battery compartment. They can leak and cause damage to the unit.
- Avoid using volatile liquids such as thinner or benzine to clean the unit. Wipe the unit with a soft, dry cloth or a cloth that has been dipped in a neutral detergent solution and wrung out.
- If malfunction of the unit should occur, either bring or send the unit to your retailer or the nearest CASIO dealer.
  Be sure to clearly explain the problem in detail.
- Before assuming malfunction of the unit, be sure to carefully reread this manual and ensure that the problem is not due to insufficient battery power, programming or operational errors.

1.

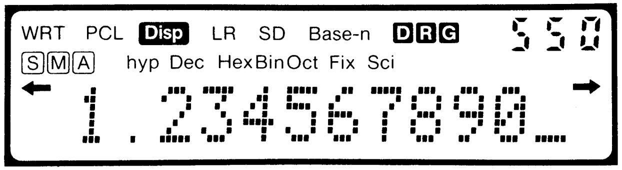
CONFIGURATION AND OPERATION



1-1 NOMENCLATURE AND FUNCTIONS



■ Display window



A "status display"is located at the top of the display window to provide information on the status of the computer. The status display illuminates for each of the operational modes (see page 4).
D, R, G are units of angular measurement and stand for degrees, radians and grads respectively.
S illuminates when the SHIFT key is pressed.
M illuminates when the MODE key is pressed.
A illuminates when the ALPHA key is pressed to put the unit in the alphabet mode.
"Fix"and"Sci" illuminate when specifying the number of decimal places or the number of significant digits.
The three-digit number located in the upper right portion of the display window indicates the characteristic throughout exponent part display. In the program writing mode, it indicates the number of remaining steps.
Formulas and/or computation results are shown at the bottom of the display. Zeros are indicated as 0 to distinguish them from the letter O. The "–"shown in the example (following the zero) is the cursor.
Subsequent inputs of letters or numbers will begin from the point indicated by the cursor.
A maximum of twelve characters can be shown on the display window at one time. The ← and → indicate that the character string being displayed exceeds the capacity of the display window in the respective direction indicated.

■ Power switch

Power is turned ON by sliding the power switch up. Sliding the power switch down turns power OFF.

■ Special operation keys

[SHIFT] Shift key

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

[MODE] Mode key

Press when setting the status of the unit or the unit of angular measurement.

[MODE] [1] ··· For manual computations and program execution.

[MODE] [2] ··· WRT illuminates on display. For writing or checking programs.

[MODE] [3] ··· PCL illuminates on display. For clearing programs.

[MODE] [4] ··· Deg illuminates on display. If [EXE] is pressed, D illuminates on display and unit of angular measurement is specified as degrees.

[MODE] [5] ··· Rad illuminates on display. If [EXE] is pressed, R illuminates on display and unit of angular measurement is specified as radians.

[MODE] [6] ··· Gra illuminates on display. If [EXE] is pressed, G illuminates on display and unit of angular measurement is specified as grads.

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

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

[MODE] [9] ··· Norm displayed. Pressing [EXE] will cancel the specified number of decimal places or the specified number of significant digits.

[MODE] [·] ··· Defm displayed. Entering a value followed by [EXE] will specify the number of memories available.
Ex. [MODE] [·] [1] [0] [EXE] → Number of memories available increased by 10.
If [EXE] is pressed without entering a value, the current number of memories available and remaining steps will be displayed. (See page 22.)
Ex. [MODE] [·] [EXE] M-36 S-470

[MODE] [+] ··· Specifies COMP mode for arithmetic computation or function computation (program execution possible).

[MODE] [−] ··· Base-n illuminates. For binary, octal or hexadecimal computations/conversions.

[MODE] [×] ··· SD illuminates. For standard deviation computations.

[MODE] [÷] ··· LR illuminates. For regression computations.

[SHIFT] [MODE] [4] ··· Pressed after a numeric value representing degrees is input.

[SHIFT] [MODE] [5] ··· Pressed after a numeric value representing radians is input.

[SHIFT] [MODE] [6] ··· Pressed after a numeric value representing grads is input.

[ALPHA] Alphabet key

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

		”			
k	m	µ	n	p	SPACE
A	B	C	D	E	F
G	H	I	J	K	L
M	N	O			
P	Q	R	S	T	
U	V	W	X	Y	
Z	[	]	■		

[Prg] Program/Goto key

Press [Prg] , enter a value from 0 to 9 and then press [EXE] to execute a program.
Ex. [Prg] [1] [EXE] → Execution of Program 1 begins.
Pressing [SHIFT] followed by [Goto] ([Prg] key) will cause Goto to appear on the display. This is a jump command used in programs.

$\overset{\text{Lbl}}{⇦}$ $\overset{\text{REPLAY}}{}$ $\overset{\text{INS}}{⇨}$ Cursor/Replay/Label/Insert 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 the cursor 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.
Re-execution can also be performed after changes have been made in the formula or numeric value. (See page 32.)
Pressing SHIFT followed by Lbl ($\overset{\text{Lbl}}{⇦}$ key) will input a label into the program, while pressing SHIFT followed by INS ($\overset{\text{INS}}{⇨}$ key) will insert a space at the current position of the cursor.

$\overset{\text{Mcl}}{\text{DEL}}$ Delete key

Press to delete the character at the current position of the cursor. When the character is deleted, everything to the right of the cursor position will shift one space to the left.
Pressing SHIFT $\overset{\text{Mcl}}{\text{DEL}}$ EXE will clear the memory contents.

AC All clear key

Press to completely clear the formula or numeric value displayed. Also used to clear errors indicated by error message displays, and to restore power after activation of the auto power off function. (See page 24.)

EXE Execute key

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

Ans Answer key

Pressing Ans followed by EXE will recall the last computation result. It can be recalled by Ans EXE even after it has been cleared using the AC key or by switching the power of the unit OFF. When used during program execution, the last result computed is recalled.

■ 1 ~ 9, · Numeric/Decimal point keys

When entering numeric values, enter the number in order. Press the · key to enter the decimal point in the desired position.
SHIFT key combinations for the various modes are as follows:

COMP mode (MODE +)

➡	=	≠
Isz	≥	≤
Dsz	>	<
Rnd	Ran#	π

Base-n mode (MODE −)

➡	=	≠
Isz	≥	≤
Dsz	>	<

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

SD mode(MODE ×)

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

Standard deviation functions can be used.

LR mode(MODE ÷)

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

Paired variable statistic functions can be used.

■ Computation keys

+ − × ÷ Arithmetic operation keys

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

COMP mode or SD mode

*n*Pr *n*Cr ($\overset{nPr}{\times}$ and $\overset{nCr}{\div}$ keys)···Permutation/combination
Pol(Rec(($\overset{\text{Pol}(x,y)}{+}$ and $\overset{\text{Rec}(r,\theta)}{-}$ keys)···Coordinate transformation

LR mode

$\hat{x}$ $\hat{y}$ ($\overset{\hat{x}}{\times}$ $\overset{\hat{y}}{\div}$ keys)···Estimated value computation of x and y
Pol(Rec(···Coordinate tarnsformation

■ Function keys

Press for functional computation. Various uses are available in combination with the SHIFT key, and/or depending on the mode being used.

: Multistatement/Display key

- Press to separate formulas or commands in programmed computations or consecutive computations.
  The result of such combinations is known as a multistatement. (See page 33.)
- When pressed following the SHIFT key, the results of each section of the programmed computations or consecutive computations are sequentially displayed with each press of EXE .

←
ENG Engineering/Negation key
Not m

- Press to convert a computation result to an exponential display whose exponent is a multiple of three. ($10^3=\overset{\text{kilo}}{K}$, $10^6=\overset{\text{mega}}{M}$, $10^9=\overset{\text{giga}}{G}$, $10^{-3}=\overset{\text{milli}}{m}$, $10^{-6}=\overset{\text{micro}}{\mu}$, $10^{-9}=\overset{\text{nano}}{n}$, $10^{-12}=\overset{\text{pico}}{p}$)
- When obtaining logical negation for a value in the Base–n mode, press prior to entering the value.

Int
√ Root/Integer key
Dec d μ

- Press prior to entering a numeric value to obtain the square root of that value.
- When pressed following the SHIFT key, the integer portion of a value can be obtained.
- Press followed by EXE in the Base-n mode to specify the decimal computation mode.
- When pressed following the SHIFT key in the Base-n mode, the subsequently entered value is specified as a decimal value.

Frac
x^2 Square/Fraction key
Hex h n

- Press after a numeric value is entered to obtain the square of that value.
- When pressed following the SHIFT key, the decimal portion of a value can be obtained.
- Press followed by EXE in the Base-n mode to specify the hexadecimal computation mode.
- When pressed following the SHIFT key in the Base-n mode, the subsequently entered value is specified as a hexadecimal value.

10^x
log Common logarithm/Antilogarithm key
Bin b P

- Press prior to entering a value to obtain the common logarithm of that value.
- When pressed following the SHIFT key, the subsequently entered value becomes an exponent of 10.
- Press followed by EXE in the Base-n mode to specify the binary computation mode.
- When pressed following the SHIFT key in the Base-n mode, the subsequently entered value is specified as a binary value.

e^x
ln Natural logarithm/Anti-natural logarithm key
Oct o SPACE

- Press prior to entering a value to obtain the natural logarithm of that value.
- When pressed following the SHIFT key, the subsequently entered value becomes an exponent of e.
- Press followed by EXE in the Base-n mode to specify the octal computation mode.
- When pressed following the SHIFT key in the Base-n mode, the subsequently entered value is specified as an octal value.

$x!$
x^{-1} Reciprocal/Factorial key
/A A

- Press prior to entering a value to obtain the reciprocal of that value.
- When pressed following the SHIFT key, the factorial of a previously entered value can be obtained.
- Press in the Base-n mode to enter A (10_{10}) of a hexadecimal value.

←
° ' " Degree/minute/second key (decimal ↔sexagesimal key)
IB B

- Press to enter sexagesimal value.
  (degree/minute/second or hour/minute/second)
  Ex. 78°45' 12"→78 ° ' " 45 ° ' " 12 ° ' "
- When pressed following the SHIFT key, a decimal based value can be displayed in degrees/minutes/seconds. (hours/minutes/seconds).
- Press in the Base-n mode to enter B (11_{10}) of a hexadecimal value.

hyp Hyperbolic key
© C

- 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 $\boxed{\text{SHIFT}}$, then $\boxed{\text{hyp}}$ and then $\boxed{\sin}$, $\boxed{\cos}$, or $\boxed{\tan}$ prior to entering a value produces the respective inverse hyperbolic function ($\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$) for the value.
- Press in the Base-n mode to enter C (12_{10}) of a hexadecimal value.

$\overset{\sin^{-1}}{\boxed{\sin}}_{D}$ $\overset{\cos^{-1}}{\boxed{\cos}}_{E}$ $\overset{\tan^{-1}}{\boxed{\tan}}_{F}$ Trigonometric function/Inverse trigonometric function keys

- Press one of these keys prior to entering a value to obtain the respective trigonometric function for the value.
- Press $\boxed{\text{SHIFT}}$ and then one of these keys prior to entering a value to obtain the respective inverse trigonometric function for the value.
- Press in the Base-n mode to enter D, E, F(13_{10}, 14_{10}, 15_{10}) of a hexadecimal value.

$\overset{\sim}{\boxed{(-)}}_{G}$ Minus key

- Press prior to entering a numeric value to make that value negative.
  Ex. $-123 \rightarrow \boxed{(-)}\ \boxed{1}\ \boxed{2}\ \boxed{3}$
- When pressed following the $\boxed{\text{SHIFT}}$ key, the same numeric value can be assigned to multiple memories.
  Ex. To assign the value 456 to memories A through F: $\boxed{4}\ \boxed{5}\ \boxed{6}$ $\boxed{\rightarrow}\ \boxed{\text{ALPHA}}\ \boxed{\text{A}}\ \boxed{\text{SHIFT}}\ \boxed{\sim}\ \boxed{\text{ALPHA}}\ \boxed{\text{F}}\ \boxed{\text{EXE}}$
- Press in the Base-n mode prior to entering a value to obtain the negative of that value. The negative number is the twos complement of the value entered.

$\overset{?}{\boxed{\rightarrow}}_{H}$ Assignment key

- Press prior to entering a memory to assign the result of a computation to that memory.
  Ex. To assign the result of 12 + 45 to memory A: $\boxed{1}\ \boxed{2}\ \boxed{+}\ \boxed{4}$ $\boxed{5}\ \boxed{\rightarrow}\ \boxed{\text{ALPHA}}\ \boxed{\text{A}}\ \boxed{\text{EXE}}$
- During execution of program computations or consecutive computations, press following the $\boxed{\text{SHIFT}}$ key to enter a numeric value.

$\overset{,}{\boxed{(}}_{I}$ $\overset{;}{\boxed{)}}_{J}$ Parenthesis keys

- Press the open parenthesis key and the closed parenthesis key at the position required in a formula.
- When pressed following the $\boxed{\text{SHIFT}}$ key, a comma or semicolon can be inserted to separate the arguments in coordinate transformation or consecutive computations.

$\overset{\text{Abs}}{\boxed{x^y}}$ Power/Absolute value key

and $\boxed{\text{Cl}}$ K

- Enter x (any number), press this key and then enter y (any number) to compute x to the power of y.
  In the SD or LR mode, this function is only available after pressing the $\boxed{\text{SHIFT}}$ key.
- Press following the $\boxed{\text{SHIFT}}$ key to obtain the absolute value of a subsequently entered numeric value.
- Press in the Base-n mode to obtain a logical product ("and").
- Press in the SD or LR mode to delete input data.

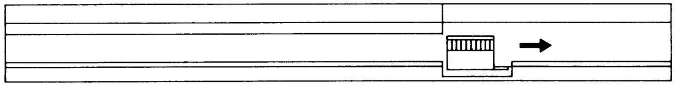
$\overset{\sqrt[3]{}}{\boxed{\sqrt[x]{}}}$ Root/Cube root key

or $\boxed{\text{DT}}$ L

- Enter x, press this key and then enter y to compute the xth root of y. In the SD or LR mode, this function is only available after pressing the $\boxed{\text{SHIFT}}$ key.
- Press following the $\boxed{\text{SHIFT}}$ key to obtain the cube root of a subsequently entered numeric value.
- Press in the Base-n mode to obtain a logical sum ("or").
- Used as a data input key in the SD or LR mode.

■ Contrast adjustment dial

From different viewing angles or due to weakening batteries, the display may sometimes appear too dark or too light. If this should occur, adjust the contrast of the display using the dial located on the right edge of the unit.



Moving the dial in the direction indicated by the arrow will cause the characters on the display to become stronger, while moving it in the opposite direction will cause them to become weaker. If, after maximum adjustment, the characters on the display still are too weak, it can indicate that battery power is too low.
In this case the batteries should be replaced as soon as possible.

1-2 POWER AND BATTERY REPLACEMENT

Power is supplied to this unit by two lithium batteries (CR2032). If the power of the batteries should diminish, the display will weaken and become difficult to read. A weak display even after contrast adjustment (see page 11) may indicate power is too low, so the batteries should be replaced. When making replacements, be sure to replace both batteries.

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

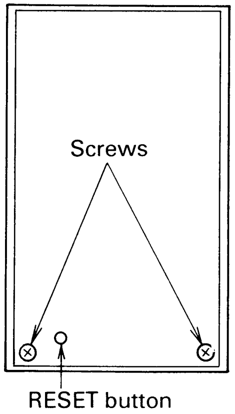
* *Stored programs or data are erased when batteries are replaced. Therefore, it is recommended that programs and data required for later use be recorded on a coding sheet before replacing batteries.*

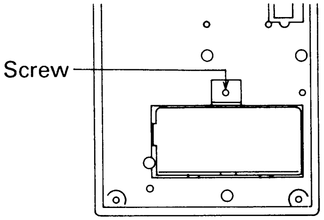
■ Procedure

① Slide the power switch to the OFF position, remove the two screws on the back of the unit with a screwdriver, and remove the back cover.

② Remove the screw holding the battery pressure plate in place and then remove the battery pressure plate.

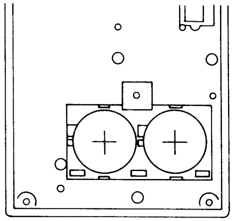
③ Remove both of the old batteries from the unit.
(This can be done easily by turning the unit so the battery compartment is facing downwards, and then lightly tapping the unit.)





④ Wipe the surfaces of two new batteries with a soft, dry cloth and load them into the unit ensuring that the positive ⊕ sides are facing upwards.

⑤ Fasten the battery pressure plate in place using the screw, and replace the back cover.



* *IMPORTANT: Never dispose of old batteries in such a way that they will be incinerated. Batteries may explode if exposed to fire.*

CAUTIONS:

If the batteries being replaced are not totally without power, it is possible to replace batteries so quickly that previously stored programs and memory contents are not erased or altered. In this case, however, all programs and memory contents should be carefully checked after battery replacement.

If battery power should be allowed to decrease or if batteries are removed from the unit for extended periods, programs and memory contents may be erased or altered. In this case, the RESET button located on the back of the unit should be pressed using pointed object.

All memory contents and programs will be erased.

Keep batteries out of the reach of small children. If a battery should inadvertantly be swallowed, contact a doctor immediately.

1-3 BEFORE BEGINNING COMPUTATIONS......

■ Computation priority sequence

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

1. Coordinate transformation Pol(x,y), Rec(r,θ)
2. Type A functions★ x^2, x^{-1}, $x!$, $^\circ$, ^{r}, ^{g}, °’’’
3. Power/root x^y, $\sqrt[x]{\ }$
4. Abbreviated multiplication format in front of π, memory, or parenthesis 2π, 4R, 3(5+6), etc.
5. Type B functions★ $\sqrt{\ }$, $\sqrt[3]{\ }$, log, 10^x, ln, e^x, sin, cos, tan, sin^{-1}, cos^{-1}, tan^{-1}, sinh, cosh, tanh, sinh^{-1}, cosh^{-1}, tanh^{-1}, (−), Abs, Int, Frac, h, d, b, o, Neg, Not
6. Abbreviated multiplication format in front of Type B functions 3sin5, 6$\sqrt{7}$, 2sin30cos60, etc.
7. Permutation, combination nPr, nCr
8. ×, ÷
9. +, −
10. and
11. or
12. Relational operators <, >, =, ≠, ≤, ≥

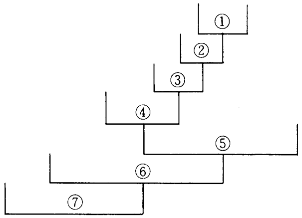
★ Functions are divided into two types.
Type A functions are entered after the argument, while Type B functions are entered before the argument.

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

* *Compound functions are executed from right to left :*
e.g., sin cos^{-1}0.6 → sin(cos^{-1}0.6).

* *Everything contained within parentheses receives highest priority.*

Ex. 2+3×(log sin2π^2rad+6.8)=22.07101691

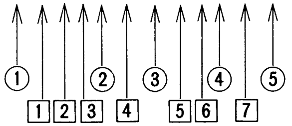


■ Number of stacks

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 eight levels, while the command stack has twenty. If a complex formula is employed that exceeds the stack space available, a stack error (Stk ERROR) message will appear on the display.

Ex. Stack counting method

2 × ((3 + 4 × (5 + 4) ÷ 3) ÷ 5) + 8 =



Numeric value stack

①	2
②	3
③	4
④	5
⑤	4
⋮	

Command stack

$\boxed{1}$	×
$\boxed{2}$	(
$\boxed{3}$	(
$\boxed{4}$	+
$\boxed{5}$	×
$\boxed{6}$	(
$\boxed{7}$	+
⋮	

* *Computations are performed in the order of the highest computation priority first. Once a computation is executed, it is cleared from the stack.*

Computation modes

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

● Operation modes

There are a total of three operation modes.

1. RUN mode
   Manual computations including functional computation and program execution.
2. WRT mode
   Program storage and editing. (See Section 2.) In this mode, WRT illuminates at the left edge of the display window.
3. PCL mode
   Deletion of stored programs. (See Section 2.) In this mode, PCL illuminates at the left edge of the display window.

● Computation modes

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

1. Comp mode
   General computations, including functional computations.
2. Base-n mode
   Binary, octal, decimal, hexadecimal conversion and computations, as well as logical operations. (See page 42.) Function computations can not be performed. In this mode, Base-n illuminates at the center top of the display window.
3. SD mode
   Standard deviation computation (1-variable statistics). (See page 46.) In this mode, SD illuminates at the center top of the display window.
4. LR mode
   Regression computation (paired variable statistics). (See page 48.) In this mode, LR illuminates at the center top of the display.

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

Number of input/output digits and computation digits

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

Ex. $3 \times 10^5 \div 7 =$

3 $\boxed{\text{EXP}}$ 5 $\boxed{\div}$ 7 $\boxed{\text{EXE}}$	42857.14286
3 $\boxed{\text{EXP}}$ 5 $\boxed{\div}$ 7 $\boxed{-}$ 42857 $\boxed{\text{EXE}}$	0.1428571

* *Computation results greater than 10^{10} (10 billion) or less than 10^{-1} (0.1) are automatically displayed in exponential form.*

Ex. $123456789 \times 9638 =$

123456789 $\boxed{\times}$ 9638 $\boxed{\text{EXE}}$	1.189876532^{12}

←Exponent

Mantissa

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

Ex. $3 \times 10^5 \div 7 =$

3 $\boxed{\text{EXP}}$ 5 $\boxed{\div}$ 7 $\boxed{\text{EXE}}$	42857.14286
$\boxed{-}$ 42857 $\boxed{\text{EXE}}$	0.14286

Overflow and errors

If the computational range of the unit is exceeded, or incorrect inputs are made, an error message will appear on the display window and subsequent operation will be impossible. This is the error check function. The following operations will result in errors:

(1) The answer, whether intermediate or final, or any value in memory exceeds the value of $\pm 9.999999999 \times 10^{99}$.

