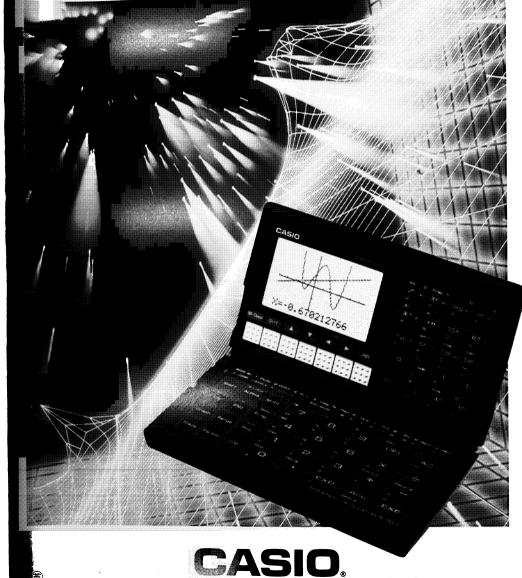
fx-7500G

OWNER'S MANUAL

fx-7500G

OWNER'S MANUAL

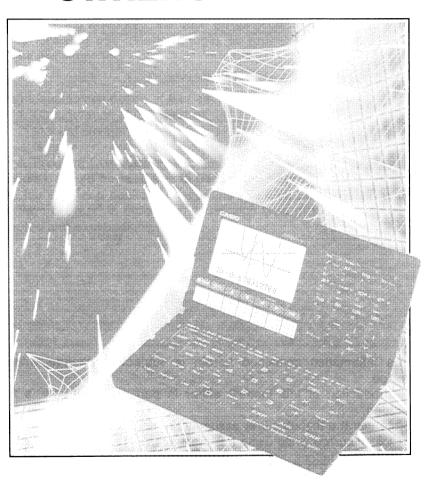


CASIO.

A @ Duinked in 1

CASIO

fx-7500G OWNER'S MANUAL



CASIO.

- The information contained herein is subject to change without notice.
- Reproduction of this manual either in part or its entirety is forbidden.
- Note that the manufacturer assumes no responsibility for any injury or loss incurred while using this manual.
- Due to limitations imposed by printing processes, the displays shown in this manual are only approximations and may differ somewhat from actual displays.

FOREWORD

Thank you for your purchase of the CASIO fx-7500G.

This unit is a totally new type of advanced programmable computer. Graph functions make it possible to produce a wide variety of useful graphs.

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 four sections:

- 1. Configuration and Operation
- 2. Manual Computations
- 3. Graphs
- 4. Program Computations

Section 1 should be read first to become familiar with the nomenclature, handling and cautions concerning this unit. Sections 2, 3 and 4 can then be read in order to master each type of functions through samples and explanations.

CONTENTS

FOREWORD	i
HANDLING PRECAUTIONS	vi
1. CONFIGURATION AND OPERATION	1
1-2 TOWER AND BATTERY RELEASE	3 3 7 7 8 8
1-3 BEFORE BEGINNING COMPUTATIONS Computation priority sequence Number of stacks Computation modes Number of input/output digits and computation digits Overflow and errors Number of input characters Graphic and text displays Display registers Corrections Memory Memory expansion Answer (Ans) function	15 16 17 18 19 20 21 22 23 24 24
2-1 BASIC COMPUTATIONS Arithmetic operations Parenthesis computations Memory computations Specifying the number of decimal places, the number of significant digits and the exponent display	2: 2: 3:

0.0	SPECIAL FUNCTIONS	33
2-2	Continuous computation function	3
	Replay function	34
	Multistatement function	36
	FUNCTIONAL COMPUTATIONS	
2-3	FUNCTIONAL COMPUTATIONS	2
	Angular measurement units	3
	Trigonometric functions and inverse trigonometric	20
	functions	39
	Logarithmic and exponential functions	4(
	Hyperbolic functions and inverse hyperbolic functions	
	Coordinate transformation	
	Other functions	4,
2-4	BINARY, OCTAL, DECIMAL, HEXADECIMAL	
	COMPUTATIONS	
	Binary, octal, decimal, hexadecimal conversions	
	Negative expressions	4
	Basic arithmetic operations using binary, octal, decimal	
	and hexadecimal values	4
	Logical operations	4
2-5	STATISTICAL COMPUTATIONS	
	Standard deviation	4
	Regression computation	5
	Linear regression	5
	Logarithmic regression	
	Exponential regression	
	Power regression	5
2 (2)	RAPHS	5
3-1	BUILT-IN FUNCTION GRAPHS	5
	Overwriting built-in function graphs	
3-2	USER GENERATED GRAPHS	5
	Ranges	5
	User generated function graphs	6
	Function graph overwrite	
	Trace function	6
	Plot function	6
	Line function	7
	Factor function	7
	Instant factor function	7

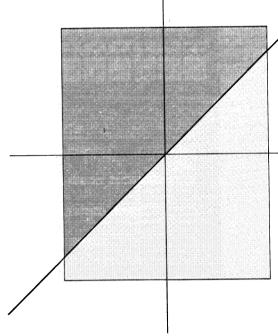
	3-3	GRAPH FUNCTION APPLICATIONS	79
	3-4	SINGLE VARIABLE STATISTICAL GRAPHS	82
		Drawing single variable statistical graphs	
		Summary	85
	3-5	PAIRED VARIABLE STATISTICAL GRAPHS	86
		Drawing paired variable statistical graphs	86
4	. PR	OGRAM COMPUTATIONS	89
	4-1	WHAT IS A PROGRAM?	90
		Formulas	90
		Programming	90
		Program storage	91
		Program execution	93
	4-2	PROGRAM CHECKING AND EDITING	
		(CORRECTION, ADDITION, DELETION)	97
		Formulas	
		Programming	97
		Program editing	98
		Program execution	99
		Summary	102
	4-3	PROGRAM DEBUGGING	
		(CORRECTING ERRORS)	103
			103
			103
		Checkpoints for each type of error	104
	4-4	COUNTING THE NUMBER OF STEPS	106
	4-5	PROGRAM AREAS AND COMPUTATION MODES	108
	. •	Program area and computation mode specification in the	!
		WRT mode	108
		Cautions concerning the computation modes	110
	4-6	ERASING PROGRAMS	111
		Erasing a single program	111
		Erasing all programs	112
	4-7		113
	7 1		113
		Unconditional jump	
		Conditional jump	115

Count jump	117
Summary	119
Subroutines	119
Carriage return function	122
4-8 ARRAY-TYPE MEMORIES	124
	124
Cautions when using array-type memories	125
Application of the array-type memories	127
4-9 DISPLAYING ALPHA-NUMERIC CHARACTERS AND	
SYMBOLS	129
Alpha-numeric characters and symbols	129
	132
4-10 Odina 1112 dita 111 otto 11011 111 111	
PROGRAM LIBRARY	135
Prime factor analysis	136
Greatest common measure	138
Definite integrals using Simpson's rule	140
△↔Y transformation	142
Minimum loss matching	144
Cantilever under concentrated load	146
Parabolic movement	148
Normal distribution	150
Circle and points of tangency	152
Rotation of figures	160
Graph variation by parameters	160
Hysteresis loop	170
Regression curve	17
Parade diagram	18
REFERENCE MATERIAL	19 ⁻
Manual computations	19
Program computations	19
Error messages	19
Input range of functions (general principles)	20
SPECIFICATIONS	20

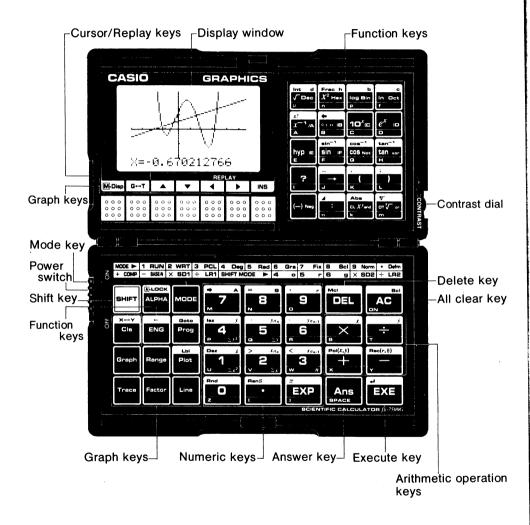
HANDLING PRECAUTIONS

- 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.
- 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.

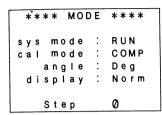
1. CONFIGURATION AND OPERATION

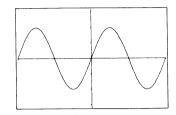


1-1 NOMENCLATURE AND FUNCTIONS



■ Display window





The display window is capable of displaying 16-character by 8-line text and symbols. Graphs are produced on a 95 by 63-dot matrix. A system display as shown on the left indicates the following: the system mode (sys mode), calculation mode (cal mode), angle unit (angle), number of decimal places or number of significant digits (display), and key input buffer status (Step).

The display on the right shows a sine graph as a representative example of the graphs.

The letter "O" is distinguished from zero by adding a slash for the zero (0).

Power switch

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

■ Special operation keys

Shift key

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

Mode Key

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

MODE 1 For manual computations and pro-	ogram execution.			
MODE 2 For writing or checking programs	5.			
MODE 3 For clearing programs.				
MODE 4 Deg displayed. If EXE is presse	ed, unit of angular measure-			
ment is specified as degress.				
MODE 5 Rad displayed. If EXE is presse	d, unit of angular measure-			
ment is specified as radians.				
MODE 6 Gra displayed. If EXE is presse	d, unit of angular measure-			
ment is specified as grads.				
MODE 7 Fix displayed. Entering a value	from 0 to 9 followed by EXE			
will specify the number of deci	mal places according to the			
value entered.				
Ex. MODE 7 3 EXE → Three decim	nal places			
MODE 8 Sci displayed. Entering a value	from 0 to 9 followed by EXE			
will specify the number of signif				
Ex. MODE 8 5 EXE → 5 significant	digits			
MODE 9 Norm displayed. Pressing EXE will cancel the specified num-				
ber of decimal places or the specified number of significant				
digits.				
MODE ☐ Defm displayed. Entering a value	e followed by EXE will specify			
the number of memories availab	ole.			
Ex. MODE · 1 0 EXE → Numbe	r of memories available in-			
creased by 10.				
If EXE is pressed without enterion	ng a value, the current num-			
ber of memories available and	remaining steps will be dis-			
played. (See page 24.)				
Ex. MODE · EXE	**Defm**			
	Program: 195			
	Memory: 56			
	3571 Bytes Free			

MODE

■ ... Specifies COMP mode for arithmetic computation or function computation (program execution possible).