(2) An attempt is made to perform functional computations that exceed the input range. (See page 124.)

(3) Improper operation during statistical computations.
(Ex. Attempting to obtain $\bar{x}$ or $x\sigma_n$ without data input.)

(4) The capacity of the numeric value stack or the command stack is exceeded.
(Ex. Entering nineteen successive $\boxed{(}$'s followed by $\boxed{2}$ $\boxed{+}$ $\boxed{3}$ $\boxed{\times}$ $\boxed{4}$)

(5) Even though memory has not been expanded, a memory name such as Z [2] is used. (See page 20 for details on memory.)

(6) Input errors are made.
(Ex. $\boxed{5}\ \boxed{+}\ \boxed{+}\ \boxed{3}\ \boxed{\text{EXE}}$)

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

(1) ~ (3) Ma ERROR
(4) Stk ERROR
(5) Mem ERROR
(6) Syn ERROR

Besides these, there are an "Ne ERROR" (nesting error) and "Go ERROR". These errors mainly occur when using programs. See page 61 or the Error Message Table on page 122.

■ Number of input characters

This unit features a 79-step area for computation execution.
One function comprises one step. Each press of numeric or $\boxed{+}$, $\boxed{-}$, $\boxed{\times}$ and $\boxed{\div}$ keys comprise one step. Though such operations as $\boxed{\text{SHIFT}}\ \boxed{x!}$ ($\boxed{x^{-1}}$ 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 $\boxed{\Leftarrow}$ or $\boxed{\Rightarrow}$ key the cursor is moved one step.
Input characters are limited to 79 steps. Usually the cursor is represented by a blinking "_", but once the 73rd step is reached the cursor changes to a blinking "■". If the "■" appears during a computation, the computation should be divided at some point and performed in two parts.

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

■ Corrections

- To make corrections in a formula that is being input, use the $\boxed{\Leftarrow}$ and $\boxed{\Rightarrow}$ keys to move to the position of the error and press the correct keys.

Ex. To change an input of 122 to 123:

$\boxed{1}\ \boxed{2}\ \boxed{2}$	122_
$\boxed{\Leftarrow}$	$12\underline{2}$
$\boxed{3}$	123_

Ex. To change an input of cos60 to sin60 :

$\boxed{\cos}\ \boxed{6}\ \boxed{0}$	cos 60_
$\boxed{\Leftarrow}\ \boxed{\Leftarrow}\ \boxed{\Leftarrow}$	$\underline{c}$os 60
$\boxed{\sin}$	sin $\underline{6}$0

* *If, after making corrections, input of the formula is complete, the answer can be obtained by pressing* $\boxed{\text{EXE}}$. *If, however, more is to be added to the formula, advance the cursor using the* $\boxed{\Rightarrow}$ *key to the end of the formula for input.*

- If an unnecessary character has been included in a formula, use the $\boxed{\Leftarrow}$ and $\boxed{\Rightarrow}$ keys to move to the position of the error and press the $\boxed{\text{DEL}}$ key. Each press of $\boxed{\text{DEL}}$ will delete one command (one step).

Ex. To correct an input of 369××2 to 369×2:

$\boxed{3}\ \boxed{6}\ \boxed{9}\ \boxed{\times}\ \boxed{\times}\ \boxed{2}$	369××2_
$\boxed{\Leftarrow}\ \boxed{\Leftarrow}\ \boxed{\text{DEL}}$	369×$\underline{2}$

- If a character has been omitted from a formula, use the $\boxed{\Leftarrow}$ and $\boxed{\Rightarrow}$ keys to move to the position where the character should have been input, and press $\boxed{\text{SHIFT}}$ followed by the $\boxed{\text{INS}}$ key. Each press of the $\boxed{\text{SHIFT}}$ $\boxed{\text{INS}}$ will create a space for input of one command.

Ex. To correct an input of 2.36^2 to $\sin 2.36^2$:

$\boxed{2}$ $\boxed{\cdot}$ $\boxed{3}$ $\boxed{6}$ $\boxed{x^2}$	2.36^2_
$\boxed{\Leftarrow}$ $\boxed{\Leftarrow}$ $\boxed{\Leftarrow}$ $\boxed{\Leftarrow}$ $\boxed{\Leftarrow}$	$\underline{2}.36^2$
$\boxed{\text{SHIFT}}$ $\boxed{\text{INS}}$	⌜⌟2.36^2
$\boxed{\sin}$	sin $\underline{2}.36^2$

** When $\boxed{\text{SHIFT}}$ $\boxed{\text{INS}}$ are pressed, the space that is opened is displayed as "⌜⌟". Though the ⌜⌟ will remain within the formula even if nothing is inserted, it is disregarded during computations, and pressing $\boxed{\text{EXE}}$ will execute the formula normally. Use the $\boxed{\text{DEL}}$ key to delete a ⌜⌟ from a formula.*

■ Memory

This unit contains 26 standard memories. Memory names are composed of the 26 letters of the alphabet. Numeric values with 10 digits for a mantissa and 2 digits for an exponent can be stored.

Ex. To store 123.45 in memory A:

123.45 $\boxed{\rightarrow}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$	123.45→A_
$\boxed{\text{EXE}}$	123.45

Values are assigned to a memory using the $\boxed{\rightarrow}$ key followed by the memory name.

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

$\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{+}$ 78.9 $\boxed{\rightarrow}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{B}}$	A+78.9→B_
$\boxed{\text{EXE}}$	202.35

Ex. To add 74.12 to memory B:

$\boxed{\text{ALPHA}}$ $\boxed{\text{B}}$ $\boxed{+}$ 74.12 $\boxed{\rightarrow}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{B}}$	B+74.12→B_
$\boxed{\text{EXE}}$	276.47

** Numeric values with 10 digits for a mantissa and 2 digits for an exponent can be stored in memory. Therefore, consecutive computations using numeric values stored in memory are less precise when compared with internal computaions (12 digits for a mantissa and 2 digits for an exponent).*

- To check the contents of a memory, press the name of the memory to be checked followed by $\boxed{\text{EXE}}$.

$\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{\text{EXE}}$	123.45

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

Ex. To clear the contents of memory A only:

0 $\boxed{\rightarrow}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{\text{EXE}}$	0.

Ex. To clear the contents of all the memories:

$\boxed{\text{SHIFT}}$ $\boxed{\text{Mcl}}$	Mcl_
$\boxed{\text{EXE}}$	0.

- To store the same numeric value to multiple memories, press $\boxed{\text{SHIFT}}$ followed by $\boxed{\sim}$ ($\boxed{(-)}$ key).

Ex. To store a value of 10 in memories A through J:

10 $\boxed{\rightarrow}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{\text{SHIFT}}$ $\boxed{\sim}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{J}}$	10→A~J_
$\boxed{\text{EXE}}$	10.

■ Memory expansion

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

* *See page 64 for information on the number of program steps.*

Number of memories	26	27	28	…	36	…	76	…	94
Number of steps	550	542	534	…	470	…	150	…	6

Memory is expanded in units of one. A maximum of 68 memories can be added for a maximum total of 94 (26 + 68). Expansion is performed by pressing MODE , followed by · , a value representing the size of the expansion, and then EXE .

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

MODE · 30	Defm 30_
EXE	M−56 S−310

Number of memories

Current number of remaining steps

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

MODE · EXE	M−56 S−310

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

MODE · 0 EXE	M−26 S−550

* *Though a maximum of 68 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 (Syn ERROR) will be generated. The size of the memory expansion must be equal to or less than the number of steps remaining.*

* *The expansion procedure (MODE · expansion value) can also be stored as a program.*

● Using expanded memories

Expanded memories are used in the same manner as standard memories, and are referred to as Z[1], Z[2], etc. The letter Z followed by a value in brackets indicating the sequential position of the memory is used as the memory name. (Brackets are formed by ALPHA · for "[" and ALPHA EXP for "]".) After the number of memories has been expanded by 5, memories Z[1] through Z[5] are available.
The use of these memories is similar to that of a standard computer array, with a subscript being appended to the name. For more information concerning an array, see page 79.

■ Answer (Ans) function

This unit has an answer function that stores the result of the most recent computation. Once a numeric value or numeric formula is entered and EXE is pressed, the result (the answer in the case of the numeric formula) 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 can be used in this form in subsequent calculations.

* *Hereinafter, Ans will be referred to as the Ans memory.*

Ex. 123+456=579
789−579=210

1 2 3 + 4 5 6 EXE	579.
7 8 9 − Ans	789−Ans_
EXE	210.

Numeric values with 10 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 switched OFF. Each time EXE is pressed, the value in the Ans memory is replaced with the new value produced by the computation executed.
When a value is stored to another memory using the EXE key, that value is not stored in the Ans memory.

Ex. Perform computation 78+56=134, then store the value 123 to memory A:

7 8 + 5 6 EXE	134.
Ans EXE … Confirm Ans memory contents.	134.
1 2 3 → ALPHA A EXE	123.
Ans EXE	134.

The Ans memory can be used in the same manner as the other memories, thus making it possible to use it in computation formulas. In multiplication operations, the $\boxed{\times}$ immediately before $\boxed{\text{Ans}}$ can be omitted.

Ex. $15\times3=\underline{45}$

$78\times\underline{45}-23=3487$

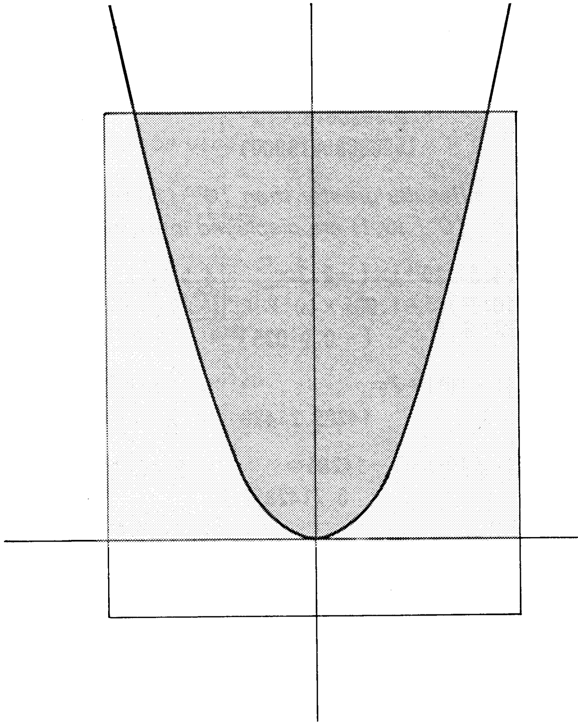
$\boxed{1}\ \boxed{5}\ \boxed{\times}\ \boxed{3}\ \boxed{\text{EXE}}$	45.
$\boxed{7}\ \boxed{8}\ \boxed{\text{Ans}}\ \boxed{-}\ \boxed{2}\ \boxed{3}\ \boxed{\text{EXE}}$	3487.

■ Auto power off function

The power of the unit is automatically switched off approximately 6 minutes after the last key operation (except during program computations). Once this occurs, power can be restored either by switching the power of the unit OFF and then ON again, or by pressing the $\boxed{\text{AC}}$ key. (Numeric values in the memories, programs or computation modes are unaffected when power is switched off.)

2.

MANUAL COMPUTATIONS



2-1 BASIC COMPUTATIONS

■ Arithmetic operations

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

Example	Operation	Display
$23+4.5-53=-25.5$	$23\ \boxed{+}\ 4.5\ \boxed{-}\ 53\ \boxed{EXE}$	−25.5
$56\times(-12)\div(-2.5)=$ 268.8	$56\ \boxed{\times}\ \boxed{(-)}\ 12\ \boxed{\div}\ \boxed{(-)}\ 2.5\ \boxed{EXE}$	268.8
$12369\times7532\times74103=$ 6.903680613×10^{12} (6903680613000)	$12369\ \boxed{\times}\ 7532\ \boxed{\times}\ 74103\ \boxed{EXE}$	6.903680613^{12}
∗Results greater than 10^{10} (10 billion) or less than 10^{-1} (0.1) are displayed in exponential form.		
$(4.5\times10^{75})\times(-2.3\times$ $10^{-78})=-1.035\times10^{-2}$ (-0.01035)	$4.5\ \boxed{EXP}\ 75\ \boxed{\times}\ \boxed{(-)}\ 2.3\ \boxed{EXP}$ $\boxed{(-)}\ 78\ \boxed{EXE}$	-1.035^{-02}
$(1\times10^{5})\div7=$ 14285.71429	$1\ \boxed{EXP}\ 5\ \boxed{\div}\ 7\ \boxed{EXE}$	14285.71429
$(1\times10^{5})\div7-14285=$ 0.7142857	$1\ \boxed{EXP}\ 5\ \boxed{\div}\ 7\ \boxed{-}\ 14285\ \boxed{EXE}$	0.7142857
∗Internal computations are computed in 12 digits for a mantissa, and the result is displayed rounded off to 10 digits.		

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

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

■ Parenthesis computations

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

■ Memory computations

- ●Numeric values stored in memory contain 10 digits for a mantissa.
- ●The contents of memories are not erased when power is switched OFF. They are cleared by pressing SHIFT followed by Mcl ($\overset{Mcl}{DEL}$ key) and then EXE .

Example	Operation	Display
9.874×7=69.118	9.874 → ALPHA A EXE	9.874
9.874×12=118.488	ALPHA A × 7 EXE	69.118
9.874×26=256.724	ALPHA A × 12 EXE	118.488
9.874×29=286.346	ALPHA A × 26 EXE ALPHA A × 29 EXE	256.724 286.346
∗The → key is used to input numeric values in memory. (Clearing a memory before input is not required, because the previous value in the memory will be automatically replaced with the new value.)		
23+9=32	23 + 9 → ALPHA B EXE	32.
53−6=47	53 − 6 EXE	47.
−)45×2=90	ALPHA B + Ans → ALPHA B EXE	79.
99÷3=33	45 × 2 EXE	90.
Total 22	ALPHA B − Ans → ALPHA B EXE	−11.
	99 ÷ 3 EXE	33.
	ALPHA B + Ans → ALPHA B EXE	22.
$12\times(\underline{2.3+3.4})-5=63.4$	2.3 + 3.4 → ALPHA G EXE 12 × ALPHA G − 5 EXE	5.7 63.4
$30\times(\underline{2.3+3.4}+\underline{4.5})-15\times\underline{4.5}=238.5$	4.5 → ALPHA H EXE 30 × (ALPHA G + ALPHA H) − 15 ALPHA H EXE	4.5 238.5
∗Multiplication signs (×) immediately before memory names can be omitted.		

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

- ●To specify the number of decimal places, press MODE followed by 7 , a value indicating the number of places (0−9) and then EXE. (The indicator "Fix" will illuminate on the display.)
- ●To specify the number of significant digits, press MODE followed by 8 , a value indicating the number of significant digits (0−9 to set from 1 to 10 digits) and then EXE . (The indicator "Sci" will illuminate on the display.)
- ●Pressing the ENG key or SHIFT followed by ← ($\overset{\leftarrow}{ENG}$ key) will cause the exponent display for the number being displayed to change in multiples of 3.
- ●The specified number of decimal places or number of significant digits will not be cancelled until another value or MODE 9 is specified using the sequence: MODE, 9, EXE. (Specified values are not cancelled even if power is switched OFF or an other mode (besides MODE 9) is specified.)
- ●Even if the number of decimal places and number of significant digits are specified, internal computations are performed in 12 digits for a mantissa, and the displayed value is stored in 10 digits. To convert these values to the specified number of decimal places and significant digits, press SHIFT followed by Rnd ($\overset{Rnd}{0}$ key) and then EXE .

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

Example	Operation	Display
100÷6=16.66666666···	100 ÷ 6 EXE	16.66666667
	MODE 7 4 EXE (Four decimal places specified.)	16.6667
	MODE 9 EXE (Specification cancelled.)	16.66666667
	MODE 8 5 EXE (Five significant digits specified.)	1.6667^{01}
	MODE 9 EXE (Specification cancelled.)	16.66666667
∗ Values are displayed rounded off to the place specified.		

200÷7×14=400	[MODE] [7] [3] [EXE] (Three decimal places specified.)	16.667
	200 [÷] 7 [EXE]	28.571
(Continues computation with 10-digit display.)	[×]	8.57142857×_
	14 [EXE]	400.000
	If the same computation is performed with the specified number of digits:	
	200 [÷] 7 [EXE]	28.571
	(Value stored internally cut off at specified decimal place.) [SHIFT] [Rnd] [EXE]	28.571
	[×]	28.571×_
	14 [EXE]	399.994
	[MODE] [9] [EXE] (Specification cancelled.)	399.994
123 m×456=56088 m	123 [×] 456 [EXE]	56088.
=56.088km	[ENG]	56.088^{03}
78 g×0.96=74.88 g	78 [×] 0.96 [EXE]	74.88
=0.07488kg	[SHIFT] [$\overset{\leftarrow}{\text{ENG}}$]	0.07488^{03}

2-2 SPECIAL FUNCTIONS

■ Continuous computation function

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

Ex. 3×4=12 Continuing ÷3.14=

3 [×] 4 [EXE]	12.
(Continuing) [÷] 3.14	12.÷3.14_
[EXE]	3.821656051

Ex. To compute 1÷3×3

1 [÷] 3 [×] 3 [EXE]	1.
1 [÷] 3 [EXE]	0.3333333333
(Continuing) [×] 3 [EXE]	0.9999999999

This function can be used with memory and Type A functions (x^2, x^{-1}, $x!$: see page 40), and +, −, $_nP_r$, $_nC_r$, x^y, $\sqrt[x]{\ }$, °' ".

Ex. To store the result of 12×45 in memory C:

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

Ex. To square the result of 78÷6 (see page 40):

78 [÷] 6 [EXE]	13.
(Continuing) [x^2]	$13.^2$_
[EXE]	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.
  Then using ⇨ and ⇦ to move the cursor, the formula can be checked and numeric values or commands can be changed for subsequent execution.

Ex.

123 × 456 EXE	56088.
⇨	$\underline{1}23\times456$
EXE	56088.
⇦	123×456_

Ex. $4.12\times3.58+\underline{6.4}=21.1496$
$4.12\times3.58\underline{-7.1}=7.6496$

4.12 × 3.58 + 6.4 EXE	21.1496
⇦	12×3.58+6.4_
⇦ ⇦ ⇦ ⇦	$4.12\times3.58\underline{+}6.$
− 7.1	12×3.58−7.1_
EXE	7.6496

- If an error is generated during computation execution, an error check function eliminates the need to clear the error using AC and then restarting input from the beginning. Pressing either ⇨ or ⇦ will automatically move the cursor to the point in the formula that generated the error and display it.

Ex. When 14÷0×2.3 is mistakenly entered for 14÷10×2.3:

14 ÷ 0 × 2.3 EXE	Ma ERROR
⇨ (or ⇦)	$14\div0\underline{\times}2.3$
	Here an error occurs.
⇦ SHIFT INS 1	$14\div1\underline{0}\times2.3$
EXE	3.22

∗As with the number of input characters (see page 18), the replay function can accept input up to 79 steps.
∗The replay function is cleared when the AC key is pressed, when power is switched OFF or when the mode is changed.

Multistatement function

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

Ex. 6.9×123=848.7
123÷3.2=38.4375

Illuminated during display caused by "◢".

123 → ALPHA A :	
6.9 × ALPHA A SHIFT ◢	
ALPHA A ÷ 3.2 EXE	Disp 848.7
EXE	38.4375

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

123×456:+5
Invalid

2-3 FUNCTIONAL COMPUTATIONS

Angular measurement units

- The unit of angular measurement (degrees, radians, grads) is set by pressing MODE followed by a value from 4 through 6 and then EXE .
- The numeric value from 4 through 6 specifies degrees, radians and grads respectively.
- Once a unit of angular measurement is set, it remains in effect until a new unit is set. Settings are not cleared when power is switched OFF.
- You cannot specify the unit of angular measurement (degrees, radians, grads) while the calculator is in the Base-n mode. Such specifications can only be made if you first exit the Base-n mode.