MODE

■ ... For binary, octal or hexadecimal computations/conversions.

MODE × For standard deviation computations (SD1 mode).
MODE : For regression computations (LR1 mode).
MODE 🗵 For production of a bar graph, line graph or normal dis-
tribution curve according to single variable statistical
data (SD2 mode).
SHIFT MODE : For production of a regression line according to paired
variable statistical data (LR2 mode).
SHIFT MODE 4 Pressed after a numeric value representing degrees is
input.
SHIFT MODE 5 Pressed after a numeric value representing radians is
input.

ALPHA Alphabet key

Press to input alphabetic characters or special characters. Pressing ALPHA displays 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 HIFT followed by ALOWA will lock the unit in this mode and allow consecutive input of alphabetic characters until ALPHA is pressed again.

SHIFT MODE 6 ... Pressed after a numeric value representing grads is



input.

Program/Goto key

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

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

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

△ ▽ ⁴ ▷ Cursor/Replay keys

Press to move the cursor (blinking "_") left, right, up, and down on the display. The decursor to the left, be moves the cursor to the right, decursor up, and decursor up, and decursor down. Holding any of the keys down will cause the cursor to continuously move in the respective direction.

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

Pressing the cursor key following shell changes their functions to those marked above the keys.

Insert key

Press to specify insert mode for insertion at the current position of the cursor.

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 DEL EXE will clear the memory contents.

AC All clear key

Press to completely clear the displayed formulas, numeric values or texts, and to clear all of the input buffer contents. Also used to release errors indicated by error message displays, and to restore power after reactivation of the auto power off function. (See page 14.) Pressing AC during graph creation or program execution suspends the operation. A suspended operation can be resumed by pressing EXE.

Execute key

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

Ans Answer key

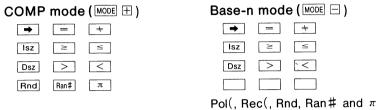
Pressing Ans followed by EXE will recall the last computation result. It can be recalled by Ans EXE even after it has been cleared using the AC key or by switching the power of the unit OFF. When used during program execution, the last result computed is recalled.

\blacksquare $\stackrel{\bullet}{\blacksquare}$ $\stackrel{\bullet}{\blacksquare}$ $\stackrel{\bullet}{\blacksquare}$ $\stackrel{\bullet}{\blacksquare}$ $\stackrel{\bullet}{\blacksquare}$ Numeric/Decimal point/Exponent input

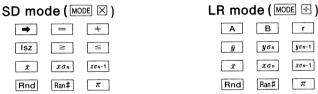
keys

When entering numeric values, enter the number in order. Press the \boxdot key to enter the decimal point in the desired position. To input 1.23×10^{-6} , press $1\boxdot23$ EXP $\boxed{-}$ 6.

SHIFT key combinations for the various modes are as follows:



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



Standard deviation functions can be used

Paired variable statistic functions can be used.

■ Computation 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

Pol(Rec(+ and - keys) ... Coordinate transformation

LR mode

 \widehat{x} \widehat{y} \widehat{X} $\widehat{\dot{+}}$ keys) ... Estimated value computation of x and y

■ Graph keys

Used to produce a variety of graphs (see page 55 for details). These keys cannot be used in the Base-n mode.

Mode display key

 Used to confirm the status of the system mode, calculation mode, angle unit and rounding. Setting status is displayed only while this key is pressed.

G⊶T Graph-text key

• Switches between the graph display and text display (see page 20).

Plot key/Label key

• Press to plot a point on the graph screen.

• Pressed following SHIFT to input label within programs.

Graph key

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

Trace key

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

Range key

• Used to confirm or set the range and size of graphs.

Factor key

Press to magnify or reduce the upper and lower ranges of graphs.

[∟] Line key

• Press to produce line graphs or regression lines.

Clear screen/x-y coordinate key

- Cis EXE clears the graph display. The text display cannot be cleared using this operation.
- ullet Pressed following shift to switch the x and y coordinate display during graph trace operations.

■ 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 36.)

• 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.

Engineering key

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

$$(10^3 = K, 10^6 = M, 10^9 = G, 10^{-3} = M, 10^{-6} = \mu, 10^{-9} = n, 10^{-12} = p)$$

Root/Integer key

 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.

Square/Fraction key

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.

Common logarithm key

Press prior to entering a value to obtain the common logarithm of that value.

● 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.

րա Natural logarithm key

Press prior to entering a value to obtain the natural logarithm of that value.

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.

Reciprocal/Factorial key

- 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 (10₁₀) of a hexadecimal value.

Degree/minute/second key (decimal↔sexagesimal key)

- 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 (11₁₀) of a hexadecimal value.

Anti-common logarithm key

- When pressed, the subsequently entered value becomes an exponent of 10.
- Press in the Base-n mode to enter C (12₁₀) of a hexadecimal value.

Anti-natural logarithm key

- When pressed, the subsequently entered value becomes an exponent of e.
- Press in the Base-n mode to enter D (13₁₀) of a hexadecimal value.

Hyperbolic key

- Pressing hyp, and then sin, cos, or tan prior to entering a value produces the respective hyperbolic function (sinh, cosh, tanh) for the value.
- Pressing SHIFT, then hyp and then sin, cos, or tan prior to entering a value produces the respective inverse hyperbolic function (sinh¹, cosh¹, tanh¹) for the value.
- Press in the Base-n mode to enter E (14₁₀) of a hexadecimal value.

Trigonometric function/Inverse trigonometric function keys

- Press one of these keys prior to entering a value to obtain the respective trigonometric function for the value.
- Press shift and then one of these keys prior to entering a value to obtain the respective inverse trigonometric function for the value.
- Press E in the Base-n mode to enter F (15₁₀) of a hexadecimal value.

- When obtaining logical negation for a value in the Base-n mode, press Not prior to entering the value.
- Press the xor key in the Base-n mode to obtain the exclusive logical sum.

Value input key

 During execution of program computations or cosecutive computations, press to enter a numeric value.

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: 1 2 ± 4 5 → ALPHA

 A EXE
- When pressed following the SHIFT key, the same numeric value can be assigned to multiple memories.
 - Ex. To assign the value 456 to memories A through F: 4 5 6 -

Parenthesis keys

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

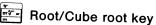
⊶ Minus key

- Press prior to entering a numeric value to make that value negative.
 Ex. -123→ (-) 1 2 3
- Press in the Base-n mode prior to entering a value to obtain the negative of that value. The negative number is the two's complement of the value entered.

Power/Absolute value key

the SHIFT key.

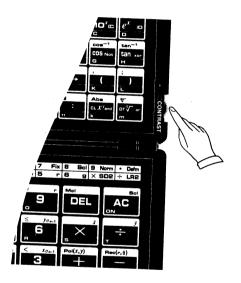
- Enter x (any number), press this key and then enter y (any number) to compute x to the power of y.
 In the SD or LR mode, this function is only available after pressing
- Press following the SHIFT key to obtain the absolute value of a subsequently entered numeric value.
- Press in the Base-n mode to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



- 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 sum key.
- Press following the SHIFT key to obtain the cube root of a subsequently entered numeric value.
- Press in the Base-n mode to obtain a logical sum ("or").
- Used as a data input key in the SD or LR mode.

Adjusting the display contrast

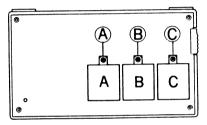
Rotating the contrast dial upwards makes the characters on the display lighter, while rotating it downwards makes the characters darker.

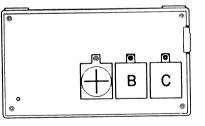


1-2 POWER AND BATTERY REPLACEMENT

This unit is powered by three lithium batteries (CR2025). Low battery power is indicated by a dim display, when the contrast is set to its strongest setting. Replace batteries as soon as possible after weakened batteries are noticed. Notice that the batteries of this unit are used for both normal operation and memory back up. Be sure to follow the sequence described below to avoid the loss of data while replacing batteries. Also, batteries should be replaced at least once every two years no matter how much they have been used. Also note that removing more than one battery from the unit at the same time can cause data stored in memory to be changed. Remove batteries one at a time only.

- 1. Switch the power of the unit OFF and remove the four screws holding the back cover in place.
- 2. Remove screw (A) and battery cover A. Then remove the old battery.
- 3. Wipe off a new battery with a dry cloth and load it into the unit with the possitive pole (⊕) facing upwards.
- 4. Press down on battery with the battery cover A and replace screw (A).
- 5. Repeat steps 1. through 4. to replace baterries B and C.
- Replace the back cover of the unit and fasten it in place using the four screws.
- 7. Switch the power of the unit ON and adjust the contrast using the procedures described on page 4.





IMPORTANT

- Never allow batteries to come into contact with direct heat. Doing so can cause them to explode.
- Be sure to load batteries with positive poles facing upwards.
- Keep batteries out of the reach of small children. Contact a physician immediately if swallowed.

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 AC key. (Numeric values in the memories, programs or computation modes are unaffected when power is switched off.)

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!, °, r, g, °'"
- 3. Power/root x^y , $\sqrt[x]{}$
- 4. Abbreviated multiplication format in front of π or memory 2π , 4R, 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^{-1} , \cosh^{-1} , \tanh^{-1} , (—), Abs, Int. Frac. h. d. b. o. Neg, Not
- 6. Abbreviated multiplication format in front of Type B functions or parenthesis 3sin5, $6\sqrt{7}$, 2sin30cos60, etc.
- 7. X, ÷
- 8. +. -
- 9. and
- 10. or, xor
- 11. Relational operators $\langle , \rangle, =, \pm, \leq, \geq$
- * 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., $e^x \ln \sqrt{120} \rightarrow e^x \left\{ \ln \sqrt{120} \right\}$. Otherwise, execution is from left to right.
- * Compound functions are executed from right to left:

e.g., $\sin \cos^{-1}0.6 \rightarrow \sin (\cos^{-1}0.6)$.