Example	Operation	Display
Conversion of 4.25 rad to degrees	MODE 4 EXE →"D" 4.25 SHIFT MODE 5 EXE	243.5070629
Conversion of 1.23 grad. to radians	MODE 5 EXE →"R" 1.23 SHIFT MODE 6 EXE	1.932079482^{-02}
Conversion of 7.89 degrees to grads	MODE 6 EXE →"G" 7.89 SHIFT MODE 4 EXE	8.766666667
Result displayed in degrees 47.3°+82.5rad= 4774.20181	MODE 4 EXE →"D" 47.3 + 82.5 SHIFT MODE 5 EXE	4774.20181
12.4°+8.3rad−1.8gra= 486.33497	12.4 + 8.3 SHIFT MODE 5 − 1.8 SHIFT MODE 6 EXE	486.33497
Result displayed in radians 24°6'31''+85.34rad= 85.76077464	MODE 5 EXE →"R" 24 °'" 6 °'" 31 °'" SHIFT MODE 4 + 85.34 EXE	85.76077464
Result displayed in grads 36.9°+41.2rad= 2663.873462	MODE 6 EXE →"G" 36.9 SHIFT MODE 4 + 41.2 SHIFT MODE 5 EXE	2663.873462

Trigonometric functions and inverse trigonometric functions

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

Example	Operation	Display
sin63°52'41''= 0.897859012	MODE 4 EXE →"D" sin 63 °'" 52 °'" 41 °'" EXE	0.897859012
$\cos(\frac{\pi}{3}\text{rad})=0.5$	MODE 5 EXE →"R" cos (SHIFT π ÷ 3) EXE	0.5
tan(−35gra)= −0.6128007881	MODE 6 EXE →"G" tan (−) 35 EXE	−0.6128007881
2·sin45°×cos65°= 0.5976724775	MODE 4 EXE →"D" 2 × sin 45 × cos 65 EXE Can be omitted.	0.5976724775
$\sin^{-1}0.5=30°$ (Determine the value of x when $\sin x=0.5$.)	SHIFT $\sin^{-1}$ 0.5 EXE Can be entered as .5	30.
$\cos^{-1}\frac{\sqrt{2}}{2}=$ 0.7853981634rad $=\frac{\pi}{4}$rad	MODE 5 EXE →"R" SHIFT $\cos^{-1}$ (√ 2 ÷ 2) EXE ÷ SHIFT π EXE	0.7853981634 0.25
$\tan^{-1}0.741=$ 36.53844577° =36°32'18.4"	MODE 4 EXE →"D" SHIFT $\tan^{-1}$ 0.741 EXE SHIFT $\overset{\leftarrow}{°'"}$	36.53844577 36°32'18.4"
2.5×($\sin^{-1}$0.8− $\cos^{-1}$0.9)=68°13'13.53"	2.5 × (SHIFT $\sin^{-1}$ 0.8 − SHIFT $\cos^{-1}$ 0.9) EXE SHIFT $\overset{\leftarrow}{°'"}$	68°13'13.53"
sin18°×cos0.25rad= 0.2994104044	sin 18 × cos 0.25 SHIFT MODE 5 EXE	0.2994104044

**If the total number of digits for degrees/minutes/seconds exceeds eight 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.*

**The above is computed in radians, and is the same as* sin 18 SHIFT MODE 4 × cos 0.25 EXE .

Logarithmic and exponential functions

Example	Operation	Display
$\log 1.23 (\log_{10} 1.23) = 8.990511144 \times 10^{-2}$	$\boxed{\text{log}}$ 1.23 $\boxed{\text{EXE}}$	8.990511144^{-02}
$\ln 90 (\log_e 90) = 4.49980967$	$\boxed{\text{ln}}$ 90 $\boxed{\text{EXE}}$	4.49980967
$\log 456 \div \ln 456 = 0.4342944819$ (log/ln ratio=constant M)	$\boxed{\text{log}}$ 456 $\boxed{\div}$ $\boxed{\text{ln}}$ 456 $\boxed{\text{EXE}}$	0.4342944819
$10^{1.23} = 16.98243652$ (To obtain the antilogarithm of common logarithm 1.23)	$\boxed{\text{SHIFT}}$ $\boxed{10^x}$ 1.23 $\boxed{\text{EXE}}$	16.98243652
$e^{4.5} = 90.0171313$ (To obtain the antilogarithm of natural logarithm 4.5)	$\boxed{\text{SHIFT}}$ $\boxed{e^x}$ 4.5 $\boxed{\text{EXE}}$	90.0171313
$10^4 \cdot e^{-4} + 1.2 \cdot 10^{2.3} = 422.5878667$	$\boxed{\text{SHIFT}}$ $\boxed{10^x}$ 4 $\boxed{\times}$ $\boxed{\text{SHIFT}}$ $\boxed{e^x}$ $\boxed{(-)}$ 4 $\boxed{+}$ 1.2 $\boxed{\times}$ $\boxed{\text{SHIFT}}$ $\boxed{10^x}$ 2.3 $\boxed{\text{EXE}}$	422.5878667
$5.6^{2.3} = 52.58143837$	5.6 $\boxed{x^y}$ 2.3 $\boxed{\text{EXE}}$	52.58143837
$\sqrt[7]{123} (= 123^{\frac{1}{7}}) = 1.988647795$	7 $\boxed{\sqrt[x]{\ }}$ 123 $\boxed{\text{EXE}}$	1.988647795
$(78-23)^{-12} = 1.305111829 \times 10^{-21}$	$\boxed{(}$ 78 $\boxed{-}$ 23 $\boxed{)}$ $\boxed{x^y}$ $\boxed{(-)}$ 12 $\boxed{\text{EXE}}$	1.305111829^{-21}
$2 + \underline{3 \times \sqrt[3]{64}} - 4 = 10$	2 $\boxed{+}$ 3 $\boxed{\times}$ 3 $\boxed{\sqrt[x]{\ }}$ 64 $\boxed{-}$ 4 $\boxed{\text{EXE}}$	10.

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

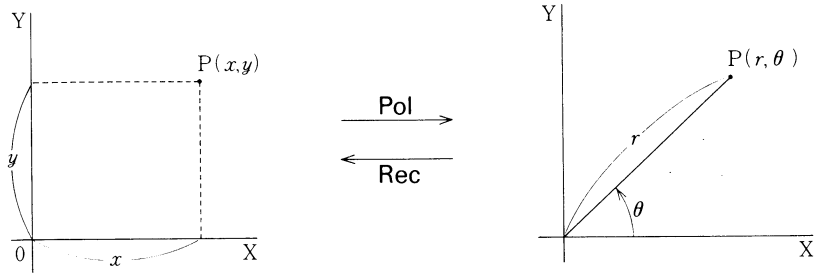
Example	Operation	Display
$2 \times 3.4^{(5+6.7)} = 3306232.001$	2 $\boxed{\times}$ 3.4 $\boxed{x^y}$ $\boxed{(}$ 5 $\boxed{+}$ 6.7 $\boxed{)}$ $\boxed{\text{EXE}}$	3306232.001

Hyperbolic functions and inverse hyperbolic functions

Example	Operation	Display
$\sinh 3.6 = 18.28545536$	$\boxed{\text{hyp}}$ $\boxed{\text{sin}}$ 3.6 $\boxed{\text{EXE}}$	18.28545536
$\cosh 1.23 = 1.856761057$	$\boxed{\text{hyp}}$ $\boxed{\text{cos}}$ 1.23 $\boxed{\text{EXE}}$	1.856761057
$\tanh 2.5 = 0.9866142981$	$\boxed{\text{hyp}}$ $\boxed{\text{tan}}$ 2.5 $\boxed{\text{EXE}}$	0.9866142981
$\cosh 1.5 - \sinh 1.5 = 0.2231301602 = e^{-1.5}$ (Proof of $\cosh x \pm \sinh x = e^{\pm x}$)	$\boxed{\text{hyp}}$ $\boxed{\text{cos}}$ 1.5 $\boxed{-}$ $\boxed{\text{hyp}}$ $\boxed{\text{sin}}$ 1.5 $\boxed{\text{EXE}}$ (Continuing) $\boxed{\text{ln}}$ $\boxed{\text{Ans}}$ $\boxed{\text{EXE}}$	0.2231301602 −1.5
$\sinh^{-1} 30 = 4.094622224$	$\boxed{\text{SHIFT}}$ $\boxed{\text{hyp}}$ $\boxed{\text{sin}^{-1}}$ 30 $\boxed{\text{EXE}}$	4.094622224
$\cosh^{-1}(\frac{20}{15}) = 0.7953654612$	$\boxed{\text{SHIFT}}$ $\boxed{\text{hyp}}$ $\boxed{\text{cos}^{-1}}$ $\boxed{(}$ 20 $\boxed{\div}$ 15 $\boxed{)}$ $\boxed{\text{EXE}}$	0.7953654612
Determine the value of x when $\tanh 4x = 0.88$ $x = \frac{\tanh^{-1} 0.88}{4} = 0.3439419141$	$\boxed{\text{SHIFT}}$ $\boxed{\text{hyp}}$ $\boxed{\text{tan}^{-1}}$ 0.88 $\boxed{\div}$ 4 $\boxed{\text{EXE}}$	0.3439419141
$\sinh^{-1} 2 \times \cosh^{-1} 1.5 = 1.389388923$	$\boxed{\text{SHIFT}}$ $\boxed{\text{hyp}}$ $\boxed{\text{sin}^{-1}}$ 2 $\boxed{\times}$ $\boxed{\text{SHIFT}}$ $\boxed{\text{hyp}}$ $\boxed{\text{cos}}$ 1.5 $\boxed{\text{EXE}}$	1.389388923
$\sinh^{-1}(\frac{2}{3}) + \tanh^{-1}(\frac{4}{5}) = 1.723757406$	$\boxed{\text{SHIFT}}$ $\boxed{\text{hyp}}$ $\boxed{\text{sin}^{-1}}$ $\boxed{(}$ 2 $\boxed{\div}$ 3 $\boxed{)}$ $\boxed{+}$ $\boxed{\text{SHIFT}}$ $\boxed{\text{hyp}}$ $\boxed{\text{tan}^{-1}}$ $\boxed{(}$ 4 $\boxed{\div}$ 5 $\boxed{)}$ $\boxed{\text{EXE}}$	1.723757406

■ Coordinate transformation

● Rectangular coordinates ● Polar coordinates



● Computation results are stored in memories I and J. (Contents of memory I displayed.)

Pol → I $=r$, J $=\theta$
Rec → I $=x$, J $=y$

● With polar coordinates, θ can be computed within a range of $-180° < \theta \leqq 180°$. (The computation range is the same with radians or grads.)

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

■ Permutation and combination

● Total number of permutations

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

● Total number of combinations

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

Example	Operation	Display
Taking any four out of ten items and arranging them in a row, how many different arrangements are possible? $_{10}P_4=5040$	10 $\boxed{\text{SHIFT}}$ $\boxed{nPr}$ 4 $\boxed{\text{EXE}}$	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 ? (3/7 of the total number of permutations will be even.) $_7P_4\times\frac{3}{7}=360$	7 $\boxed{\text{SHIFT}}$ $\boxed{nPr}$ 4 $\boxed{\times}$ 3 $\boxed{\div}$ 7 $\boxed{\text{EXE}}$	360.
If any four items are removed from 10 items, how many different combinations are possible? $_{10}C_4=210$	10 $\boxed{\text{SHIFT}}$ $\boxed{nCr}$ 4 $\boxed{\text{EXE}}$	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. $_{25}C_5-{}_{15}C_5=50127$	25 $\boxed{\text{SHIFT}}$ $\boxed{nCr}$ 5 $\boxed{-}$ 15 $\boxed{\text{SHIFT}}$ $\boxed{nCr}$ 5 $\boxed{\text{EXE}}$	50127.

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

Example	Operation	Display
$\sqrt{2}+\sqrt{5}=3.65028154$	$\boxed{\sqrt{\ }}$ 2 $\boxed{+}$ $\boxed{\sqrt{\ }}$ 5 $\boxed{\text{EXE}}$	3.65028154
$2^2+3^2+4^2+5^2=54$	2 $\boxed{x^2}$ $\boxed{+}$ 3 $\boxed{x^2}$ $\boxed{+}$ 4 $\boxed{x^2}$ $\boxed{+}$ 5 $\boxed{x^2}$ $\boxed{\text{EXE}}$	54.
$\frac{1}{\frac{1}{3}-\frac{1}{4}}=12$	$\boxed{(}$ 3 $\boxed{x^{-1}}$ − 4 $\boxed{x^{-1}}$ $\boxed{)}$ $\boxed{x^{-1}}$ $\boxed{\text{EXE}}$	12.
$8!(=1\times2\times3\times\cdots\cdots\times8)$ $=40320$	8 $\boxed{\text{SHIFT}}$ $\boxed{x!}$ $\boxed{\text{EXE}}$	40320.
$\sqrt[3]{36\times42\times49}=42$	$\boxed{\text{SHIFT}}$ $\boxed{\sqrt[3]{\ }}$ $\boxed{(}$ 36 $\boxed{\times}$ 42 $\boxed{\times}$ 49 $\boxed{)}$ $\boxed{\text{EXE}}$	42.
Random number generation (pseudorandom number from 0.000 to 0.999)	$\boxed{\text{SHIFT}}$ $\boxed{\text{Ran\#}}$ $\boxed{\text{EXE}}$	(Ex) 0.792
$\sqrt{13^2-5^2}+\sqrt{3^2+4^2}=17$	$\boxed{\sqrt{\ }}$ $\boxed{(}$ 13 $\boxed{x^2}$ $\boxed{-}$ 5 $\boxed{x^2}$ $\boxed{)}$ $\boxed{+}$ $\boxed{\sqrt{\ }}$ $\boxed{(}$ 3 $\boxed{x^2}$ $\boxed{+}$ 4 $\boxed{x^2}$ $\boxed{)}$ $\boxed{\text{EXE}}$	17.
$\sqrt{1-\sin^2 40^\circ}=$ $0.7660444431=\cos 40^\circ$	$\boxed{\text{MODE}}$ $\boxed{4}$ $\boxed{\text{EXE}}$ →"D" $\boxed{\sqrt{\ }}$ $\boxed{(}$ 1 $\boxed{-}$ $\boxed{(}$ $\boxed{\text{sin}}$ 40 $\boxed{)}$ $\boxed{x^2}$ $\boxed{)}$ $\boxed{\text{EXE}}$	0.7660444431
(Proof of $\cos\theta=$ $\sqrt{1-\sin^2\theta}$)	(Continuing) $\boxed{\text{SHIFT}}$ $\boxed{\cos^{-1}}$ $\boxed{\text{Ans}}$ $\boxed{\text{EXE}}$	40.
$\frac{1}{2!}+\frac{1}{4!}+\frac{1}{6!}+\frac{1}{8!}=$ 0.5430803571	2 $\boxed{\text{SHIFT}}$ $\boxed{x!}$ $\boxed{x^{-1}}$ $\boxed{+}$ 4 $\boxed{\text{SHIFT}}$ $\boxed{x!}$ $\boxed{x^{-1}}$ $\boxed{+}$ 6 $\boxed{\text{SHIFT}}$ $\boxed{x!}$ $\boxed{x^{-1}}$ $\boxed{+}$ 8 $\boxed{\text{SHIFT}}$ $\boxed{x!}$ $\boxed{x^{-1}}$ $\boxed{\text{EXE}}$	0.5430803571
What is the absolute value of the common logarithm of $\frac{3}{4}$? $\left\lvert\log\frac{3}{4}\right\rvert=0.1249387366$	$\boxed{\text{SHIFT}}$ $\boxed{\text{Abs}}$ $\boxed{\text{log}}$ $\boxed{(}$ 3 $\boxed{\div}$ 4 $\boxed{)}$ $\boxed{\text{EXE}}$	0.1249387366
What is the integer part of $\frac{7800}{96}$?	$\boxed{\text{SHIFT}}$ $\boxed{\text{Int}}$ $\boxed{(}$ 7800 $\boxed{\div}$ 96 $\boxed{)}$ $\boxed{\text{EXE}}$	81.
What is the fraction part of $\frac{7800}{96}$?	$\boxed{\text{SHIFT}}$ $\boxed{\text{Frac}}$ $\boxed{(}$ 7800 $\boxed{\div}$ 96 $\boxed{)}$ $\boxed{\text{EXE}}$	0.25
What is the aliquot part of 2512549139÷2141?	2512549139 $\boxed{\div}$ 2141 $\boxed{\text{EXE}}$ $\boxed{\text{SHIFT}}$ $\boxed{\text{Frac}}$ $\boxed{(}$ 2512549139 $\boxed{\div}$ 2141 $\boxed{)}$ $\boxed{\text{EXE}}$	1173540. 0.99953

2-4 BINARY, OCTAL, DECIMAL, HEXADECIMAL COMPUTATIONS

- Binary, octal, decimal and hexadecimal computations, conversions and logical operations are performed in the Base-n mode (press $\boxed{\text{MODE}}$ $\boxed{-}$)
- The number system (2, 8, 10, 16) is set by respectively pressing $\boxed{\text{Bin}}$, $\boxed{\text{Oct}}$, $\boxed{\text{Dec}}$ or $\boxed{\text{Hex}}$, followed by $\boxed{\text{EXE}}$.
- Number systems are specified for specific values by pressing $\boxed{\text{SHIFT}}$, then the number system designator ($\boxed{\text{b}}$, $\boxed{\text{o}}$, $\boxed{\text{d}}$ or $\boxed{\text{h}}$), immediately followed by the value.
- General function computations cannot be performed in the Base-n mode.
- Only integers can be handled in the Base-n mode. If a computation produces a result that includes a decimal value, the decimal portion is cut off.
- Computations can be handled up to 32 bits.

Binary	Up to 32 digits (displayed by pressing $\boxed{\text{Block}}$ n ($n=1\sim4$) $\boxed{\text{EXE}}$)
Octal	Up to 11 digits
Decimal	Up to 10 digits
Hexadecimal	Up to 8 digits

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

Valid values

Binary	0, 1
Octal	0, 1, 2, 3, 4, 5, 6, 7
Decimal	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

- Negative numbers in binary, octal and hexadecimal are expressed as twos complements.
- To distinguish the A, B, C, D, E, F used in the hexadecimal system from standard letters they appear as: 𝔸, 𝔹, ℂ, 𝔻, 𝔼, 𝔽.

- Computation range (in Base-n mode)

Binary	Positive: $1111111111111111111111111111111 \geqq x \geqq 0$ Negative: $11111111111111111111111111111111 \geqq x \geqq 10000000000000000000000000000000$
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: $FFFFFFFF \geqq x \geqq 80000000$

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

■ Binary, octal, decimal, hexadecimal conversions

Example	Operation	Display
	$\boxed{\text{MODE}}$ $\boxed{-}$ →"Base-n"	
What are the decimal values for $2A_{16}$ and 274_8?	$\boxed{\text{Dec}}$ $\boxed{\text{EXE}}$ →"Dec"	
	$\boxed{\text{SHIFT}}$ $\boxed{\text{h}}$ 2 A $\boxed{\text{EXE}}$	42.
	$\boxed{\text{SHIFT}}$ $\boxed{\text{o}}$ 274 $\boxed{\text{EXE}}$	188.
What are the hexadecimal values for 123_{10} and 1010_2?	$\boxed{\text{Hex}}$ $\boxed{\text{EXE}}$ →"Hex"	
	$\boxed{\text{SHIFT}}$ $\boxed{\text{d}}$ 123 $\boxed{\text{EXE}}$	0000007 𝔹
	$\boxed{\text{SHIFT}}$ $\boxed{\text{b}}$ 1010 $\boxed{\text{EXE}}$	0000000 𝔸
What are the octal values for 15_{16} and 1100_2?	$\boxed{\text{Oct}}$ $\boxed{\text{EXE}}$ →"Oct"	
	$\boxed{\text{SHIFT}}$ $\boxed{\text{h}}$ 15 $\boxed{\text{EXE}}$	00000000025
	$\boxed{\text{SHIFT}}$ $\boxed{\text{b}}$ 1100 $\boxed{\text{EXE}}$	00000000014
What are the binary values for 36_{10} and $3B7_{16}$?	$\boxed{\text{Bin}}$ $\boxed{\text{EXE}}$ →"Bin"	
	$\boxed{\text{SHIFT}}$ $\boxed{\text{d}}$ 36 $\boxed{\text{EXE}}$	← 00100100
	$\boxed{\text{SHIFT}}$ $\boxed{\text{h}}$ 3B7 $\boxed{\text{EXE}}$	← 10110111
	(Continuing) $\boxed{\text{Block}}$ 2 $\boxed{\text{EXE}}$	← 00000011→ (=1110110111)

Negative expressions

Example	Operation	Display
	[MODE] [−] →"Base-n"	
How is 110010_2 expressed as a negative?	[Bin] [EXE] →"Bin" [Neg] 110010 [EXE]	(1st block) **11001110**
	[Block] 2 [EXE]	(2nd block) **11111111**
	[Block] 3 [EXE]	(3rd block) **11111111**
	[Block] 4 [EXE]	(4th block) **11111111**
	(=1111111111111111111111111001110)	
How is 72_8 expressed as a negative?	[Oct] [EXE] →"Oct" [Neg] 72 [EXE]	**37777777706**
How is $3A_{16}$ expressed as a negative?	[Hex] [EXE] →"Hex" [Neg] 3A [EXE]	**FFFFFFC6**

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

Example	Operation	Display
	[MODE] [−] →"Base-n"	
$10111_2+11010_2=110001_2$	[Bin] [EXE] →"Bin" 10111 [+] 11010 [EXE]	**00110001**
$B47_{16}-DF_{16}=A68_{16}$	[Hex] [EXE] →"Hex" B47 [−] DF [EXE]	**00000A68**
$123_8\times ABC_{16}=37AF4_{16}$ $=228084_{10}$	[SHIFT] [o] 123 [×] ABC [EXE] [Dec] [EXE]	**00037AF4** **228084**
$1F2D_{16}-100_{10}=7881_{10}$ $=1EC9_{16}$	[SHIFT] [h] 1F2D [−] 100 [EXE] [Hex] [EXE]	**7881** **00001EC9**
$7654_8\div 12_{10}$ $=334.3333333_{10}$ $=516_8$	[Dec] [EXE] →"Dec" [SHIFT] [o] 7654 [÷] 12 [EXE] [Oct] [EXE]	 **334** **00000000516**
$1234+1EF_{16}\div 24_8=2352_8$ $=1258_{10}$	[SHIFT] [d] 1234 [+] [SHIFT] [h] 1EF [÷] 24 [EXE] [Dec] [EXE]	**00000002352** **1258**

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

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

Logical operations

Logical operations are performed through logical product (AND), logical sum (OR) and negation (NOT).