* Everything contained within parentheses receives highest priority.

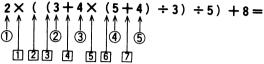
Ex.
$$2+3 \times (\log \sin 2\pi^2_{rad} + 6.8) = 22.07101691$$



■ 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



value stack		
1	2	
2	3	
3	4	

5

Numeric

stack		
1	×	
2	(
3	(
4	+	
5	×	
6	(
7	+	
:		

Command

■ 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
 - Graph production as well as manual computations and program executions.
- 2. WRT mode
 - Program storage and editing. (See Section 4.)
- PCL mode Deletion of stored programs. (See Section 4.)

Computation modes

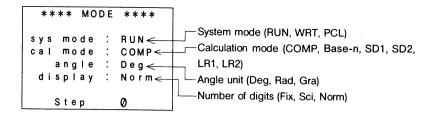
There are a total of six computation modes which are employed according to the type of computation.

- 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 44.) Function computations and graph drawing cannot be performed.
- 3. SD1 mode
 - Standard deviation computation (single variable statistics). (See page 48.)
- 4. SD2 mode
 - For production of bar graph, line graph or normal distribution curve according to single variable statistical data. (See page 83.)
- 5. LR1 mode
 - Regression computation (paired variable statistics). (See page 50.)
- 6. LR2 mode
 - For production of regression line graph according to paired variable statistical data. (See page 86.)

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

* IMPORTANT: When the power of the unit is switched OFF (including auto power off), the current system mode is cancelled, and the unit will be set to the RUN mode when switched ON again. However, the calculation mode, number of decimal place setting (MODE 7 n), number of significant digits (MODE 8 n), and angle unit (Deg, Rad, Gra) will be retained in memory. The mode setting is displayed when the power of the unit is switched ON. Confirm whether the desired mode is set before performing calculations.

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



■ 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 13 digits for a mantissa and 2 digits for an exponent.

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

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

42857.14286
0.14285714

* Computation results greater than 10¹⁰ (10 billion) or less than 10⁻² (0.01) are automatically displayed in exponential form.

Ex. $123456789 \times 9638 =$

123456789 ⊠ 9638 EXE

1.189876532 _E +12

↑

Mantissa Exponent

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. 3×105÷7=

3 EXP 5 → 7 EXE — 42857 EXE

42857.14286
 0.14286

^{*} Values are stored in memory with 13 digits for the mantissa and 2 digits for the exponent.

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 $\pm 9.99999999\times 10^{99}$.
- (2) An attempt is made to perform functional computations that exceed the input range. (See page 201.)
- (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 \square 's followed by $\square \oplus \square \boxtimes \square$
- (5) Even though memory has not been expanded, a memory name such as Z [2] is used. (See page 24 for details on memory.)
- (6) Input errors are made. (Ex. 5 ± ± 3 Ex.)
- (7) When improper arguments are used in commands or functions that require arguments. (i.e. Input of an argument outside of the range of 0∼9 for Sci or Fix.)

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

- (I)~(3) Ma ERROR
- (4) Stk ERROR
- (5) Mem ERROR
- (6) Syn ERROR
- (7) Ara ERROR

Besides these, there are an "Ne ERROR" (nesting error) and a "Go ERROR". These errors mainly occur when using programs. See page 103 or the Error Message Table on page 199.

■ Number of input characters

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

One function comprises one step. Each press of numeric or \boxplus , \boxminus , \boxtimes and \boxdot keys comprise one step. Though such operations as $\exists \exists \exists x \in \mathbb{R}$ ($x \in \mathbb{R}$) require two key operations, they actually comprise only one function and, therefore, only one step.

These steps can be confirmed using the cursor. With each press of the or b key the cursor is moved one step.

Input characters are limited to 127-steps. Usually the cursor is represented by a blinking "__", but once the 123rd 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.

■ Graphic and text displays

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

Switching between graph and text displays is performed using the Get key. Each press of Get switches from the current type of display to the other.

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

Graphs: CIS EXE

Text: AC

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

■ Display registers

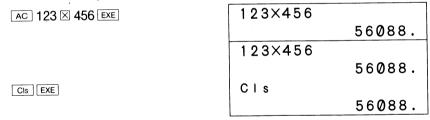
This unit has separate registers for storing text and graph displays. Both of these two registers are unaffected by key operations except for those related to their functions (calculations or AC key operation during text display; graph drawing, switching to text display by G-T after clearing graph display by CIS

Since the register stores the previous calculation results, they can be recalled. This is especially useful in the text mode for binary, octal, decimal, and hexadecimal conversions, as well as decimal and significant digit settings. The following commands will produce previous calculation results:

• Lbl 🔾	• Deg	• Prog 🔾
• Dsz 🔾	Rad	
• Isz 🔘	• Gra	
• Mcl	• Fix	
• Hex	• Sci 🔘	
• Dec	 Norm 	
• Bin	Rnd	
• Oct	• Scl	

Ex. Perform the calculation 123×456, and then clear the graph display.

* The CIS EXE operation during graph display does not affect the calculation, so the previous calculation result appears on the display.



A calculation result displayed as shown here is cleared to 0 by pressing AC, or if the power of the unit is switched OFF (including auto power off).

Co	rr۵	cti	۸r	10
	пe	Cu	OI.	IJ

- To make corrections in a formula that is being input, use the <a> and <a> b< a> keys to move to the position of the error and press the correct keys.
 - Ex. To change an input of 122 to 123:

122

3

sin

122_	
1 2 <u>2</u>	
123_	

Ex. To change an input of cos60 to sin60:

cos	60	
4	4	4

ÇOS	60_	
<u>c</u> o s	60	
sin	<u>6</u> 0	

- * If, after making corrections, input of the formula is complete, the answer can be obtained by pressing EXE. If, however, more is to be added to the formula, advance the cursor using the \(\bigsicap\) key to the end of the formula for input.
- If an unnecessary character has been included in a formula, use the □ and □ keys to move to the position of the error and press the □ key. Each press of □ will delete one command (one step).
- Ex. To correct an input of 369×2:

369XX2

369××2_	
369× <u>2</u>	

- If a character has been omitted from a formula, use the d and b keys to move to the position where the character should have been input, and press the NS key. Press NS and insertions can be subsequently performed as designed.
 - Ex. To correct an input of 2.362 to sin2.362:

 $2 \cdot 3 \cdot 6 \cdot x^{2}$ $4 \cdot 4 \cdot 4 \cdot 4$

INS

sin

2.362_	
<u>2</u> .36 ²	
2.36 ²	
sin 22.36 ²	

* When NS is pressed, the letter at the insertion position is surrounded by "[]" and blinks. As many letters and/or commands as desired can be inserted at this position until 4, D, D, D, or AC is pressed.

This blinking [] is indicated by "__" in the alphabet mode (ALPHA key), while it is indicated by "[]" in the shift mode (SHIFT).

■ Memory

This unit contains 26 standard memories. Memory names are composed of the 26 letters of the alphabet. Numeric values with 13 digits for a mantissa and 2 digits for an exponent can be stored.

Ex. To store 123.45 in memory A:

123.45 → ALPHA A

EXE

EXE

EXE

123.45→A_	
	123.45

Values are assigned to a memory using the — key followed by the memory name.

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

ALPHA A + 78.9 - ALPHA B

A + 78.9→B_	
	202.35

Ex. To add 74.12 to memory B:

ALPHA B + 74.12 → ALPHA B

B+74.12	2→B_		
		276	. 47

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

ALPHA A EXE

123.45

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

Ex. To clear the contents of memory A only:

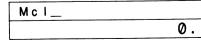
O → ALPHA A EXE

0.

Ex. To clear the contents of all the memories:

SHIFT McI

EXE



- To store the same numeric value to multiple memories, press shift followed by (→ key).
 - Ex. To store a value of 10 in memories A through J:

10 → ALPHA A SHIFT ~ ALPHA J

■ Memory expansion

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

* See page 106 for information on the number of program steps.

Number of memories	26	27	28	 36	 76	 526
Number of steps	4006	3998	3990	 3926	 3606	 6

Memory is expanded in units of one. A maximum of 500 memories can be added for a maximum total of 526 (26 \pm 500). Expansion is performed by pressing $\frac{\text{MODE}}{\text{F}}$, followed by \odot , a value representing the size of the expansion, and then $\frac{\text{EXE}}{\text{EXE}}$.

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

The number of steps used, number of memories and number of remaining steps are displayed. The number of remaining steps indicates the current unused area, and will differ according to the size of the program stored. To check the current number of memories, press $\boxed{\texttt{MODE}}$, followed by $\boxed{\cdot}$ and then $\boxed{\texttt{EXE}}$.

MODE • EXE

Defm

Program : 0

Memory : 56

3766 Bytes Free

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 · O EXE

Defm

Program : Ø

Memory : 26

4006 Bytes Free

- * Though a maximum of 500 memories can be added, if a program has already been stored and the number of remaining steps is less than the desired expansion, an error will be generated. The size of the memory expansion must be equal to or less than the number of steps remaining.
- * The expansion procedure (MODE : expansion value) can also be stored as a program.

Using expanded memories

Expanded memories are used in the same manner as standard memories, and are referred to as Z [1], Z [2], etc. The letter Z followed by a value in brackets indicating the sequential position of the memory is used as the memory name. (Brackets are formed by ALPHA of or "[" 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 124.

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 <code>EXE</code> 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 <code>Ans</code> key.

When $\overline{\mbox{\sc Ans}}$ 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.
$$123+456=\underline{579}$$

 $789-579=\underline{210}$

1 2 3 + 4 5 6 EXE

7 8 9 - Ans EXE

123+456	
	579.
789-Ans	
	210.

Numeric values with 13 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:

78 + 56 EXE

Ans EXE ... Checking the content of ... Ans memory

1 2 3 - ALPHA A EXE

Ans EXE

78+56	
	134.
Ans	
	134.
123→A	
	123.
Ans	
	134.