Example	Operation	Display
	[MODE] [−] →"Base-n"	
19_{16} AND $1A_{16}=18_{16}$	[Hex] [EXE] →"Hex" 19 [and] 1A [EXE]	**00000018**
1110_2 AND $36_8=1110_2$	[Bin] [EXE] →"Bin" 1110 [and] [SHIFT] [o] 36 [EXE]	**00001110**
23_8 OR $61_8=63_8$	[Oct] [EXE] →"Oct" 23 [or] 61 [EXE]	**00000000063**
120_{16} OR $1101_2=12D_{16}$	[Hex] [EXE] →"Hex" 120 [or] [SHIFT] [b] 1101 [EXE]	**0000012D**
1010_2 AND (A_{16} OR 7_{16}) $=1010_2$	[Bin] [EXE] →"Bin" 1010 [and] [(] [SHIFT] [h] A [or] [SHIFT] [h] 7 [)] [EXE]	**00001010**
Negation of 1234_8	[Oct] [EXE] →"Oct" [Not] 1234 [EXE]	**37777776543**
Negation of $2FFFED_{16}$	[Hex] [EXE] →"Hex" [Not] 2FFFED [EXE]	**FFD00012**

2-5 STATISTICAL COMPUTATIONS

■ Standard deviation

- ●Standard deviation computations are performed in the SD mode. (Press $\boxed{\text{MODE}}$ $\boxed{\times}$, and "SD" illuminates on display.)
- ●Before beginning computations, the tabulation memories are cleared by pressing $\boxed{\text{SHIFT}}$ followed by $\boxed{\text{Scl}}$ ($\underset{\boxed{\text{Scl}}}{\boxed{\text{AC}}}$ key) and then $\boxed{\text{EXE}}$.
- ●Individual data is input using $\boxed{\text{DT}}$ ($\underset{\boxed{\text{DT}}}{\boxed{\sqrt[x]{\ }}}$ key).
- ●Multiple data of the same value can be input either by repeatedly pressing $\boxed{\text{DT}}$ or by entering the data, pressing $\boxed{\text{SHIFT}}$, followed by $\boxed{;}$, a value that represents the number of times the data is repeated, and then $\boxed{\text{DT}}$.
- ●Standard deviation

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

(Using the entire data of a finite population to determine the standard deviation for the population.)

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

(Using sample data for a population to determine the standard deviation for the population.)

- ●Mean

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

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

Example	Operation	Display
Data 55,54,51,55, 53,53,54,52	$\boxed{\text{MODE}}$ $\boxed{\times}$ →"SD" $\boxed{\text{SHIFT}}$ $\boxed{\text{Scl}}$ $\boxed{\text{EXE}}$ (Memory clear) 55 $\boxed{\text{DT}}$ 54 $\boxed{\text{DT}}$ 51 $\boxed{\text{DT}}$ 55 $\boxed{\text{DT}}$ 53 $\boxed{\text{DT}}$ $\boxed{\text{DT}}$ 54 $\boxed{\text{DT}}$ 52 $\boxed{\text{DT}}$	52.
∗ Results can be obtained in any order desired.		
	(Standard deviation σ_n) $\boxed{\text{SHIFT}}$ $\boxed{x\sigma_n}$ $\boxed{\text{EXE}}$	1.316956719
	(Standard deviation σ_{n-1}) $\boxed{\text{SHIFT}}$ $\boxed{x\sigma_{n-1}}$ $\boxed{\text{EXE}}$	1.407885953
	(Mean $\bar{x}$) $\boxed{\text{SHIFT}}$ $\boxed{\bar{x}}$ $\boxed{\text{EXE}}$	53.375
	(Number of data n) $\boxed{\text{ALPHA}}$ $\boxed{\text{W}}$ $\boxed{\text{EXE}}$	8.
	(Sum total Σx) $\boxed{\text{ALPHA}}$ $\boxed{\text{V}}$ $\boxed{\text{EXE}}$	427.
	(Sum of squares Σx^2) $\boxed{\text{ALPHA}}$ $\boxed{\text{U}}$ $\boxed{\text{EXE}}$	22805.
What is deviation of the unbiased variance, the difference between each datum, and the mean of the above data?	(Continuing) $\boxed{\text{SHIFT}}$ $\boxed{x\sigma_{n-1}}$ $\boxed{x^2}$ $\boxed{\text{EXE}}$	1.982142857
	55 $\boxed{-}$ $\boxed{\text{SHIFT}}$ $\boxed{\bar{x}}$ $\boxed{\text{EXE}}$	1.625
	54 $\boxed{-}$ $\boxed{\text{SHIFT}}$ $\boxed{\bar{x}}$ $\boxed{\text{EXE}}$	0.625
	51 $\boxed{-}$ $\boxed{\text{SHIFT}}$ $\boxed{\bar{x}}$ $\boxed{\text{EXE}}$	−2.375
	⋮	⋮
What is $\bar{x}$ and $x\sigma_{n-1}$ for the following table?	$\boxed{\text{SHIFT}}$ $\boxed{\text{Scl}}$ $\boxed{\text{EXE}}$	
	110 $\boxed{\text{SHIFT}}$ $\boxed{;}$ 10 $\boxed{\text{DT}}$	110.
	130 $\boxed{\text{SHIFT}}$ $\boxed{;}$ 31 $\boxed{\text{DT}}$	130.
	150 $\boxed{\text{SHIFT}}$ $\boxed{;}$ 24 $\boxed{\text{DT}}$	150.
	170 $\boxed{\text{DT}}$ $\boxed{\text{DT}}$	170.
	190 $\boxed{\text{DT}}$ $\boxed{\text{DT}}$ $\boxed{\text{DT}}$	190.
	$\boxed{\text{ALPHA}}$ $\boxed{\text{W}}$ $\boxed{\text{EXE}}$	70.
	$\boxed{\text{SHIFT}}$ $\boxed{\bar{x}}$ $\boxed{\text{EXE}}$	137.7142857
	$\boxed{\text{SHIFT}}$ $\boxed{x\sigma_{n-1}}$ $\boxed{\text{EXE}}$	18.42898069

Class No.	Value	Fre-quency
1	110	10
2	130	31
3	150	24
4	170	2
5	190	3

∗Erroneous data clearing/correction I (correct data operation: 51 $\boxed{\text{DT}}$)

① If 50 $\boxed{\text{DT}}$ is entered, enter correct data after pressing $\boxed{\text{Cl}}$ ($\boxed{x^y}$ key).

② If 49 $\boxed{\text{DT}}$ was input a number of entries previously, enter correct data after pressing 49 $\boxed{\text{Cl}}$.

*Erroneous data clearing/correction II (correct data operation : 130 $\boxed{\text{SHIFT}}$ $\boxed{;}$ 31 $\boxed{\text{DT}}$)

① If 120 $\boxed{\text{SHIFT}}$ $\boxed{;}$ is entered, enter correct data after pressing $\boxed{\text{AC}}$.

② If 120 $\boxed{\text{SHIFT}}$ $\boxed{;}$ 31 is entered, enter correct data after pressing $\boxed{\text{AC}}$.

③ If 120 $\boxed{\text{SHIFT}}$ $\boxed{;}$ 30 $\boxed{\text{DT}}$ is entered, enter correct data after pressing $\boxed{\text{Cl}}$.

④ If 120 $\boxed{\text{SHIFT}}$ $\boxed{;}$ 30 $\boxed{\text{DT}}$ was entered previously, enter correct data after pressing 120 $\boxed{\text{SHIFT}}$ $\boxed{;}$ 30 $\boxed{\text{Cl}}$.

■ Regression computation

- ● Regression computations are performed in the LR mode. (Press $\boxed{\text{MODE}}$ $\boxed{\div}$, and "LR" illuminates on display.)
- ● Before beginning computations, the tabulation memories are cleared by pressing $\boxed{\text{SHIFT}}$ followed by $\boxed{\text{Scl}}$ and then $\boxed{\text{EXE}}$.
- ● Individual data are entered as x data $\boxed{\text{SHIFT}}$ $\boxed{,}$ y data $\boxed{\text{DT}}$.
- ● Multiple data of the same value can be entered by repeatedly pressing $\boxed{\text{DT}}$. This operation can also be performed by entering x data $\boxed{\text{SHIFT}}$ $\boxed{,}$ y data $\boxed{\text{SHIFT}}$ $\boxed{;}$ followed by a value representing the number of times the data is repeated, and then $\boxed{\text{DT}}$.
- ● If only x data is repeated (x data having the same value), enter $\boxed{\text{SHIFT}}$ $\boxed{,}$ y data $\boxed{\text{DT}}$ or $\boxed{\text{SHIFT}}$ $\boxed{,}$ y data $\boxed{\text{SHIFT}}$ $\boxed{;}$ followed by a value representing the number of times the data is repeated, and then $\boxed{\text{DT}}$.
- ● If only y data is repeated (y data having the same value), enter x data $\boxed{\text{DT}}$ or x data $\boxed{\text{SHIFT}}$ $\boxed{;}$ followed by a value representing the total number of times the data is repeated, and then $\boxed{\text{DT}}$.
- ● The regression formula is $y=\text{A}+\text{B}x$, and constant term A and regression coefficient B are computed using the following formulas:

Regression coefficient of regression formula	Constant term of regression formula
$\text{B}=\dfrac{n\cdot\Sigma xy-\Sigma x\cdot\Sigma y}{n\cdot\Sigma x^2-(\Sigma x)^2}$	$\text{A}=\dfrac{\Sigma y-\text{B}\cdot\Sigma x}{n}$

- ● Estimated values $\hat{x}$ and $\hat{y}$ based on the regression formula can be computed.
- ● The correlation coefficient r for input data can be computed using the following formula:

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

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

◆ Linear regression

<table>
<tr><th>Example</th><th>Operation</th><th>Display</th></tr>
<tr><td rowspan="14">● Temperature and the length of a steel bar
<table>
<tr><th>Temp.</th><th>Length</th></tr>
<tr><td>10℃</td><td>1003mm</td></tr>
<tr><td>15</td><td>1005</td></tr>
<tr><td>20</td><td>1010</td></tr>
<tr><td>25</td><td>1011</td></tr>
<tr><td>30</td><td>1014</td></tr>
</table>
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℃ and the temperature at 1000mm can be estimated. Furthermore, the critical coefficient (r²) and covariance $\left(\dfrac{\Sigma xy-n\cdot\bar{x}\cdot\bar{y}}{n-1}\right)$ can also be computed.</td><td>$\boxed{\text{MODE}}$ $\boxed{\div}$ →"LR"</td><td></td></tr>
<tr><td>$\boxed{\text{SHIFT}}$ $\boxed{\text{Scl}}$ $\boxed{\text{EXE}}$ (Memory clear)</td><td></td></tr>
<tr><td>10 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1003 $\boxed{\text{DT}}$</td><td>10.</td></tr>
<tr><td>15 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1005 $\boxed{\text{DT}}$</td><td>15.</td></tr>
<tr><td>20 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1010 $\boxed{\text{DT}}$</td><td>20.</td></tr>
<tr><td>25 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1011 $\boxed{\text{DT}}$</td><td>25.</td></tr>
<tr><td>30 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1014 $\boxed{\text{DT}}$</td><td>30.</td></tr>
<tr><td>(Constant term A)
$\boxed{\text{SHIFT}}$ $\boxed{\text{A}}$ $\boxed{\text{EXE}}$</td><td>997.4</td></tr>
<tr><td>(Regression coefficient B)
$\boxed{\text{SHIFT}}$ $\boxed{\text{B}}$ $\boxed{\text{EXE}}$</td><td>0.56</td></tr>
<tr><td>(Correlation coefficient r)
$\boxed{\text{SHIFT}}$ $\boxed{\text{r}}$ $\boxed{\text{EXE}}$</td><td>0.9826073689</td></tr>
<tr><td>(Length at 18℃)
18 $\boxed{\text{SHIFT}}$ $\boxed{\hat{y}}$ $\boxed{\text{EXE}}$</td><td>1007.48</td></tr>
<tr><td>(Temperature at 1000mm)
1000 $\boxed{\text{SHIFT}}$ $\boxed{\hat{x}}$ $\boxed{\text{EXE}}$</td><td>4.642857142</td></tr>
<tr><td>(Critical coefficient)
$\boxed{\text{SHIFT}}$ $\boxed{\text{r}}$ $\boxed{x^2}$ $\boxed{\text{EXE}}$</td><td>0.9655172414</td></tr>
<tr><td>(Covariance) $\boxed{(}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{R}}$ $\boxed{-}$
$\boxed{\text{ALPHA}}$ $\boxed{\text{W}}$ $\boxed{\times}$ $\boxed{\text{SHIFT}}$ $\boxed{\bar{x}}$ $\boxed{\times}$ $\boxed{\text{SHIFT}}$ $\boxed{\bar{y}}$
$\boxed{)}$ $\boxed{\div}$ $\boxed{(}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{W}}$ $\boxed{-}$ 1 $\boxed{)}$
$\boxed{\text{EXE}}$</td><td>35.</td></tr>
</table>

*Erroneous data clearing/correction (correct data operation : 10 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1003 $\boxed{\text{DT}}$)

① If 11 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1003 is entered, enter correct data after pressing $\boxed{\text{AC}}$.

② If 11 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1003 $\boxed{\text{DT}}$ is entered, enter correct data after pressing $\boxed{\text{Cl}}$

③ If 11 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1003 $\boxed{\text{DT}}$ was entered previously, enter correct data after pressing 11 $\boxed{\text{SHIFT}}$ $\boxed{,}$ 1003 $\boxed{\text{Cl}}$.

◆ Logarithmic regression

- The regression formula is $y=\mathrm{A}+\mathrm{B}\cdot\ln x$. Enter the x data as the logarithm (ln) of x, and the y data inputs the same as that for linear regression.
- The same operation as with linear regression can be used to obtain the regression coefficient and for making corrections. To obtain the estimated value $\hat{y}$, $\boxed{\ln}\ x\ \boxed{\text{SHIFT}}\ \boxed{\hat{y}}\ \boxed{\text{EXE}}$ is used, and to obtain estimated value $\hat{x}$, $y\ \boxed{\text{SHIFT}}\ \boxed{\hat{x}}\ \boxed{\text{EXE}}\ \boxed{\text{SHIFT}}\ \boxed{e^x}\ \boxed{\text{Ans}}\ \boxed{\text{EXE}}$ is used.
  Furthermore, Σx, $\Sigma x^2$and Σxy are obtained as $\Sigma\ln x$, $\Sigma(\ln x)^2$, and $\Sigma\ln xy$ respectively.

x_i	y_i
29	1.6
50	23.5
74	38.0
103	46.4
118	48.9

Example	Operation	Display
	$\boxed{\text{MODE}}\ \boxed{\div}$ →"LR"	
	$\boxed{\text{SHIFT}}\ \boxed{\text{Scl}}\ \boxed{\text{EXE}}$	
	$\boxed{\ln}$ 29 $\boxed{\text{SHIFT}}\ \boxed{,}$ 1.6 $\boxed{\text{DT}}$	3.36729583
	$\boxed{\ln}$ 50 $\boxed{\text{SHIFT}}\ \boxed{,}$ 23.5 $\boxed{\text{DT}}$	3.912023005
	$\boxed{\ln}$ 74 $\boxed{\text{SHIFT}}\ \boxed{,}$ 38.0 $\boxed{\text{DT}}$	4.304065093
	$\boxed{\ln}$ 103 $\boxed{\text{SHIFT}}\ \boxed{,}$ 46.4 $\boxed{\text{DT}}$	4.634728988
	$\boxed{\ln}$ 118 $\boxed{\text{SHIFT}}\ \boxed{,}$ 48.9 $\boxed{\text{DT}}$	4.770684624
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.	(Constant term A) $\boxed{\text{SHIFT}}\ \boxed{\text{A}}\ \boxed{\text{EXE}}$	−111.1283983
	(Regression coefficient B) $\boxed{\text{SHIFT}}\ \boxed{\text{B}}\ \boxed{\text{EXE}}$	34.02014766
	(Correlation coefficient r) $\boxed{\text{SHIFT}}\ \boxed{\text{r}}\ \boxed{\text{EXE}}$	0.9940139485
	($\hat{y}$ when $xi=80$) $\boxed{\ln}$ 80 $\boxed{\text{SHIFT}}\ \boxed{\hat{y}}\ \boxed{\text{EXE}}$	37.94879487
	($\hat{x}$ when $yi=73$) 73 $\boxed{\text{SHIFT}}\ \boxed{\hat{x}}\ \boxed{\text{EXE}}\ \boxed{\text{SHIFT}}\ \boxed{e^x}\ \boxed{\text{Ans}}\ \boxed{\text{EXE}}$	224.1541299

◆ Exponential regression

- The regression formula is $y=\mathrm{A}\cdot e^{\mathrm{B}\cdot x}(\ln y=\ln\mathrm{A}+\mathrm{B}x)$. Enter the y data as the logarithm of y(ln), and the x data the same as that for linear regression.
- Correction is performed the same as in linear regression. Constant term A is obtained by $\boxed{\text{SHIFT}}\ \boxed{e^x}\ \boxed{\text{SHIFT}}\ \boxed{\text{A}}\ \boxed{\text{EXE}}$, estimated value $\hat{y}$ is obtained by $x\ \boxed{\text{SHIFT}}\ \boxed{\hat{y}}\ \boxed{\text{EXE}}\ \boxed{\text{SHIFT}}\ \boxed{e^x}\ \boxed{\text{Ans}}\ \boxed{\text{EXE}}$, and estimated value $\hat{x}$ is obtained by $\boxed{\ln}\ y\ \boxed{\text{SHIFT}}\ \boxed{\hat{x}}\ \boxed{\text{EXE}}$. Σy, Σy^2 and Σxy are obtained by $\Sigma\ln y$, $\Sigma(\ln y)^2$ and $\Sigma x\cdot\ln y$ respectively.

x_i	y_i
6.9	21.4
12.9	15.7
19.8	12.1
26.7	8.5
35.1	5.2

Example	Operation	Display
	$\boxed{\text{SHIFT}}\ \boxed{\text{Scl}}\ \boxed{\text{EXE}}$	
	6.9 $\boxed{\text{SHIFT}}\ \boxed{,}\ \boxed{\ln}$ 21.4 $\boxed{\text{DT}}$	6.9
	12.9 $\boxed{\text{SHIFT}}\ \boxed{,}\ \boxed{\ln}$ 15.7 $\boxed{\text{DT}}$	12.9
	19.8 $\boxed{\text{SHIFT}}\ \boxed{,}\ \boxed{\ln}$ 12.1 $\boxed{\text{DT}}$	19.8
	26.7 $\boxed{\text{SHIFT}}\ \boxed{,}\ \boxed{\ln}$ 8.5 $\boxed{\text{DT}}$	26.7
	35.1 $\boxed{\text{SHIFT}}\ \boxed{,}\ \boxed{\ln}$ 5.2 $\boxed{\text{DT}}$	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 value $\hat{y}$ and $\hat{x}$ when $xi=16$ and $yi=20$.	(Constant term A) $\boxed{\text{SHIFT}}\ \boxed{e^x}\ \boxed{\text{SHIFT}}\ \boxed{\text{A}}\ \boxed{\text{EXE}}$	30.49758747
	(Regression coefficient B) $\boxed{\text{SHIFT}}\ \boxed{\text{B}}\ \boxed{\text{EXE}}$	-4.920370834^{-02}
	(Correlation coefficient r) $\boxed{\text{SHIFT}}\ \boxed{\text{r}}\ \boxed{\text{EXE}}$	−0.9972473591
	($\hat{y}$ when $xi=16$) 16 $\boxed{\text{SHIFT}}\ \boxed{\hat{y}}\ \boxed{\text{EXE}}\ \boxed{\text{SHIFT}}\ \boxed{e^x}\ \boxed{\text{Ans}}\ \boxed{\text{EXE}}$	13.8791574
	($\hat{x}$ when $yi=20$) $\boxed{\ln}$ 20 $\boxed{\text{SHIFT}}\ \boxed{\hat{x}}\ \boxed{\text{EXE}}$	8.574868061

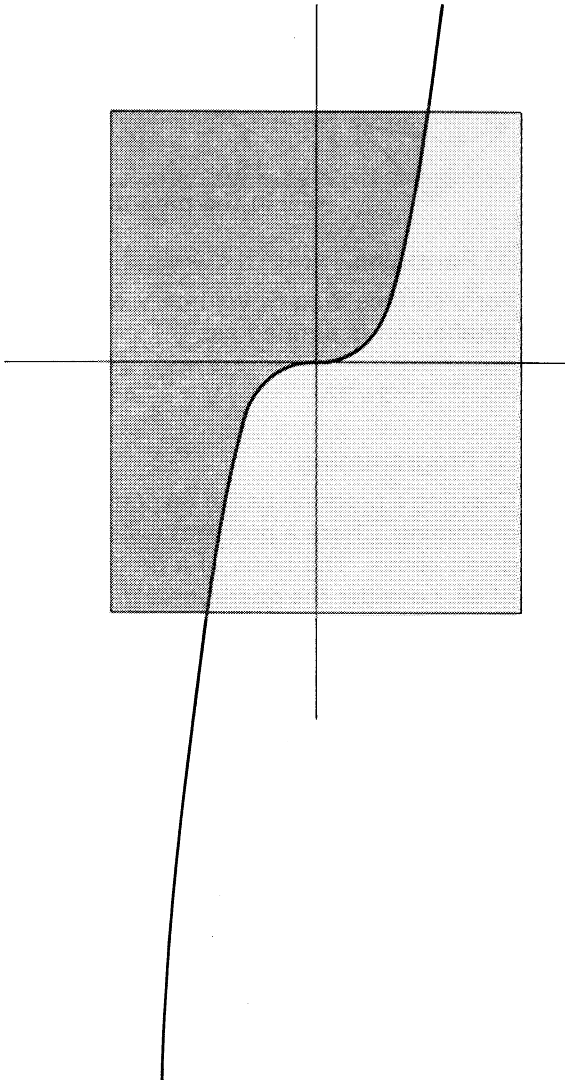
◆ Power regression

- The regression formula is $y = \text{A} \cdot x^{\text{B}}$ ($\ln y = \ln \text{A} + \text{B} \ln x$). Enter both data x and y as logarithms (ln).
- Correction is performed the same as in linear regression. Constant term A is obtained by SHIFT e^x SHIFT A EXE , estimated value $\hat{y}$ is obtained by ln x SHIFT $\hat{y}$ EXE SHIFT e^x Ans EXE, and estimated value $\hat{x}$ is obtained by ln y SHIFT $\hat{x}$ EXE SHIFT e^x Ans EXE. Σx, Σx^2, Σy, Σy^2 and Σxy are obtained by $\Sigma \ln x$, $\Sigma (\ln x)^2$, $\Sigma \ln y$, $\Sigma (\ln y)^2$ and $\Sigma \ln x \cdot \ln y$ respectively.

Example		Operation	Display
x_i	y_i	SHIFT Scl EXE	
28	2410	ln 28 SHIFT , ln 2410 DT	3.33220451
30	3033	ln 30 SHIFT , ln 3033 DT	3.401197382
33	3895	ln 33 SHIFT , ln 3895 DT	3.496507561
35	4491	ln 35 SHIFT , ln 4491 DT	3.555348061
38	5717	ln 38 SHIFT , ln 5717 DT	3.63758616
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 $\hat{y}$ and $\hat{x}$ when $x_i = 40$ and $y_i = 1000$.		(Constant term A) SHIFT e^x SHIFT A EXE	0.2388003228
		(Regression coefficient B) SHIFT B EXE	2.771867061
		(Correlation coefficient r) SHIFT r EXE	0.9989066443
		($\hat{y}$ when $x_i = 40$) ln 40 SHIFT $\hat{y}$ EXE SHIFT e^x Ans EXE	6587.675955
		($\hat{x}$ when $y_i = 1000$) ln 1000 SHIFT $\hat{x}$ EXE SHIFT e^x Ans EXE	20.26225977

3.

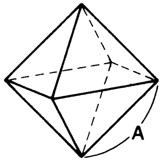
PROGRAM COMPUTATIONS



3-1 WHAT IS A PROGRAM?

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

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



Length of one side (A)	Surface area (S)	Volume (V)
10cm	( )cm²	( )cm³
7	( )	( )
15	( )	( )

*Fill in the parentheses.

① Formulas

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

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

② Programming

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

Surface area (S): $2\ \boxed{\times}\ \boxed{\sqrt{}}\ 3\ \boxed{\times}$ Numeric value A $\boxed{x^2}\ \boxed{\text{EXE}}$

Volume(V): $\boxed{\sqrt{}}\ 2\ \boxed{\div}\ 3\ \boxed{\times}$ Numeric value A $\boxed{x^y}\ 3\ \boxed{\text{EXE}}$

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

Numeric value A $\rightarrow\ \boxed{\text{ALPHA}}\ \boxed{\text{A}}\ \boxed{\text{EXE}}$

$2\ \boxed{\times}\ \boxed{\sqrt{}}\ 3\ \boxed{\times}\ \boxed{\text{ALPHA}}\ \boxed{\text{A}}\ \boxed{x^2}\ \boxed{\text{EXE}}$ ································ S

$\boxed{\sqrt{}}\ 2\ \boxed{\div}\ 3\ \boxed{\times}\ \boxed{\text{ALPHA}}\ \boxed{\text{A}}\ \boxed{x^y}\ 3\ \boxed{\text{EXE}}$ ····························· V

With this unit, the operations performed for manual computations can be used as they are in a program. Once program execution starts, it will continue in order without stopping. Therefore, commands are required to request the input of data and to display results. The command to request data input is "?", while that to display results is "◢". A "?"within a program will cause execution to stop temporarily and a "?" to appear on the display as the unit waits for data input. This command cannot be used independently, and is used together with $\boxed{\rightarrow}$ as "$\boxed{\text{SHIFT}}\ \boxed{?}\ \boxed{\rightarrow}$ memory name". To store a numeric value in memory A, for example:

?→A

The "◢" command causes program execution to stop temporarily and the latest formula result or alphanumeric characters and symbols (see page 84) to be displayed. This command is used to mark positions in formulas where results are to be displayed. Since programs are ended and their final results displayed automatically, this command can be omitted at the end of a program.
Here these two commands will be used in the previously presented procedure:

$$\underbrace{\boxed{\text{SHIFT}}\ \boxed{?}\ \boxed{\rightarrow}\ \boxed{\text{ALPHA}}\ \boxed{\text{A}}}_{\text{Input to memory A}}\ \boxed{:}\ 2\ \boxed{\times}\ \boxed{\sqrt{}}\ 3\ \boxed{\times}\ \boxed{\text{ALPHA}}\ \boxed{\text{A}}\ \boxed{x^2}\ \underbrace{\boxed{\text{SHIFT}}\ \boxed{\text{◢}}}_{\text{Display S}}$$

$$\boxed{\sqrt{}}\ 2\ \boxed{\div}\ 3\ \boxed{\times}\ \boxed{\text{ALPHA}}\ \boxed{\text{A}}\ \boxed{x^y}\ 3\ \underset{\uparrow\ \text{◢ omitted}}{}$$

Now the program is complete.

③ Program storage

The storage of programs is performed in the WRT mode which is specified by pressing $\boxed{\text{MODE}}\ \boxed{2}$ ("WRT" illuminates on the display).

Operation	Display
$\boxed{\text{MODE}}\ \boxed{2}$	WRT D 550 P 0123456789

When $\boxed{\text{MODE}}\ \boxed{2}$ are pressed, "WRT" appears at the upper left of the display to indicate that the unit is in the WRT mode. Then, the number of remaining steps (see page 64) is indicated at the upper right of the display. The number of remaining steps is decreased when programs are input or when memories are expanded. If no programs have been input and the number of memories equals 26 (the number of memories at initialization), the number of usable steps should equal 550.

The larger figures located below indicate the program areas (see page 66). If the letter "P" is followed by the numbers 0 through 9, it indicates that there are no programs stored in areas P0 through P9. The blinking zero here indicates the current program area is P0.
Areas into which programs have already been stored are indicated by "_" instead of numbers.

```
WRT             248
 P  01_34_6789
```

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

Operation	**Display**
EXE (Writing starts)	WRT 0 _
SHIFT ? → ALPHA A :	4 ?→A : _
2 × √ 3 × ALPHA A x^2 SHIFT ◢	← 12 →A:2×√3×A²◢_
√ 2 ÷ 3 × ALPHA A x^y 3	← 2 20 ◢√2÷3×Ax^y3_

(The number at the upper right indicates the number of steps used in the currently specified program area.)

(When the number of input characters exceeds the capacity of the display, each subsequent input character will cause the displayed characters to shift to the left. This condition will be indicated by a "←".)

After these operations are complete, the program is stored.

*After the program is stored, press MODE 1 to return to the RUN mode.
*Note that the following functions cannot be used in a program:
°'", $\overleftarrow{°'"}$, ENG, $\overleftarrow{\text{ENG}}$

---

④ Program execution

Programs are executed in the RUN mode (MODE 1). The program area to be executed is specified using the Prg key.

To execute P0: Prg 0 EXE
To execute P3: Prg 3 EXE
To execute P8: Prg 8 EXE

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

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

Operation	**Display**	
MODE 1	_	(Not required if already in the RUN mode.)
Prg 0 EXE	?	
10 EXE (Value of A)	346.4101615	(S when A=10)
EXE	471.4045208	(V when A=10)
Prg 0 EXE	?	
7 EXE (Value of A)	169.7409791	(S when A=7)
EXE	161.6917506	(V when A=7)
Prg 0 EXE	?	
15 EXE (Value of A)	779.4228634	(S when A=15)
EXE	1590.990258	(V when A=15)

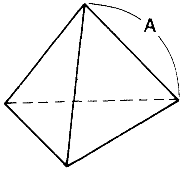
*Program computations are performed automatically with each press of EXE when it is pressed after data is input or after the result is read.

3-2 PROGRAM CHECKING AND EDITING (CORRECTION, ADDITION, DELETION)

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

EXAMPLE:

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



Length of one side (A)	Surface area (S)	Volume (V)
10 cm	( )cm²	( )cm³
7.5	( )	( )
20	( )	( )

① Formulas

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

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

② Programming

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

Numeric value A→ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{\text{EXE}}$

$\boxed{\sqrt{}}$ 3 $\boxed{\times}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{x^2}$ $\boxed{\text{EXE}}$ ······································ S

$\boxed{\sqrt{}}$ 2 $\boxed{\div}$ 12 $\boxed{\times}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{x^y}$ 3 $\boxed{\text{EXE}}$ ···························· V

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

$\boxed{\text{SHIFT}}$ $\boxed{?}$ $\boxed{\rightarrow}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{:}$ $\boxed{\sqrt{}}$ 3 $\boxed{\times}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{x^2}$ $\boxed{\text{SHIFT}}$ $\boxed{◢}$
$\boxed{\sqrt{}}$ 2 $\boxed{\div}$ 12 $\boxed{\times}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{x^y}$ 3

③ Program editing

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

Octahedron: $\boxed{\text{SHIFT}}$ $\boxed{?}$ $\boxed{\rightarrow}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{:}$ $\utilde{2 \boxed{\times}}$ $\boxed{\sqrt{}}$ 3 $\boxed{\times}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{x^2}$ $\boxed{\text{SHIFT}}$ $\boxed{◢}$
$\boxed{\sqrt{}}$ 2 $\boxed{\div}$ $\underline{3}$ $\boxed{\times}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{x^y}$ 3

Tetrahedron: $\boxed{\text{SHIFT}}$ $\boxed{?}$ $\boxed{\rightarrow}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{:}$ $\boxed{\sqrt{}}$ 3 $\boxed{\times}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{x^2}$ $\boxed{\text{SHIFT}}$ $\boxed{◢}$
$\boxed{\sqrt{}}$ 2 $\boxed{\div}$ $\underline{12}$ $\boxed{\times}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{A}}$ $\boxed{x^y}$ 3

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

In actual practice, this would be performed as follows:

Operation	Display	
$\boxed{\text{MODE}}$ $\boxed{2}$	WRT 530 P _123456789	
$\boxed{\text{EXE}}$	WRT 0 ?→A:2×√3×A²◢→	(Displayed from the beginning. To display from the end, press $\boxed{\text{SHIFT}}$ $\boxed{\text{EXE}}$.)
$\boxed{\Rightarrow}$ $\boxed{\Rightarrow}$ $\boxed{\Rightarrow}$ $\boxed{\Rightarrow}$	WRT 4 ?→A:2×√3×A²◢→	(Move the cursor to the position to be deleted.)
$\boxed{\text{DEL}}$ $\boxed{\text{DEL}}$	WRT 4 ←?→A:√3×A²◢√2→	(Delete two characters.)
$\boxed{\Rightarrow}$ ······ $\boxed{\Rightarrow}$	WRT 13 ←A:√3×A²◢√2÷3→	(Move the cursor to the position to be corrected.)
$\boxed{\text{SHIFT}}$ $\boxed{\text{INS}}$	WRT 13 ←A:√3×A²◢√2÷[]→	(Open a space for one character.)
12	WRT 15 ←√3×A²◢√2÷12×→	(Correct the numeric value.)
$\boxed{\text{MODE}}$ $\boxed{1}$	_	(After editing is complete, return to the RUN mode.)

④ Program execution

Now this program will be executed.

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

Operation	**Display**
MODE 1	_
Prg 0 EXE	?
10 EXE	173.2050808
EXE	117.8511302
Prg 0 EXE	?
7.5 EXE	97.42785793
EXE	49.71844555
Prg 0 EXE	?
20 EXE	692.820323
EXE	942.8090416

〈Summary〉

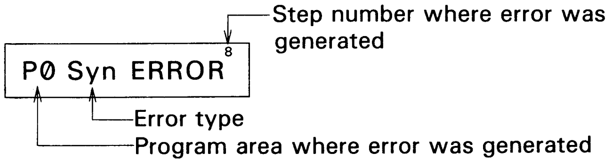
	Operation	Keys used
Program check	•WRT mode specification •Program area specification(Omitted if P0) •Start verification •Verification of contents	MODE 2 ⇦ ⇨ EXE , SHIFT EXE ⇦ ⇨
Correction	•Move the cursor to the position to be corrected. •Press correct keys.	⇦ ⇨
Deletion	•Move the cursor to the position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

3-3 PROGRAM DEBUGGING (CORRECTING ERRORS)

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

■ Debugging when an error message is generated

An error message is displayed as follows:



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

■ Error messages

There are a total of six error messages.

① **Syn ERROR** (Syntax error)
Indicates a mistake in the formula or a misuse of program commands.

② **Ma ERROR** (Mathematical error)
Indicates the computation result of a numeric expression exceeds 10^{100}, an illogical operation (i.e. division by zero), or the input of an argument that exceeds the input range of the function.

③ **Go ERROR** (Jump error)
Indicates a missing Lbl for the Goto command (see page 69), or that the program area (see page 76) for the Prg command (see page 66) does not contain a program.

④ **Ne ERROR** (Nesting error)
Indicates a subroutine nesting overflow by the Prg command.

⑤ **Stk ERROR** (Stack error)
Indicates the computation performed exceeds the capacity of the stack for numeric values or for commands (see page 15).

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

Key operation is impossible once an error message appears. The error can be cancelled by simply pressing the $\boxed{\text{AC}}$ key, but pressing either $\boxed{\Leftarrow}$ or $\boxed{\Rightarrow}$ before $\boxed{\text{AC}}$ will display the location where the error was generated as long as either key is held down.

* *Simply pressing* $\boxed{\Leftarrow}$ *or* $\boxed{\Rightarrow}$ *will not cancel the error.*
Be sure to press $\boxed{\text{AC}}$ *and then specify the WRT mode when making corrections.*

■ Checkpoints for each type of error

The following are checkpoints for each type of error:

① **Syn ERROR**
Verify again that there are no errors in the program.
Check to see that there are no mistakes in the arguments used in the program commands (i.e. Prg 10).

② **Ma ERROR**
For computations that require use of the memories, check to see that the numeric values in the memories do not exceed the range of the arguments. This type of error often occurs with division by 0 or the computation of negative square roots.

③ **Go ERROR**
Check to see that there is a corresponding Lbl n when Goto n is used. Also check to see that the program in Pn has been correctly input when Prg n is used.

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

⑤ **Stk ERROR**
Check to see that the formula is not too long thus causing a stack overflow. If this is the case, the formula should be divided into two or more parts.

⑥ **Mem ERROR**
Check to see that memories were properly expanded using "$\boxed{\text{MODE}}$ $\boxed{\cdot}$ n $\boxed{\text{EXE}}$" (Defm). When using array-type memories (see page 79), check to see that the subscripts are correct.

* *When an error is generated during program execution, the symbol "←" or "–" may illuminate.*

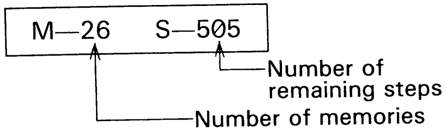
3-4 COUNTING THE NUMBER OF STEPS

The program capacity of this unit consists of a total of 550 steps. The number of steps indicates the amount of storage space available for programs, and it will decrease as programs are input. The number of remaining steps will also be decreased when steps are converted to memories. (See page 22).
There are two methods to determine the current number of remaining steps:

① When $\boxed{\text{MODE}}\ \boxed{\cdot}\ \boxed{\text{EXE}}$ are pressed in the RUN mode, the number of remaining steps will be displayed together with the number of memories.

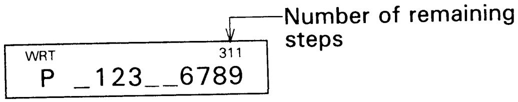
Example:

$\boxed{\text{MODE}}\ \boxed{\cdot}\ \boxed{\text{EXE}}$



② Specify the WRT mode ($\boxed{\text{MODE}}\ \boxed{2}$) and the number of remaining steps will appear at the upper right of the display. At this time the status of the program areas can also be determined.

$\boxed{\text{MODE}}\ \boxed{2}$

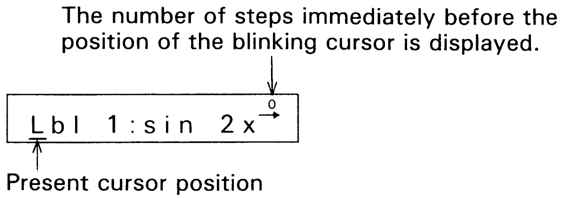


Basically, one function requires a single step, but there are some commands where one function requires two steps.

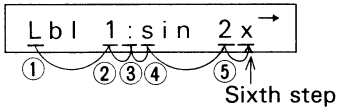
- •One function/one step: sin、cos、tan、log、(、)、:、A、B、1、2、3、etc.
- •One function/two steps: Lbl 1、Goto 2、Prg 8、etc.

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

Example:



At this time, each press of a cursor key ($\boxed{\Leftarrow}$ or $\boxed{\Rightarrow}$)will cause the cursor to move to the next sequential step. For example:



3-5 PROGRAM AREAS AND COMPUTATION MODES

This unit contains a total of 10 program areas (P0 through P9) for the storage of programs. These program areas are all utilized in the same manner, and 10 independent programs can be input. One main program (main routine) and a number of secondary programs (subroutines) can also be stored. The total number of steps available for storage in program areas P0 through P9 is 550 maximum.
Specification of a program area is performed as follows:

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

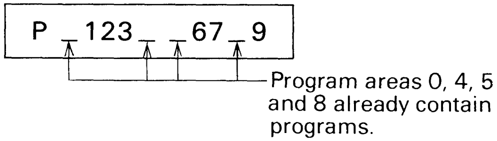
Example:	P0	[Prg] [0] [EXE]
	P8	[Prg] [8] [EXE]

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

WRT mode: Use [⇦] or [⇨] to move the cursor under the program area to be specified and press [EXE].

Only the numbers of the program areas that do not yet contain programs will be displayed. "_" symbols indicate program areas which already contain programs.

Example:



■ Program area and computation mode specification in the WRT mode

Besides normal function computations, to perform binary, octal, decimal and hexadecimal computations and conversions, standard deviation computations, and regression computations in a program, a computation mode must be specified. Program mode specification and program area specification are performed at the same time.

First the WRT mode is specified ([MODE] [2]), and then a computation mode is specified. Next, the program area is specified, and, when [EXE] is pressed, the computation mode is memorized in the program area. Henceforth, stored programs will be accompanied with the computation mode.

Example: Memorizing the Base-n mode in P2

[MODE] [2]	D P 0123456789	(Assuming that nothing is stored.)
[⇨] [⇨]	D P 0123456789	(Specify P2.)
[MODE] [−]	Dec Base-n P 0123456789	(Specify the Base-n mode.)
[EXE]	Dec Base-n _	

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

■ Cautions concerning the computation modes

All key operation available in each computation mode can be stored as programs, but, depending on the computation mode, certain commands or functions cannot be used.

Base-n mode

- Function computations cannot be performed.
- Units of angular measurement cannot be specified.
- All program commands can be used.
- Since a computation result using binary numbers is handled as 32 bits, the whole bits are not displayed at one time.
  Input "Block n ◢" in the program to display the respective 8-bit portion of the result (binary number).

SD mode

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

LR mode

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

3-6 ERASING PROGRAMS

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

■ Erasing a single program

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

Example: Erase the program in P3 only.

Operation	**Display**	
[MODE] [3]	PCL 485 P _12_45678_	(Program areas P0, P3 and P9 contain programs.)
[⇨] [⇨] [⇨]	485 P _12_45678_	(Move the cursor under P3.)
[AC]	502 P _12345678_	(The program is erased and the number "3" is displayed.)
[MODE] [1]	_	(Return to the RUN mode.)

■ Erasing all programs

To erase all programs stored in program areas 0 through 9, specify the PCL mode and press [SHIFT] and then [DEL] .

Example: Delete the programs stored in P0, P4, P8 and P9.

Operation	**Display**
[MODE] [3]	217 P _123_567__
[SHIFT] [DEL]	550 P 0123456789

3-7 CONVENIENT PROGRAM COMMANDS

The programs for this unit are made based upon manual computations. 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 command are used to change the flow of program execution. Programs are executed in the order that they are input (from the lowest step number first) until the end of the program is reached. This system is not very convenient when there are repeat computations to be performed or when it is desirable to transfer execution to another formula. It is in these cases, however, that the jump commands are very effective. There are three types of jump commands: a simple unconditional jump to a branch destination, a conditional jump that decides the branch destination by whether a certain condition is true or not, and count jump that increases or decreases a specific memory by one and then decides the branch destination after checking whether the value stored equals zero or not.

◆ Unconditional jump

The unconditional jump is composed of "Goto" and "Lbl" . When program execution reaches the statement "Goto *n*" (where *n* is a number from 0 through 9), execution then jumps to "Lbl *n*" (*n* is the same value as Goto *n*). The unconditional jump is often used in simple programs to return execution to the beginning for repetitive computations, or to repeat computations from a point within a program.

Unconditional jumps are also used in combination with conditional and count jumps.

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

The previous program contained:

?、→、A、:、√ 、3、×、A、x^2、◢、
√ 、2、÷、1、2、×、A、x^y、3 19 steps

* *Hereinafter, commas (、) will be used to separate steps for the sake of clarity.*

Add "Goto 1" to the end of the program, and add "Lbl 1" to the beginning of the program as the branch destination.
If this is simply left the way it is, however, the volume will not be displayed and execution will move immediately to the input of one side at the beginning. To prevent this situation, insert a display command (◢) in front of the "Goto 1".

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

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

Now let's try executing this program.
**For details on inputting programs and editing programs, see sections 3-1 and 3-2.*

Operation	**Display**	
Prg 0 EXE	?	(Stored in P0.)
10 EXE	173. 2050808	(The length of the side = 10)
EXE	117. 8511302	
EXE	?	
7.5 EXE	97. 42785793	(The length of the side = 7.5)
EXE	49. 71844555	
EXE	?	

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

MODE 1	—

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

Example: Compute y=ax+ b when the value for x changes each time, while a and b can also change depending upon the computation.