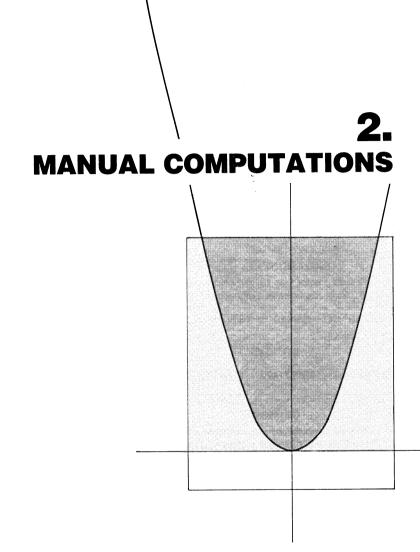
The Ans memory can be used in the same manner as the other memories, thus making it possible to use it in computation formulas. In multiplication operations, the \boxtimes immediately before $\widehat{\mbox{\sc Ans}}$ can be omitted.

Ex.
$$15 \times 3 = \underline{45}$$

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

1 5 × 3 EXE

7 8 Ans – 2 3 EXE



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 (-) before entering the value.

Example	Operation	Display
23+4.5-53=-25.5	23 ⊞ 4.5 ⊡ 53 EXE	-25.5
$56 \times (-12) \div (-2.5) = 268.8$	56 ⋈ (-) 12 ÷ (-) 2.5 EXE	268.8
12369×7532×74103= 6.903680613×10 ¹² (6903680613000) * Results greater than 10 ¹⁰ (0.01) are displayed in exp	10 billion) or less than 10 ⁻²	6.903680613∈+12
$(4.5\times10^{75})\times(-2.3\times10^{-79})$ =-1.035×10 ⁻³ (-0.001035)	4.5 EXP 75 ⋈ (-) 2.3 EXP (-) 79 EXE	−1.035 _∈ −03
(1×10 ⁵)÷7=14285.71429	1 EXP 5 ÷ 7 EXE	14285.71429
(1×10⁵)÷7−14285= 0.7142857	1 EXP 5 ⊕ 7 — 14285 EXE	0.71428571
* Internal computations are mantissa, and the result is digits.	computed in 13 digits for a displayed rounded off to 10	

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

Example	Operation	Display
3+ <u>5×6</u> =33	3 ± 5 ⊠ 6 EXE	33.
$7\times8-4\times5=36$	7 × 8 - 4 × 5 EXE	36.
1+2- <u>3×4÷5</u> +6=6.6	1 ± 2 − 3 × 4 ÷ 5 ± 6 EXE	6.6

■ Parenthesis computations

Example	Operation	Display
100-(2+3)×4=80	100 — (2 ± 3) × 4 EXE	80.
* Closed parentheses occui	2 ⊞ 3 ⊠ [] 4 ⊞ 5 EXE rring immediately before opay be omitted, no matter how	29.
(/-2)/(010)	(7 – 2) (8±5 ExE ccurring immediately before be omitted.	65.
10-\{2+7\times(3+6)\} =-55 * Henceforth, abbreviated s manual.	10 — (2 ± 7 (3 ± 6 EXE tyle will not be used in this	-55.
$\frac{2 \times 3 + 4}{5} = (2 \times 3 + 4) \div 5 = 2$	(2 × 3 ± 4) ÷ 5 EXE	2.
$\frac{5\times6+6\times8}{15\times4+12\times3} = 0.8125$	(() 5 ⋈ 6 ⊞ 6 ⋈ 8 () ⊕ (() 15 ⋈ 4 ⊞ 12 ⋈ 3 ()) EXE	0.8125
$(1.2\times10^{19}) - \{(2.5\times10^{20}) \times \frac{3}{100}\} = 4.5\times10^{18}$	1.2 EXP 19 - (2.5 EXP 20 X 3 ÷ 100) EXE	4.5 £ + 18
$\frac{6}{4\times5} = 0.3$	6 ÷ (4 × 5) EXE	0.3
* The above is the same as	s 6	

Memory computations

● The contents of memories are not erased when power is switched OFF. They are cleared by pressing SHIFT followed by Mol (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 X 7 EXE	69.118
9.874×26=256.724	ALPHA A X 12 EXE	118.488
9.874×29=286.346	ALPHA A × 26 EXE ALPHA A × 29 EXE	256.724 286.346
ory. (Clearing a memory	input numeric values in mem- before input is not required, lue in the memory will be rith the new value.)	
23+9=32	23 ± 9 → ALPHA B EXE	32.
53-6=47	53 - 6 EXE	47.
$-)45\times2=90$	ALPHA B + Ans - ALPHA B	
99÷3=33	EXE	79.
Total 22	45 ⊠ 2 EXE	90.
	ALPHA B - Ans - ALPHA B	
	EXE	-11.
	99	33.
	ALPHA B + Ans - ALPHA B	
	EXE	22.
12×(2.3+3.4)-5=63.4	2.3 ± 3.4 → ALPHA G EXE	5.7
	12 X ALPHA G - 5 EXE	63.4
$30 \times (\underline{2.3 + 3.4 + 4.5}) - 15$	4.5 → ALPHA H EXE	4.5
\times 4.5=238.5	30 🗵 (ALPHA G 🕂 ALPHA H)	
	15 ALPHA H EXE	238.5
 Multiplication signs (X) in names can be omitted. 	mmediately before memory	