Program

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

When this program is executed, the values for a and b are stored in memories A and B respectively. After that, only the value for x can be changed.
In this way an unconditional jump is made in accordance with "Goto" and "Lbl", and the flow of program execution is changed. When there is no "Lbl n" to correspond to a "Goto n", an error (Go ERROR) is generated.

◈ Conditional jump

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

Left side　Relational operator　Right side ⇒ Statement { : / ◢ }*　Statement

* *Either can be used.*

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

Left side = right side (left side equals right side)
Left side ≠ right side (left side does not equal right side)
Left side ≥ right side (left side is greater than or equal to right side)
Left side ≤ right side (left side is less than or equal to right side)
Left side > right side (left side is greater than right side)
Left side < right side (left side is less than right side)

The "⇒" is displayed when SHIFT 7 are pressed. If the condition is true, execution advances to the statement following ⇒. If the condition is not true, the statement following ⇒ is skipped and execution jumps to the statement following the next ":" or "◢".

If true
Left side　Relational operator　Right side ⇒ Statement { : / ◢ }　Statement
If not true

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

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

Program

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

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

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

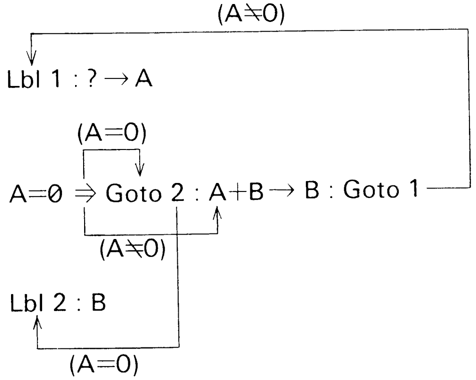
Program

```
0、→、B、:、
Lbl、1、:、?、→、A、:、A、=、0、⇒、Goto、2、:、
A、+、B、→、B、:、Goto、1、:、
Lbl、2、:、B                                31 steps
```

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

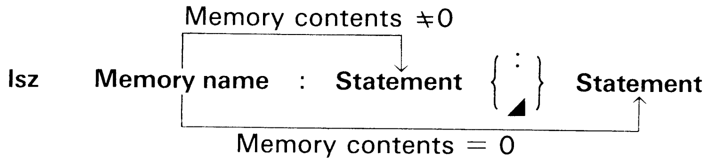
Execution from Lbl 2 will display the sum that has been stored in memory B. Actually, the display command "◢" is inserted following B, but here it can be omitted.

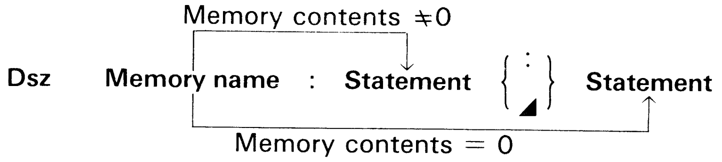
The following illustration shows the flow of the program:



◆ Count jump

The count jump causes the value in a specified memory to be increased or decreased by 1. If the value does equal 0, the following statement is skipped, and the statement following the next ":" or "◢" is executed. The "Isz" command is used to increase the value in memory by 1 and decide the subsequent execution, while the "Dsz" command is used to decrease the value by 1 and decide.





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

Example: Determine the average of 10 input numeric values.

Program

```
1、0、→、A、:、0、→、C、:、
Lbl、1、:、?、→、B、:、B、+、C、→、C、:、
Dsz、A、:、Goto、1、:、C、÷、1、0               32 steps
```

In this program, first 10 is stored in memory A, and 0 is stored in memory C. Memory A is used as the "counter" and countdown is performed the specified number of times by the Dsz command. Memory C is used to store the sum of the inputs, and so first must be cleared by inputting a 0.

The numeric value input in response to "?" is stored in memory B, and then the sum of the input values is stored in memory C by "B+C→C".

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

Example: Determine the altitude at one-second intervals of a ball thrown into the air at an initial velocity of Vm/sec and an angle of S°. The formula is expressed as: $h=V\sin\theta t-\frac{1}{2}gt^2$, with $g=9.8$, with the effects of air resistance being disregarded.

Program

Deg、 :、 0、 →、 T、 :、 ?、 →、 V、 :、 ?、 →、 S、 :、
Lbl、 1、 :、 Isz、 T、 :、 V、 ×、 sin、 S、 ×、 T、 −、
9、 ·、 8、 ×、 T、 x^2、 ÷、 2、 ◢、 Goto、 1
38 steps

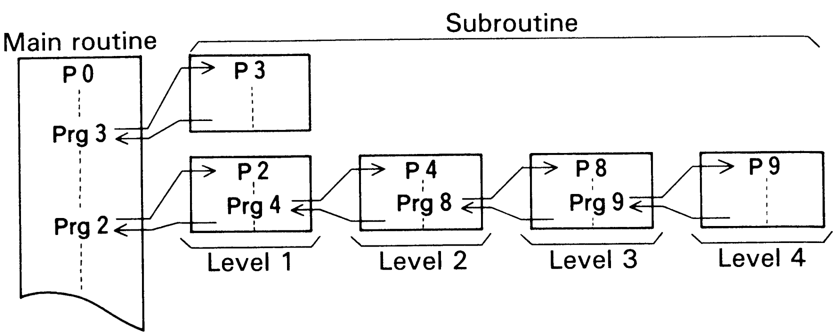
In this program the unit of angular measurement is set and memory T is first initialized (cleared). Then the initial velocity and angle are input into memories V and S respectively.
Lbl 1 is used at the beginning of the repeat computations. The numeric value stored in memory T is counted up (increased by 1) by Isz T. In this case、 the Isz command is used only for the purpose of increasing the value stored in memory T, and the subsequent jump does not depend upon any comparison or decision. The Isz command can also be used in the same manner as seen with the Dsz command for jumps that require decisions, but, as can be seen here, it can also be used to simply increase values. If, in place of the Isz command, another method such as "T+1→T" is used, five steps are required instead of the two for the (Isz T) method shown here. Such commands are convenient ways of conserving memory space.
Each time memory T is increased computation is performed according to the formula, and the altitude is displayed. It should be noted that this program is endless, so when the required value is obtained, `MODE` `1` are pressed to terminate the program.

〈**Summary**〉

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

■ Subroutines

A program contained in a single program area is called a "main routine". Often used program segments stored in other program areas are called "subroutines".
Subroutines can be used in a variety of ways to help make computations easier. They can be used to store formulas for repeat computations 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 "Prg" followed by a number from 0 through 9 which indicates the program area.

Example: Prg 0……Jump to program area 0
Prg 2……Jump to program area 2

After the jump is performed using the Prg 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 Prg n command in the original program area. Jumps can be performed from one subroutine to another, and this procedure is known as "nesting". Nesting can be performed to a maximum of 9 levels, and attempts to exceed this limit will cause an error (Ne ERROR) to be generated. Attempting to use Prg to jump to a program area in which there is no program stored will also result in an error (Go ERROR).

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

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

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

Now let's review the two original programs.

Regular octahedron
P0 Fix、3、:、?、→、A、:、2、×、√、3、×、A、x^2、◢、
√、2、÷、3、×、A、x^y、3 23 steps

Regular tetrahedron
P1 Fix、3、:、?、→、A、:、√、3、×、A、x^2、◢、
√、2、÷、1、2、×、A、x^y、3 22 steps

Total: 45 steps

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

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

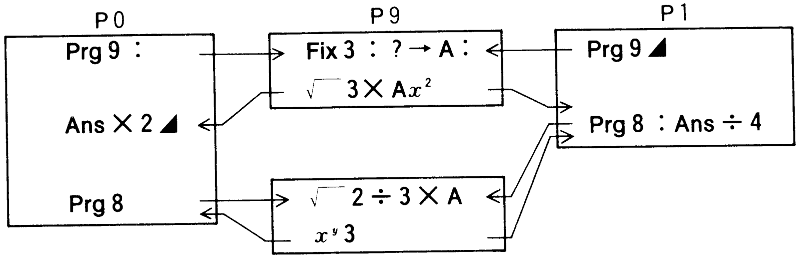
After the common segments have been removed, the remainder of the regular octahedron formula is stored in P0, and that of the regular tetrahedron is stored in P1. Of course, the "Prg 9" and "Prg 8" must be added to jump to subroutines P9 and P8.

P0 Prg、9、:、Ans、×、2、◢、Prg、8 9 steps
P1 Prg、9、◢、Prg、8、:、Ans、÷、4 9 steps
Total: 38 steps

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

Computation of the volumes is also performed in a similar manner. After a jump is made to P8 for computation, execution returns to the main routines. In P0, the program ends after the volume of the octahedron is displayed. In P1, however, the result computed in P8 is divided by four to obtain the volume of the tetrahedron. By using subroutines in this manner, steps can be shortened and programs become neat and easy to read.

The following illustration shows the flow of the program just presented.



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

3-8 ARRAY-TYPE MEMORIES

■ Using array-type memories

Up to this point all of the memories used have been referred to by single alphabetic characters such as A, B, X, or Y.

With the array-type memory introduced here, a memory name (one alphabetic character from A through Z) is appended with a subscript such as [1] or [2].

** Brackets are input by* ALPHA · *and* ALPHA EXP.

Standard memory	Array-type 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.

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

Using standard memories

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

Using array-type memories

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

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

Using standard memories

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

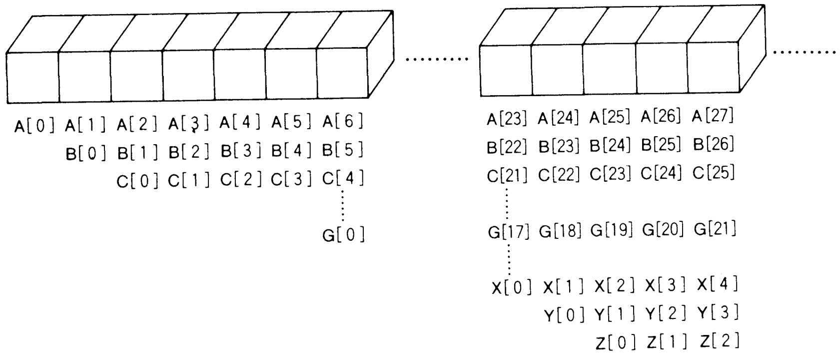
Using array-type memories

Lbl、1、:、?、→、Z、:、A、[、Z、−、1、]、◢、
Goto、1 16 steps

The difference is readily apparent. When using the standard memories, the input value is compared one by one with the value assigned to each memory (i.e. A=1, B=2, . .).
With the array-type memories, the input value is immediately stored in the proper memory determined by "[Z−1]". Formulas (Z−1, A+10, etc.) can even be used for the subscript.

■ Cautions when using array-type memories

When using array-type memories, a subscript is appended to an alphabetic character that represents a standard memory from A through Z.
Therefore, care must be taken to prevent overlap of memories.
The relation is as follows:



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

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

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

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

Operation	**Display**
Prg 0 EXE	1.
EXE	0.
EXE	3.
EXE	4.
EXE	5.

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

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

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

■ Application of the array-type memories

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

Example: Store data x and y in memories. When an x value is input, the corresponding y value is displayed. There will be a total of 15 pieces of data.

Example program 1

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

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

In this program、memories are used as follows:

x data

C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]	C[8]
D	E	F	G	H	I	J	K
C[9]	C[10]	C[11]	C[12]	C[13]	C[14]	C[15]	
L	M	N	O	P	Q	R	

y data

C[16]	C[17]	C[18]	C[19]	C[20]	C[21]	C[22]	C[23]
S	T	U	V	W	X	Y	Z
C[24]	C[25]	C[26]	C[27]	C[28]	C[29]	C[30]	
Z(1)	Z(2)	Z(3)	Z(4)	Z(5)	Z(6)	Z(7)	

Example program 2

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

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

Memories are used as follows:

x data

C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]	C[8]
D	E	F	G	H	I	J	K
C[9]	C[10]	C[11]	C[12]	C[13]	C[14]	C[15]	
L	M	N	O	P	Q	R	

y data

R [1]	R [2]	R [3]	R [4]	R [5]	R [6]	R [7]	R [8]
S	T	U	V	W	X	Y	Z
R [9]	R [10]	R [11]	R [12]	R [13]	R [14]	R [15]	
Z(1)	Z(2)	Z(3)	Z(4)	Z(5)	Z(6)	Z(7)	

In this way, the memory names can be changed. However, since memory names are restricted to the letters from A through Z, the expanded memories (MODE ·) can only be used as array-type memories.
** The memory expansion command (Defm) can be used in a program.*

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

Defm、1、4、:、……

3-9 DISPLAYING ALPHA-NUMERIC CHARACTERS AND SYMBOLS

Alphabetic characters, numbers, computation command symbols, etc. can be displayed as messages. They are enclosed in quotation marks ($\boxed{\text{ALPHA}}$ $\boxed{\text{Prg}}$).

■ Alpha-numeric characters and symbols

- **Characters and symbols displayed when pressed following $\boxed{\text{ALPHA}}$:**
  [、]、 ■、 k、 m、 μ、 n、 p、 space、
  A、 B、 C、 D、 E、 F、 G、 H、 I、 J、 K、 L、 M、 N、
  O、 P、 Q、 R、 S、 T、 U、 V、 W、 X、 Y、 Z
- **Numbers and symbols dipslayed when pressed directly (Comp mode):**
  √、 ^{2}($\boxed{x^2}$)、 ${}^{-1}$($\boxed{x^{-1}}$)、 ${}^{\square}$($\boxed{\circ\,,\,,,}$)、 −($\boxed{(-)}$)、 →、 (、)、
  0、 1、 2、 3、 4、 5、 6、 7、 8、 9、
  .、 E($\boxed{\text{EXP}}$)、 +、 −、 ×、 ÷、 :
- **Symbols displayed when pressed following $\boxed{\text{SHIFT}}$ (Comp mode):**
  ?、 ◢、 ${}_{10}$($\boxed{10^x}$)、 e($\boxed{e^x}$)、 !($\boxed{x!}$)、 ~、 ,、 ;、 ⇒、
  =、 ≠、 ≥、 ≤、 >、 <、 π、 ℙ($\boxed{nPr}$)、 ℂ($\boxed{nCr}$)
- **Symbols displayed when pressed directly (Base-n mode):**
  𝔸、 𝔹、 ℂ、 𝔻、 𝔼、 𝔽
- **Symbols displayed when pressed following $\boxed{\text{SHIFT}}$ (Base-n mode):**
  d、 h、 b、 o
- **Symbol displayed when pressed following $\boxed{\text{SHIFT}}$ (SD mode):**
  $\bar{x}$
- **Symbols displayed when pressed following $\boxed{\text{SHIFT}}$ (LR mode):**
  $\bar{x}$、 $\bar{y}$、 A、 B、 r、 $\hat{x}$、 $\hat{y}$
- **Symbols displayed when pressed following $\boxed{\text{SHIFT}}$ $\boxed{\text{MODE}}$ (all modes except Base-n mode):**
  °($\boxed{4}$)、 ʳ($\boxed{5}$)、 ᵍ($\boxed{6}$)

**Alphanumeric characters and symbols are one-letter commands. Therefore, such expressions as "sin" or "$x\sigma_n$" cannot be used.*

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

Lbl、 1、 :、 ?、 →、 X、 :、 ?、 →、 Y、 :、 ······

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

Lbl、 1、 :、 $\underline{”、X、=、”}$、 :、 ?、 →、 X、 :、
◢ not used

$\underline{”、Y、=、”}$、 :、 ?、 →、 Y、 :、 ······
◢ not used

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

$\boxed{\text{Prg}}$ 1 $\boxed{\text{EXE}}$	X=?
10 $\boxed{\text{EXE}}$	Y=?
⋮	⋮

**Always follow a message with "◢" (to display the message) when the formula continues following the message.*

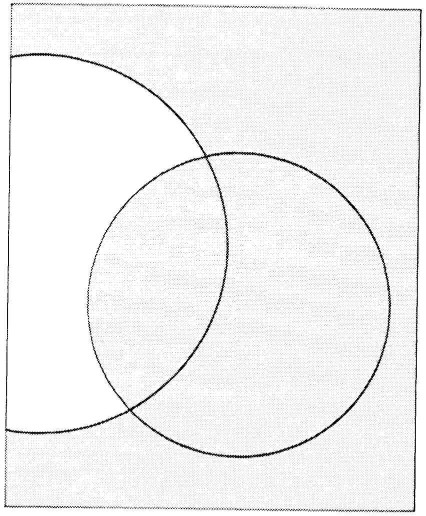
When a message exceeds 12 characters, it is displayed by shifting from right to left, one character at a time.

Example:

For: ”、A、B、C、D、E、F、G、H、I、J、K、L、M、N、”、◢、······

Shift to the left,	ABCDEFGHIJKL	MN←Still not displayed.
off of the display.→A	BCDEFGHIJKLM	N
AB	CDEFGHIJKLMN	

PROGRAM LIBRARY



CASIO PROGRAM SHEET

Program for	No.
Prime factor analysis	1

Description

Prime factors of arbitrary positive integers are produced.

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

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

〈Overview〉

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

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

Example

〈1〉

$119 = 7 \times 17$

〈2〉

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

〈3〉

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

Preparation and operation

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

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

No. 1

Line	MODE 2 Program															Notes	Number of steps
1	Mcl	:															2
2	Lbl	0	:	”	M	”	:	?	→	A	:	Goto	2	:			16
3	Lbl	1	:	2	◢	A	÷	2	→	A	:	A	=	1	⇒		31
4	Goto	9	:														34
5	Lbl	2	:	Frac	(	A	÷	2	)	=	0	⇒	Goto	1	:		49
6	3	→	B	:													53
7	Lbl	3	:	√	A	+	1	→	C	:							63
8	Lbl	4	:	B	≥	C	⇒	Goto	8	:	Frac	(	A	÷	B		78
9	)	=	0	⇒	Goto	6	:										85
10	Lbl	5	:	B	+	2	→	B	:	Goto	4	:					97
11	Lbl	6	:	A	÷	B	×	B	−	A	=	0	⇒	Goto	7		112
12	:	Goto	5	:													116
13	Lbl	7	:	B	◢	A	÷	B	→	A	:	Goto	3	:			130
14	Lbl	8	:	A	◢												135
15	Lbl	9	:	”	E	N	D	”	◢	Goto	0						146
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents

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

CASIO PROGRAM SHEET

Program for	No.
Greatest common measure	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⟩

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

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

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

$$k = 2,\ 3 \ldots\ldots\ldots$$

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

Example

	⟨1⟩	⟨2⟩	⟨3⟩
When	$a = 238$	$a = 23345$	$a = 522952$
	$b = 374$	$b = 9135$	$b = 3208137866$
	↓	↓	↓
	$c = 34$	$c = 1015$	$c = 998$

Preparation and operation

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

Step	Key operation	Display	Step	Key operation	Display
1	Prg 0 EXE	A ?	11		
2	238 EXE	B ?	12		
3	374 EXE	34.	13		
4	EXE	A ?	14		
5	23345 EXE	B ?	15		
6	9135 EXE	1015.	16		
7	EXE	A ?	17		
8	522952 EXE	B ?	18		
9	3208137866 EXE	998.	19		
10			20		

No.
2

Line	MODE 2						Program									Notes	Number of steps
1	Lbl	1	:	"	A	"	:	?	→	A	:	"	B	"	:		15
2	?	→	B	:													19
3	Abs	A	→	A	:	Abs	B	→	B	:							29
4	B	<	A	⇒	Goto	2	:										36
5	A	→	C	:	B	→	A	:	C	→	B	:					48
6	Lbl	2	:	(−)	(	Int	(	A	÷	B	)	×	B	−	A		63
7	)	→	C	:													67
8	C	=	0	⇒	Goto	3	:										74
9	B	→	A	:	C	→	B	:	Goto	2	:						85
10	Lbl	3	:	B	◢	Goto	1										92
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

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
Definite integrals using the Simpson's rule

No. 3

Description

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

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

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

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

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

Example

〈1〉 $a=0,\ b=1,\ 2m=10$

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

〈2〉 $a=2,\ b=5,\ 2m=20$

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

Preparation and operation

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

Step	Key operation	Display	Step	Key operation	Display
1	Prg 0 EXE	A ?	11		
2	0 EXE	B ?	12		
3	1 EXE	2 M ?	13		
4	10 EXE	0.7853981537	14		
5	EXE	A ?	15		
6	2 EXE	B ?	16		
7	5 EXE	2 M ?	17		
8	20 EXE	0.2662526769	18		
9			19		
10			20		

Line	MODE 2 Program															Notes	Number of steps
1	P 0																
2	Lbl	1	:	Mcl	:												5
3	"	A	"	:	?	→	A	:	"	B	"	:	?	→	B		20
4	:	"	2	M	"	:	?	→	M	:							30
5	A	→	G	:	Prg	1	:	P	→	I	:	(	B	−	A		45
6	)	÷	M	→	D	:	M	÷	2	→	O	:					57
7	Lbl	2	:	G	+	D	→	G	:	Prg	1	:	I	+	P		72
8	×	4	→	I	:												77
9	G	+	D	→	G	:	Prg	1	:	I	+	P	×	2	→		92
10	I	:	O	−	1	→	O	:									100
11	O	≠	0	⇒	Goto	2	:										107
12	B	→	G	:	Prg	1	:	I	−	P	→	I	:				120
13	D	×	I	÷	3	◢											126
14	Goto	1															128
15																	
16	P 1																
17	1	÷	(	G	×	G	+	1	)	→	P						11
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

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

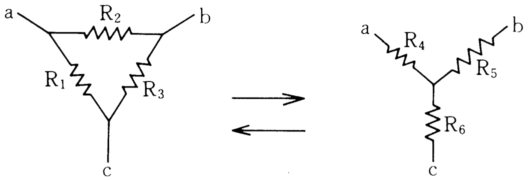
CASIO PROGRAM SHEET

Program for

△ ⟷ Y transformation

No. 4

Description



1) △→ 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 →△

$$R_1=\frac{R_4R_5+R_5R_6+R_6R_4}{R_5}$$

$$R_2=\frac{R_4R_5+R_5R_6+R_6R_4}{R_6}$$

$$R_3=\frac{R_4R_5+R_5R_6+R_6R_4}{R_4}$$

Example

〈1〉

R_1=12(Ω)

R_2=47(Ω)

R_3=82(Ω)

〈2〉

R_4=100(Ω)

R_5=150(Ω)

R_6=220(Ω)

Preparation and operation

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

Step	Key operation	Display	Step	Key operation	Display
1	Prg 0 EXE	◢→Y:1, Y→◢:2?	11	EXE	◢→Y:1, Y→◢:2?
2	1 EXE	R 1= ?	12	2 EXE	R 4= ?
3	12 EXE	R 2= ?	13	100 EXE	R 5= ?
4	47 EXE	R 3= ?	14	150 EXE	R 6= ?
5	82 EXE	R 4= ?	15	220 EXE	R 1=
6	EXE	4.	16	EXE	466.6666667
7	EXE	R 5=	17	EXE	R 2=
8	EXE	27.33333333	18	EXE	318.1818182
9	EXE	R 6=	19	EXE	R 3=
10	EXE	6.978723404	20	EXE	700.

No. 4

Line	MODE 2 Program															Notes	Number of steps
1	Lbl	1	:	"	◢	→	Y	:	1	,	Y	→	◢	:	2		15
2	"	:	?	→	N	:											21
3	N	=	2	⇒	Goto	2	:	N	≠	1	⇒	Goto	1	:			35
4	"	R	1	=	"	:	?	→	A	:							45
5	"	R	2	=	"	:	?	→	B	:							55
6	"	R	3	=	"	:	?	→	C	:							65
7	A	+	B	+	C	→	D	:									73
8	"	R	4	=	"	◢	A	×	B	÷	D	◢					85
9	"	R	5	=	"	◢	B	×	C	÷	D	◢					97
10	"	R	6	=	"	◢	A	×	C	÷	D	◢					109
11	Goto	1	:														112
12	Lbl	2	:														115
13	"	R	4	=	"	:	?	→	E	:							125
14	"	R	5	=	"	:	?	→	F	:							135
15	"	R	6	=	"	:	?	→	G	:							145
16	E	×	F	+	F	×	G	+	G	×	E	→	H	:			159
17	"	R	1	=	"	◢	H	÷	F	◢							169
18	"	R	2	=	"	◢	H	÷	G	◢							179
19	"	R	3	=	"	◢	H	÷	E	◢							189
20	Goto	1															191
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

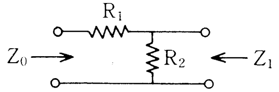
Memory contents

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

Program for	No.
Minimum loss matching	5

Description

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



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

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

Example

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

Preparation and operation

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

Step	Key operation	Display	Step	Key operation	Display
1	Prg 0 EXE	Z 0= ?	11		
2	500 EXE	Z 1= ?	12		
3	200 EXE	R 1=	13		
4	EXE	387.2983346	14		
5	EXE	R 2=	15		
6	EXE	258.1988898	16		
7	EXE	LMIN =	17		
8	EXE	8.961393328	18		
9			19		
10			20		

No. 5

Line	MODE 2					Program										Notes	Number of steps
1	"	Z	0	=	"	:	?	→	Y	:							10
2	"	Z	1	=	"	:	?	→	Z	:							20
3	√	(	1	−	Z	÷	Y	)	→	A	:						31
4	Y	×	A	→	R	:	Z	÷	A	→	S	:	Y	÷	Z		46
5	→	B	:	2	0	×	log	(	√	B	+	√	(	B	−		61
6	1	)	)	→	T	:											67
7	"	R	1	=	"	◢	R	◢									75
8	"	R	2	=	"	◢	S	◢									83
9	"	L	M	I	N	=	"	◢	T								92
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents

A	$\sqrt{1-\frac{Z_1}{Z_0}}$	H		O		V	
B	$\frac{Z_0}{Z_1}$	I		P		W	
C		J		Q		X	
D		K		R	R 1	Y	Z_0
E		L		S	R 2	Z	Z_1
F		M		T	Lmin		
G		N		U			

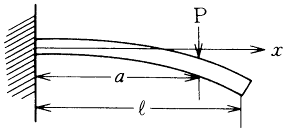
CASIO PROGRAM SHEET

Program for

Cantilever under concentrated load

No. 6

Description



E : Young's modulus 〔kg/mm^2〕

I : Geometrical moment of inertia 〔mm^4〕

a : Distance of concentrated load from support 〔mm〕

P : Load 〔kg〕

x : Distance of point of interest from the support 〔mm〕

Deflection y 〔mm〕, Angle of deflection s 〔°〕, Bending moment M 〔kg · mm〕

① $\ell > x > a$

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

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

M = 0 (shearing load Ws = 0)

② $x \leqq a$

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

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

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

Example

E = 4000 kg/mm^2
I = 5 mm^4
a = 30 mm
P = 2 kg

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

Preparation and operation

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

Step	Key operation	Display	Step	Key operation	Display
1	Prg 0 EXE	E = ?	11	EXE	−10.
2	4000 EXE	I = ?	12	EXE	X = ?
3	5 EXE	A = ?	13	32 EXE	Y =
4	30 EXE	P = ?	14	EXE	−0.99
5	2 EXE	X = ?	15	EXE	S =
6	25 EXE	Y =	16	EXE	−2.57657183
7	EXE	−0.6770833333	17	EXE	M =
8	EXE	S =	18	EXE	0.
9	EXE	−2.505092867	19	EXE	X = ?
10	EXE	M =	20	Repeat from step 6.	

No. 6

Line	MODE 2 Program															Notes	Number of steps
1	Deg	:	”	E	=	”	:	?	→	E	:	”	I	=	”		15
2	:	?	→	I	:	”	A	=	”	:	?	→	A	:	”		30
3	P	=	”	:	?	→	P	:									38
4	Lbl	1	:	”	X	=	”	:	?	→	X	:					50
5	X	≤	A	⇒	Goto	2	:										57
6	”	Y	=	”	◢	P	×	A	x^2	÷	(	2	×	E	×		72
7	I	)	×	(	A	÷	3	−	X	)	◢						83
8	”	S	=	”	◢	$\tan^{-1}$	(	(−)	P	×	A	x^2	÷	(	2		98
9	×	E	×	I	)	)	◢	”	M	=	”	◢	0	◢			112
10	Goto	1	:														115
11	Lbl	2	:														118
12	”	Y	=	”	◢	P	×	X	x^2	÷	(	2	×	E	×		133
13	I	)	×	(	X	÷	3	−	A	)	◢						144
14	”	S	=	”	◢	$\tan^{-1}$	(	P	×	X	÷	(	2	×	E		159
15	×	I	)	×	(	X	−	2	×	A	)	)	◢				172
16	”	M	=	”	◢	P	×	(	X	−	A	)	◢				185
17	Goto	1															187
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents

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

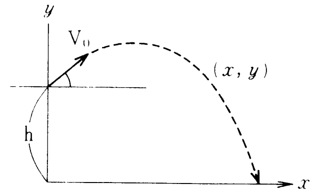
CASIO PROGRAM SHEET

Program for

Parabolic movement

No. 7

Description



$x = (V_0 \cos a) \cdot t$

$y = (V_0 \sin a) \cdot t - \frac{1}{2} gt^2 + h$

$g = 9.8$ 〔$\mathrm{m/s^2}$〕

V_0 〔m/s〕

a 〔°〕

Δ t 〔sec.〕

h 〔m〕

Example

Initial velocity $V_0 = 130$(m/sec.)

Initial angle a =25(°)

Height h = 0 (m)

Δ t =0.5(sec.)

Plot the trace of movement in intervals of Δt.

Preparation and operation

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

Step	Key operation	Display	Step	Key operation	Display
1	Prg 0 EXE	V0= ?	11	EXE	T =
2	130 EXE	A = ?	12	EXE	0.5
3	25 EXE	H = ?	13	EXE	X =
4	0 EXE	◢ T = ?	14	EXE	58.91000616
5	0.5 EXE	T =	15	EXE	Y =
6	EXE	0.	16	EXE	26.24518701
7	EXE	X =	17	EXE	T =
8	EXE	0.	18	Repeat from step 12.	
9	EXE	Y =	19		
10	EXE	0.	20		

No. 7

Line	MODE 2 Program															Notes	Number of steps
1	Deg	:	0	→	S	:											6
2	"	V	0	=	"	:	?	→	V	:	"	A	=	"	:		21
3	?	→	A	:	"	H	=	"	:	?	→	H	:	"	◢		36
4	T	=	"	:	?	→	T	:									44
5	Lbl	1	:	V	×	cos	A	×	S	→	X	:	V	×	sin		59
6	A	×	S	−	9	.	8	×	S	x^2	÷	2	+	H	→		74
7	Y	:															76
8	"	T	=	"	◢	S	◢	S	+	T	→	S	:				89
9	"	X	=	"	◢	X	◢	"	Y	=	"	◢	Y	◢			103
10	Y	≥	0	⇒	Goto	1											109
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents

A	a	H	h	O		V	V_0
B		I		P		W	
C		J		Q		X	
D		K		R		Y	
E		L		S		Z	
F		M		T	Δ t		
G		N		U			

CASIO PROGRAM SHEET

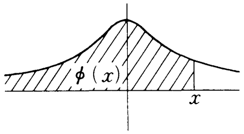
Program for **Normal distribution**

No. 8

Description

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

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



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

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

$\phi(x)\fallingdotseq 1-\phi t(c_1t+c_2t^2+c_3t^3+c_4t^4+c_5t^5)$

$P=0.2316419$

$C_1=0.31938153$

$C_2=-0.356563782$

$C_3=1.78147937$

$C_4=-1.821255978$

$C_5=1.330274429$

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 in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prg 0 EXE	X = ?	11		
2	1.18 EXE	PX =	12		
3	EXE	0.880999696	13		
4	Prg 0 EXE	X = ?	14		
5	0.7 EXE	PX =	15		
6	EXE	0.7580361367	16		
7			17		
8			18		
9			19		
10			20		

No. 8

Line	MODE 2 Program															Notes	Number of steps
1	"	X	=	"	:	?	→	X	:								9
2	1	÷	(	1	+	0	.	2	3	1	6	4	1	9	×		24
3	X	)	→	T	:	1	÷	√	(	2	×	π	)	×	e^x		39
4	(	(−)	X	x^2	÷	2	)	→	Q	:							49
5	"	P	X	=	"	◢	1	−	Q	×	(	0	.	3	1		64
6	9	3	8	1	5	3	×	T	+	(−)	0	.	3	5	6		79
7	5	6	3	7	8	2	×	T	x^2	+	1	.	7	8	1		94
8	4	7	9	3	7	×	T	x^y	3	+	(−)	1	.	8	2		109
9	1	2	5	5	9	7	8	×	T	x^y	4	+	1	.	3		124
10	3	0	2	7	4	4	2	9	×	T	x^y	5	)				137
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents

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

CASIO PROGRAM SHEET

Program for	No.

Description

Example

Preparation and operation

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

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

No.

Line	MODE 2 Program	Notes	Number of steps
1			
2			
3			
4			
5			
6			
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		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.

Description

Example

Preparation and operation

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

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

No.

Line	MODE 2 Program	Notes	Number of steps
1			
2			
3			
4			
5			
6			
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		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.

Description

Example

Preparation and operation

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

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

No.

Line	MODE 2 Program	Notes	Number of steps
1			
2			
3			
4			
5			
6			
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		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.

Description

Example

Preparation and operation

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

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

No.

Line	MODE 2 Program	Notes	Number of steps
1			
2			
3			
4			
5			
6			
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		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.

Description

Example

Preparation and operation

- ●Store the program written on the next page.
- ●Execute the program as shown below in the RUN mode ($\boxed{\text{MODE}}$ $\boxed{1}$).

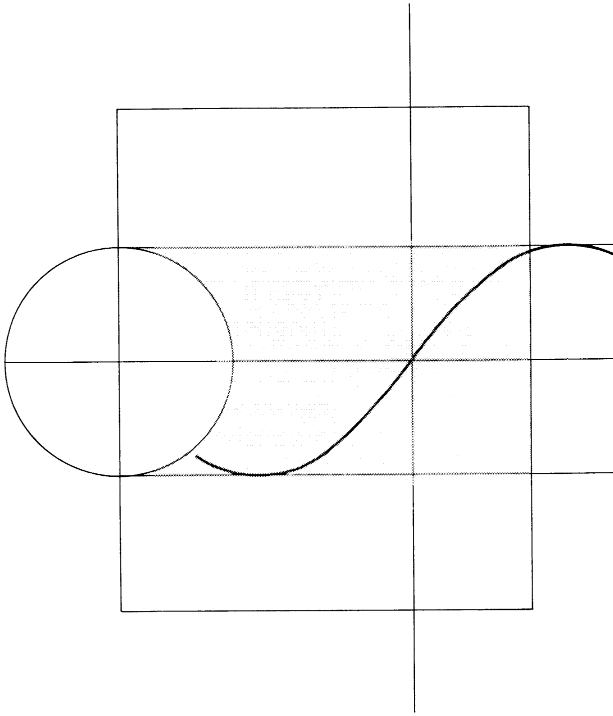
Step	Key operation	Display	Step	Key operation	Display
1	☐		11	☐	
2	☐		12	☐	
3	☐		13	☐	
4	☐		14	☐	
5	☐		15	☐	
6	☐		16	☐	
7	☐		17	☐	
8	☐		18	☐	
9	☐		19	☐	
10	☐		20	☐	

No.

Line	$\boxed{\text{MODE}}$ $\boxed{2}$ Program	Notes	Number of steps
1			
2			
3			
4			
5			
6			
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		V	
	B		I		P		W	
	C		J		Q		X	
	D		K		R		Y	
	E		L		S		Z	
	F		M		T			
	G		N		U			

REFERENCE MATERIAL



■ Manual computations

Mode specification	Comp mode ($\boxed{\text{MODE}}\ \boxed{+}$)	Four arithmetic computations and function computations.
	Base-n mode ($\boxed{\text{MODE}}\ \boxed{-}$)	Binary, octal, decimal, hexadecimal conversions and computations, logical operations.
	SD mode ($\boxed{\text{MODE}}\ \boxed{\times}$)	Standard deviation computations (1-variable statistical computations).
	LR mode ($\boxed{\text{MODE}}\ \boxed{\div}$)	Regression computations (paired variable statistical computations).
Functions	Type A functions	Function command input immediately before numeric value. (sin, cos, tan, $\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$, sinh, cosh, tanh, $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$, log, ln, e^x, 10^x, $\sqrt{\ }$, $\sqrt[3]{\ }$, Abs, Int, Frac)
	Type B functions	Function command input immediately after numeric value. 〔x^2, x^{-1}, $x!$〕
	Paired variable functions	Function command input between two numeric values. Numeric value enclosed in parentheses input immediately after function command. (A x^y B (A to the Bth power), B $\sqrt[x]{\ }$ A (A to the 1/Bth power), A $n\text{P}r$ B, A $n\text{C}r$ B, Pol(A,B), Rec(A,B)) **A and B are numeric values.*
	Immediately executed functions	Displayed value changed with each press of a key. 〔ENG, $\overleftarrow{\text{ENG}}$, $\overleftarrow{^{\circ}\ '\ ''}$〕

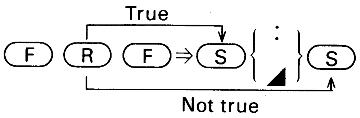
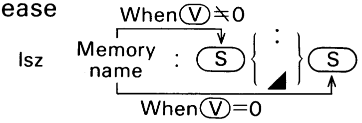
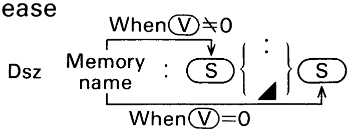
Binary, octal, decimal, hexadecimal computations	Setting number system	Decimal ············ $\boxed{\text{Dec}}\ \boxed{\text{EXE}}$ ($\boxed{\text{Dec}} = \boxed{\sqrt{\ }}$) Hexadecimal ······ $\boxed{\text{Hex}}\ \boxed{\text{EXE}}$ ($\boxed{\text{Hex}} = \boxed{x^2}$) Binary ··············· $\boxed{\text{Bin}}\ \boxed{\text{EXE}}$ ($\boxed{\text{Bin}} = \boxed{\text{log}}$) Octal ················· $\boxed{\text{Oct}}\ \boxed{\text{EXE}}$ ($\boxed{\text{Oct}} = \boxed{\text{ln}}$)
	Number system specification	Number system for the numeric value entered immediately after can be specified regardless of the currently set number system. To specify: Decimal··············· $\boxed{\text{SHIFT}}\ \boxed{\text{d}}$ ($\boxed{\text{d}} = \boxed{\sqrt{\ }}$) Hexadecimal ······ $\boxed{\text{SHIFT}}\ \boxed{\text{h}}$ ($\boxed{\text{h}} = \boxed{x^2}$) Binary ··············· $\boxed{\text{SHIFT}}\ \boxed{\text{b}}$ ($\boxed{\text{b}} = \boxed{\text{log}}$) Octal ················· $\boxed{\text{SHIFT}}\ \boxed{\text{o}}$ ($\boxed{\text{o}} = \boxed{\text{ln}}$)
	Logical operations	A input numeric value converted to binary and each bit computed. Result converted back to number system used for input, and then displayed. Not .... Reverse of each bit and .... Logical product of each bit or ...... Logical sum of each bit
Standard deviation computations	Data clear	$\boxed{\text{SHIFT}}\ \boxed{\text{Scl}}\ \boxed{\text{EXE}}$ ($\boxed{\text{Scl}} = \boxed{\text{AC}}$)
	Data input	Data 〔;frequency〕 $\boxed{\text{DT}}$ ($\boxed{\text{DT}} = \boxed{\sqrt[x]{\ }}$) **Frequency can be omitted.*
	Data deletion	Data 〔;frequency〕 $\boxed{\text{Cl}}$ ($\boxed{\text{Cl}} = \boxed{x^y}$) **Frequency can be omitted.*
	Result display	Number of data (n) ··············· $\boxed{\text{ALPHA}}\ \boxed{n}\ \boxed{\text{EXE}}$ ($\boxed{n} = \boxed{3}$) Sum (Σx) ······ $\boxed{\text{ALPHA}}\ \boxed{\Sigma x}\ \boxed{\text{EXE}}$ ($\boxed{\Sigma x} = \boxed{2}$) Sum of squares (Σx^2) ················ $\boxed{\text{ALPHA}}\ \boxed{\Sigma x^2}\ \boxed{\text{EXE}}$ ($\boxed{\Sigma x^2} = \boxed{1}$) Mean ($\bar{x}$)······· $\boxed{\text{SHIFT}}\ \boxed{\bar{x}}\ \boxed{\text{EXE}}$ ($\boxed{\bar{x}} = \boxed{1}$) Population standard deviation ($x\sigma_n$) ··············· $\boxed{\text{SHIFT}}\ \boxed{x\sigma_n}\ \boxed{\text{EXE}}$ ($\boxed{x\sigma_n} = \boxed{2}$) Sample standard deviation ($x\sigma_{n-1}$) ··············· $\boxed{\text{SHIFT}}\ \boxed{x\sigma_{n-1}}\ \boxed{\text{EXE}}$ ($\boxed{x\sigma_{n-1}} = \boxed{3}$)

Regression computations	Data clear	$\boxed{\text{SHIFT}}$ $\boxed{\text{Scl}}$ $\boxed{\text{EXE}}$ ($\boxed{\text{Scl}} = \boxed{\text{AC}}$)
	Data input	x data, y data 〔; frequency〕 $\boxed{\text{DT}}$ ($\boxed{\text{DT}} = \boxed{\sqrt[x]{\ }}$) **Frequency can be omitted.*
	Data deletion	x data, y data 〔; frequency〕 $\boxed{\text{Cl}}$ ($\boxed{\text{Cl}} = \boxed{x^y}$) **Frequency can be omitted.*
	Result display	Number of data (n) ·················· $\boxed{\text{ALPHA}}$ $\boxed{n}$ $\boxed{\text{EXE}}$ ($\boxed{n} = \boxed{3}$) Sum of x (Σx)··· $\boxed{\text{ALPHA}}$ $\boxed{\Sigma x}$ $\boxed{\text{EXE}}$ ($\boxed{\Sigma x} = \boxed{2}$) Sum of y (Σy)··· $\boxed{\text{ALPHA}}$ $\boxed{\Sigma y}$ $\boxed{\text{EXE}}$ ($\boxed{\Sigma y} = \boxed{5}$) Sum of squares of x (Σx^2) ··············· $\boxed{\text{ALPHA}}$ $\boxed{\Sigma x^2}$ $\boxed{\text{EXE}}$ ($\boxed{\Sigma x^2} = \boxed{1}$) Sum of squares of y (Σy^2) ··············· $\boxed{\text{ALPHA}}$ $\boxed{\Sigma y^2}$ $\boxed{\text{EXE}}$ ($\boxed{\Sigma y^2} = \boxed{4}$) Sum of products of x and y (Σxy) ············· $\boxed{\text{ALPHA}}$ $\boxed{\Sigma xy}$ $\boxed{\text{EXE}}$ ($\boxed{\Sigma xy} = \boxed{6}$) Mean of x ($\bar{x}$)··· $\boxed{\text{SHIFT}}$ $\boxed{\bar{x}}$ $\boxed{\text{EXE}}$ ($\boxed{\bar{x}} = \boxed{1}$) Mean of y ($\bar{y}$)··· $\boxed{\text{SHIFT}}$ $\boxed{\bar{y}}$ $\boxed{\text{EXE}}$ ($\boxed{\bar{y}} = \boxed{4}$) Population standard deviation of x ($x\sigma_n$) ········ $\boxed{\text{SHIFT}}$ $\boxed{x\sigma_n}$ $\boxed{\text{EXE}}$ ($\boxed{x\sigma_n} = \boxed{2}$) Population standard deviation of y ($y\sigma_n$) ········ $\boxed{\text{SHIFT}}$ $\boxed{y\sigma_n}$ $\boxed{\text{EXE}}$ ($\boxed{y\sigma_n} = \boxed{5}$) Sample standard deviation of x ($x\sigma_{n-1}$) ······ $\boxed{\text{SHIFT}}$ $\boxed{x\sigma_{n-1}}$ $\boxed{\text{EXE}}$ ($\boxed{x\sigma_{n-1}} = \boxed{3}$) Sample standard deviation of y ($y\sigma_{n-1}$) ······ $\boxed{\text{SHIFT}}$ $\boxed{y\sigma_{n-1}}$ $\boxed{\text{EXE}}$ ($\boxed{y\sigma_{n-1}} = \boxed{6}$) Constant term of regression formula (A) ············ $\boxed{\text{SHIFT}}$ $\boxed{\text{A}}$ $\boxed{\text{EXE}}$ ($\boxed{\text{A}} = \boxed{7}$) Regression coefficient (B) ··············· $\boxed{\text{SHIFT}}$ $\boxed{\text{B}}$ $\boxed{\text{EXE}}$ ($\boxed{\text{B}} = \boxed{8}$) Correlation coefficient (r) ·················· $\boxed{\text{SHIFT}}$ $\boxed{\text{r}}$ $\boxed{\text{EXE}}$ ($\boxed{\text{r}} = \boxed{9}$) Estimated value of x ($\hat{x}$) ········ y data $\boxed{\text{SHIFT}}$ $\boxed{\hat{x}}$ $\boxed{\text{EXE}}$ ($\boxed{\hat{x}} = \boxed{\times}$) Estimated value of y ($\hat{y}$) ········ x data $\boxed{\text{SHIFT}}$ $\boxed{\hat{y}}$ $\boxed{\text{EXE}}$ ($\boxed{\hat{y}} = \boxed{\div}$)

Special functions	Ans function	The latest result obtained in manual or program computations is stored in memory. It is recalled by pressing $\boxed{\text{Ans}}$. **Mantissa of numeric value is 10 digits.*
	Replay function	•After computation results are obtained, the computation formula can be recalled by pressing either $\boxed{\Leftarrow}$ or $\boxed{\Rightarrow}$. •If an error is generated, pressing either $\boxed{\Leftarrow}$ or $\boxed{\Rightarrow}$ will cancel the error and the point where the error was generated will be indicated by a blinking cursor.
	Multistatement function	Colons are used to join a series of statements or computation formulas. If joined using "◢", the computation result to that point is displayed.
	Memory expansion	The Number of memories can be expanded from the standard 26. Memories can be expanded in units of one up to 68 (for a total of 94). Eight steps are required for one memory expansion. $\boxed{\text{MODE}}$ $\boxed{\cdot}$ number of memories to be expanded $\boxed{\text{EXE}}$

Program computations

Program input	Input mode	WRT mode (MODE 2)
	Computation mode	Mode that conforms with program specified by: MODE +, MODE −, MODE ×, or MODE ÷.
	Program area specification	Cursor is moved to the desired program area number (P0 through P9) using ⇦ and ⇨, and EXE is pressed.
Program execution	Execution mode	RUN mode (MODE 1)
	Program area specification	Execution starts with Prg program area No. EXE.
Program editing	Input mode	WRT mode (MODE 2)
	Program area specification	Cursor is moved to the desired program area number (P0 through P9) using ⇦ and ⇨, and EXE or SHIFT EXE are pressed.
	Editing	Cursor is moved to position to be edited using ⇦ and ⇨. •Press correct key for corrections. •Press DEL for deletions. •Press SHIFT INS to open a space ([ ]) for insertion.
Program erasing	Erase mode	PCL mode (MODE 3)
	Erasing a program in a single program area	Cursor is moved to the desired program area number (P0 through P9) using ⇦ and ⇨, and AC is pressed.
	Erasing the programs in all program areas	Press SHIFT Mcl.

Program commands	Unconditional jump	Program execution jumps to the Lbl n which corresponds to Goto n. *n = *0 through 9*
	Conditional jump	If conditional expression is true, the statement after "⇒" is executed. If not true, execution jumps to the statement following next ":" or "◢".  Ⓕ: Formula Ⓡ: Relational operator Ⓢ: Statement * *The relational operator is :* =, ≠, >, <, ≥ or ≤.
	Count jump	The value in a memory is increased or decreased. If the value does not equal 0, the next statement is executed. If it is 0, a jump is performed to the statement following the next ":" or "◢". Increase  Decrease  Ⓢ: Statement Ⓥ: Value in memory
	Subroutines	Program execution jumps from main routine to subroutine indicated by Prg n (n =0 through 9). After execution of the subroutine, execution returns to the point following Prg n in the original program area.

Error messages

Message	Meaning	Countermeasure
Syn ERROR	① Computation formula contains an error. ② Formula in a program contains an error.	① Use ⇦ or ⇨ to display the point where the error was generated and correct it. ② Use ⇦ or ⇨ to display the point where the error was generated, press AC and then correct the program in the WRT mode.
Ma ERROR	① Computation result exceeds computation range. ② Computation is performed outside the input range of a function. ③ Illogical operation (division by zero, etc.)	①②③ Check the input numeric value and correct it. When using memories, check that the numeric values stored in memories are correct.
Go ERROR	① No corresponding Lbl n to Goto n. ② No program stored in program area Pn which corresponds to Prg n.	① Correctly input a Lbl n to correspond to the Goto n, or delete the Goto n if not required. ② Store a program in program area Pn to correspond to Prg n, or delete the Prg n if not required.
Ne ERROR	• Nesting of subroutines by Prg n exceeds 9 levels.	• Ensure that Prg n is not used to return from subroutines to main routine. If used, delete any unnecessary Prg n. • Trace the subroutine jump destinations and ensure that no jumps are made back to the original program area. Ensure that returns are made correctly.
Stk ERROR	• Execution of computations that exceed the capacity of the stack for numeric values or stack for computations.	• Simplify the formulas to keep stacks within 8 levels for the numeric values and 20 levels for the computations. • Divide the formula into two or more parts.
Mem ERROR	• Attempt to use a memory such as Z[5] when no memory has been expanded.	• Expand memories using MODE · (Defm). • Use memories within the current number of memories.

■ Input range of functions (general principles)

Function name	Input range
$\sin x$, $\cos x$, $\tan x$	$\lvert x\rvert < 1440^\circ$ (8πrad, 1600grad.)
$\sin^{-1}x$, $\cos^{-1}x$	$\lvert x\rvert \leqq 1$
$\tan^{-1}x$	$\lvert x\rvert < 10^{100}$
e^x	$-10^{100} < x \leqq 230.2585092$
$\sinh x$, $\cosh x$	$-10^{100} < x \leqq 230.2585092$
$\tanh x$	$\lvert x\rvert < 10^{100}$
$\sinh^{-1}x$	$\lvert x\rvert < 5^{99}$
$\cosh^{-1}x$	$1 \leqq x < 5^{99}$
$\tanh^{-1}x$	$\lvert x\rvert < 1$
$\log x$, $\ln x$	$0 < x < 10^{100}$
10^x	$-10^{100} < x < 100$
$\sqrt{x}$	$0 \leqq x < 10^{100}$
x^2	$\lvert x\rvert < 10^{50}$
x^{-1} $(1/x)$	$\lvert x\rvert < 10^{100}$, $x \neq 0$
$\sqrt[3]{x}$	$\lvert x\rvert < 10^{100}$
$x!$	$0 \leqq x \leqq 69$ (x is an integer.)
x^y	When $x < 0$, y is a natural number.
$\sqrt[y]{x}$ $(x^{1/y})$	$x \geqq 0$, $y \neq 0$
$n\mathrm{P}r$, $n\mathrm{C}r$	$0 \leqq r \leqq n$, $n < 10^{10}$ (n and r are positive integers or 0.)
Pol (x,y)	$\lvert x\rvert < 10^{100}$, $\lvert y\rvert < 10^{100}$ However, $\sqrt{x^2+y^2} < 10^{100}$.
Rec (r,θ)	$\lvert r\rvert < 10^{100}$, $\lvert\theta\rvert < 1440^\circ$ (8πrad, 1600grad.)
Binary number	(Positive)$1111111111111111111111111111111 \geqq x \geqq 0$ (Negative)$11111111111111111111111111111111 \geqq x \geqq 10000000000000000000000000000000$
Octal number	(Positive)$17777777777 \geqq x \geqq 0$ (Negative)$37777777777 \geqq x \geqq 20000000000$
Hexadecimal number	(Positive)$7\mathrm{FFFFFFF} \geqq x \geqq 0$ (Negative)$\mathrm{FFFFFFFF} \geqq x \geqq 80000000$
Decimal→ sexagesimal	$\lvert x\rvert \leqq 2777777.777$. If degrees, minutes and seconds exceed a total of 9 digits, the higher (degrees, minutes) values will be given priority, and displayed in 9 digits.
Statistical computation	$\lvert x\rvert < 10^{50}$, $\lvert y\rvert < 10^{50}$, $\lvert n\rvert < 10^{100}$

- * As a rule, the accuracy of a result is ±1 at the 10th digit.
- * Errors may be cumulative with such internal continuous computations with the functions, x^y, $x^{1/y}$, $x!$, $\sqrt[3]{x}$, $n\mathrm{P}r$, $n\mathrm{C}r$ and accuracy is sometimes affected.
- * In $\tan x$, $\lvert x\rvert \neq 90^\circ \times (2n+1)$, $\lvert x\rvert \neq \frac{\pi}{2}\mathrm{rad} \times (2n+1)$, $\lvert x\rvert \neq 100\mathrm{grad.}(2n+1)$ (n is an integer.)
- * With $\sinh x$ and $\tanh x$, when $x=0$, errors are cumulative and accuracy is affected.
- * Certain combinations or permutations may cause errors due to overflow during internal calculations.

SPECIFICATIONS

Model: fx-4000P

Computations

Basic computation 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 of angular measurement: degrees, radians, grads), hyperbolic/inverse hyperbolic functions, logarithmic/exponential functions, reciprocals, factorials, square roots, cube roots, powers, roots, squares, decimal-sexagesimal conversions, binary-octal-hexadecimal conversions/computations, coordinate transformations, permutations/combinations, π, random numbers, absolute values, integers, fractions.
Statistical computation 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
Memories:	26 standard (94 maximum)
Computation 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.

Programs

Number of steps:	550 maximum
Jump function:	Unconditional jump (Goto), 10 maximum Conditional jump ($=, \neq, >, <, \geq, \leq$) Count jump (Isz, Dsz)
Subroutines:	9 levels
Number of stored programs:	10 maximum (P0 to P9)
Check function:	Program checking, debugging, deletion, addition, etc.

Common section

Display system and contents:	Liquid crystal display (12 digits), 10-digit mantissa and 2-digit exponent, binary, octal, hexadecimal display, sexagesimal display, condition displays (WRT, PCL, Disp, LR, SD, Base-n, D, R, G, S, M, A, hyp, Dec, Hex, Bin, Oct, Fix, Sci)
Error check function:	Checks for values exceeding 10^{100}, illogical computations and illogical jumps; error messages displayed.
Power supply:	Two lithium batteries (CR2032)
Power consumption:	0.01W
Battery life:	Approximately 450 hours on type CR2032.
Auto power off:	Power is automatically switched off approximately 6 minutes after last operation.
Ambient temperature range:	0℃–40℃ (32°F–104°F)
Dimensions:	9.3mmH × 71.5mmW × 132.5mmD ($\frac{3}{8}$"H × 2$\frac{3}{4}$"W × 5$\frac{3}{16}$"D)
Weight:	87 g (3.1 oz) including batteries

fx-4000P
OPERATION GUIDE

(英) SH1001811A Printed in Japan

Manual computations

* Mode specification

Comp mode $\boxed{\text{MODE}}\ \boxed{+}$	Four arithmetic computations and function computations.
Base-n mode $\boxed{\text{MODE}}\ \boxed{-}$	Binary, octal, decimal, hexadecimal conversions and computations, logical operations.
SD mode $\boxed{\text{MODE}}\ \boxed{\times}$	Standard deviation computations (1-variable statistical computations).
LR mode $\boxed{\text{MODE}}\ \boxed{\div}$	Regression computations (paired variable statistical computations).

* Binary, octal, decimal, hexadecimal computations

Setting number system	Decimal .... $\boxed{\text{Dec}}\ \boxed{\text{EXE}}\ (\boxed{\text{Dec}} = \boxed{\sqrt{\ }})$ Hexadecimal ..... $\boxed{\text{Hex}}\ \boxed{\text{EXE}}\ (\boxed{\text{Hex}} = \boxed{x^2})$ Binary .... $\boxed{\text{Bin}}\ \boxed{\text{EXE}}\ (\boxed{\text{Bin}} = \boxed{\log})$ Octal .... $\boxed{\text{Oct}}\ \boxed{\text{EXE}}\ (\boxed{\text{Oct}} = \boxed{\ln})$
Number system specification	Decimal .... $\boxed{\text{SHIFT}}\ \boxed{\text{d}}\ (\boxed{\text{d}} = \boxed{\sqrt{\ }})$ Hexadecimal ..... $\boxed{\text{SHIFT}}\ \boxed{\text{h}}\ (\boxed{\text{h}} = \boxed{x^2})$ Binary .... $\boxed{\text{SHIFT}}\ \boxed{\text{b}}\ (\boxed{\text{b}} = \boxed{\log})$ Octal .... $\boxed{\text{SHIFT}}\ \boxed{\text{o}}\ (\boxed{\text{o}} = \boxed{\ln})$
Logical operations	Not — Reverse of each bit and — Logical product of each bit or — Logical sum of each bit

* Standard deviation computations

Data clear .... $\boxed{\text{SHIFT}}\ \boxed{\text{Scl}}\ \boxed{\text{EXE}}\ (\boxed{\text{Scl}} = \boxed{\text{AC}})$

Data input x [; Frequency) .... $\boxed{\text{DT}}\ (\boxed{\text{DT}} = \boxed{\sqrt[x]{\ }})$

Data deletion x [; Frequency) .... $\boxed{\text{Cl}}\ (\boxed{\text{Cl}} = \boxed{x^y})$

Result display

Number of data .... $\boxed{\text{ALPHA}}\ \boxed{n}\ \boxed{\text{EXE}}\ (\boxed{n} = \boxed{3})$

Sum .... $\boxed{\text{ALPHA}}\ \boxed{\Sigma x}\ \boxed{\text{EXE}}\ (\boxed{\Sigma x} = \boxed{2})$

Sum of squares .... $\boxed{\text{ALPHA}}\ \boxed{\Sigma x^2}\ \boxed{\text{EXE}}\ (\boxed{\Sigma x^2} = \boxed{1})$

Mean .... $\boxed{\text{SHIFT}}\ \boxed{\bar{x}}\ \boxed{\text{EXE}}\ (\boxed{\bar{x}} = \boxed{1})$

Population standard deviation ... $\boxed{\text{SHIFT}}\ \boxed{x\sigma_n}\ \boxed{\text{EXE}}\ (\boxed{x\sigma_n} = \boxed{2})$

Sample standard deviation ..... $\boxed{\text{SHIFT}}\ \boxed{x\sigma_{n-1}}\ \boxed{\text{EXE}}\ (\boxed{x\sigma_{n-1}} = \boxed{3})$

* Regression computations

Data clear .... $\boxed{\text{SHIFT}}\ \boxed{\text{Scl}}\ \boxed{\text{EXE}}\ (\boxed{\text{Scl}} = \boxed{\text{AC}})$

Data input x, y [; Frequency) .... $\boxed{\text{DT}}\ (\boxed{\text{DT}} = \boxed{\sqrt[x]{\ }})$

Data deletion x, y [; Frequency) .... $\boxed{\text{Cl}}\ (\boxed{\text{Cl}} = \boxed{x^y})$

Result display

Number of data .... $\boxed{\text{ALPHA}}\ \boxed{n}\ \boxed{\text{EXE}}\ (\boxed{n} = \boxed{3})$

Sum of x .... $\boxed{\text{ALPHA}}\ \boxed{\Sigma x}\ \boxed{\text{EXE}}\ (\boxed{\Sigma x} = \boxed{2})$

Sum of y .... $\boxed{\text{ALPHA}}\ \boxed{\Sigma y}\ \boxed{\text{EXE}}\ (\boxed{\Sigma y} = \boxed{5})$

Sum of squares of x .... $\boxed{\text{ALPHA}}\ \boxed{\Sigma x^2}\ \boxed{\text{EXE}}\ (\boxed{\Sigma x^2} = \boxed{1})$

Sum of squares of y .... $\boxed{\text{ALPHA}}\ \boxed{\Sigma y^2}\ \boxed{\text{EXE}}\ (\boxed{\Sigma y^2} = \boxed{4})$

Sum of products of x and y ...... $\boxed{\text{ALPHA}}\ \boxed{\Sigma xy}\ \boxed{\text{EXE}}\ (\boxed{\Sigma xy} = \boxed{6})$

Mean of x .... $\boxed{\text{SHIFT}}\ \boxed{\bar{x}}\ \boxed{\text{EXE}}\ (\boxed{\bar{x}} = \boxed{1})$

Mean of y .... $\boxed{\text{SHIFT}}\ \boxed{\bar{y}}\ \boxed{\text{EXE}}\ (\boxed{\bar{y}} = \boxed{4})$

Population standard deviation of x

.... $\boxed{\text{SHIFT}}\ \boxed{x\sigma_n}\ \boxed{\text{EXE}}\ (\boxed{x\sigma_n} = \boxed{2})$

Population standard deviation of y

.... $\boxed{\text{SHIFT}}\ \boxed{y\sigma_n}\ \boxed{\text{EXE}}\ (\boxed{y\sigma_n} = \boxed{5})$

Sample standard deviation of x .. $\boxed{\text{SHIFT}}\ \boxed{x\sigma_{n-1}}\ \boxed{\text{EXE}}\ (\boxed{x\sigma_{n-1}} = \boxed{3})$

Sample standard deviation of y .. $\boxed{\text{SHIFT}}\ \boxed{y\sigma_{n-1}}\ \boxed{\text{EXE}}\ (\boxed{y\sigma_{n-1}} = \boxed{6})$

Constant term of regression formula

.... $\boxed{\text{SHIFT}}\ \boxed{\text{A}}\ \boxed{\text{EXE}}\ (\boxed{\text{A}} = \boxed{7})$

Regression coefficient .... $\boxed{\text{SHIFT}}\ \boxed{\text{B}}\ \boxed{\text{EXE}}\ (\boxed{\text{B}} = \boxed{8})$

Correlation coefficient .... $\boxed{\text{SHIFT}}\ \boxed{\text{r}}\ \boxed{\text{EXE}}\ (\boxed{\text{r}} = \boxed{9})$

Estimated value of x .... y $\boxed{\text{SHIFT}}\ \boxed{\hat{x}}\ \boxed{\text{EXE}}\ (\boxed{\hat{x}} = \boxed{\times})$

Estimated value of y .... x $\boxed{\text{SHIFT}}\ \boxed{\hat{y}}\ \boxed{\text{EXE}}\ (\boxed{\hat{y}} = \boxed{\div})$

Program computations

* Program input

(1) Input mode: WRT mode ([MODE] [2])

(2) Computation mode: Mode that conforms with program specified by [MODE] [+], [MODE] [−], [MODE] [×] or [MODE] [÷].

(3) Program area specification: Cursor is moved to the desired program area number (P0 through P9) using [⇦] and [⇨], and [EXE] is pressed.

* Program execution

(1) Execution mode: RUN mode ([MODE] [1])

(2) Program area specification: Execution starts with [Prg] program area No. [EXE].

* Program erase

(1) Erase mode: PCL mode ([MODE] [3])

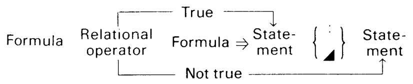
(2) Erasing the program in a single program area
Cursor is moved to the desired program area number (P0 through P9) using [⇦] and [⇨], and [AC] is pressed.

(3) Erasing the programs in all program areas
Press [SHIFT] [Mcl].

* Program commands

Unconditional jump: Goto n → Lbl n (n=0 — 9)

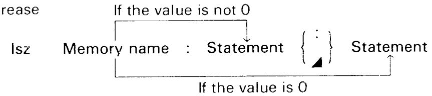
Conditional jump:



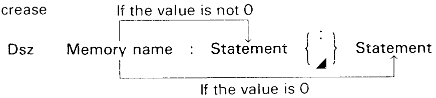
*The relational operator is:
=, ≠, >, <, ≥, or ≤.

Count jump: The value in a memory is increased or decreased.

Increase



Decrease



Error message list

Message	Meaning
Syn ERROR	● Computation formula contains an error. ● Formula in a program contains an error.
Ma ERROR	● Computation result exceeds computation range. ● Computation is performed outside the input range of a function. ● Illogical operation (division by zero, etc.)
Go ERROR	● No corresponding Lbl n to Goto n. ● No program stored in program area Pn which corresponds to Prg n.
Ne ERROR	● Nesting of subroutines by Prg exceeds 9 levels.
Stk ERROR	● Execution of computations that exceed the capacity of the stack for numeric values or stack for computations
Mem ERROR	● Attempt to use a memory such as Z [5] when no memory has been expanded.