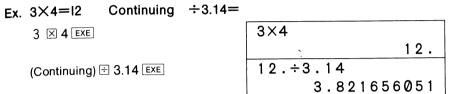
- Specifying the number of decimal places, the number of significant digits and the exponent display
- To specify the number of decimal places, press MODE followed by ⑦, a value indicating the number of places (0—9) and then EXE.
- To specify the number of significant digits, press MODE followed by ⑧, a value indicating the number of significant digits (0 9 to set from 1 to 10 digits) and then EXE.
- Pressing the ENG key or SHIFT followed by (ENG 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 ③ is specified using the sequence: MODE, ⑤, EXE. (Specified values are not cancelled even if power is switched OFF or an other mode (besides MODE) is specified.)
- Even if the number of decimal places and number of significant digits are specified, internal computations are performed in 13 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 (② 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.

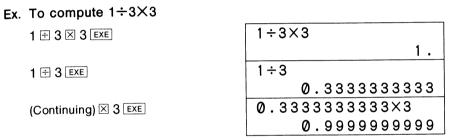
Example	Operation	Display
100÷6=16.6666666	100 ÷ 6 EXE	16.6666667
	MODE 7 4 EXE (Four dec-	16.6667
	imal places specified.)	
	MODE 9 EXE (Specification	16.6666667
•	cancelled.)	
	MODE 8 5 EXE (Five sig-	1.6667 _€ + 0 1
	nificant digits specified.)	
	MODE 9 EXE (Specification	16.6666667
	cancelled.)	
 Values are displayed roun specified. 	ded off to the place	
200÷7×14=400	MODE 7 3 EXE (Three dec-	16.667
	imal places specified.)	
	200 ⊕ 7 EXE	28.571
(Continues computation with	\boxtimes	28.57142857×_
10-digit display.)	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
	\boxtimes	28.571×_
	14 EXE	399.994
	MODE 9 EXE (Specification	399.994
	cancelled.)	
123m×456=56088m	123 ⊠ 456 EXE	56088.
=56.088km	ENG	56.088 _E +03
78g×0.96=74.88g	78 ⊠ 0.96 EXE	74.88
=0.07488kg	SHIFT 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.





This function can be used with memory and Type A functions $(x^2, x^{-1}, x!)$: see page 44), and x^2 , x^3 , x^4 , x^4 , x^5 , x^6 , x^8 ,

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

	•	
12 ⊠ 45 EXE	12×45	
		540.
(Continuing) → ALPHA C EXE	540.→C	
		540.

Ex. To square the result of $78 \div 6$ (see page 42):

78	78÷6	
		13.
(Continuing) x^2 EXE	13.2	
		169.

■ Replay function

● This function stores formulas that have been executed. After execution is complete pressing either the □ or □ key will display the formula executed.

Pressing will display the formula, with the cursor located under the first character.

Pressing d will display the formula, with the cursor located at the space following the last character.

Then using \triangleright , \triangleleft , \triangle and ∇ to move the cursor, the formula can be checked and numeric values or commands can be changed for subsequent execution.

Ex.

123 \times 456 EXE 123 \times 456

56088.

123 \times 456

* The formula appears after clearing the display.

EXE 123 \times 456

56088.

123×456_

Ex. $4.12 \times 3.58 + 6.4 = 21.1496$ $4.12 \times 3.58 - 7.1 = 7.6496$

> 4.12×3.58+6.4_

4.12×3.58±6.4

☐ 7.1 EXE 4.12×3.58−7.1 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 $14 \div 0 \times 2.3$ is mistakenly entered for $14 \div 10 \times 2.3$:

14 ⊕ 0 ⊠ 2.3 EXE	14÷0×2.3
	Ma ERROR
	Step 4
	14÷0×2.3
	Error generated here.
INS 1 EXE	14÷10×2.3
	3.22

- * As with the number of input characters (see page 20), the replay function can accept input up to 127 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.
$$6.9 \times 123 = 848.7$$

 $123 \div 3.2 = 38.4375$

ALPHA A ÷ 3.2 EXE

123
$$\longrightarrow$$
 ALPHA A : 6.9 \boxtimes ALPHA A SHIFT \blacksquare

The display halted by the **⊿** command is represented with −Disp−

EXE

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.
- The unit of angular measurement can be checked by pressing the Mose key.
- You cannot specify the unit of angular measurement (degrees, radians, grads) while the calculator is in the Base-n mode. Such specifications can only be made if you first exit the Base-n mode.

Example	Operation	Display
Conversion of 4.25 rad to degrees	MODE 4 EXE 4.25 SHIFT MODE 5 EXE	243.5070629
Conversion of 1.23 grad to radians	MODE 5 EXE 1.23 SHIFT MODE 6 EXE	0.01932079482
Conversion of 7.89 degrees to grads	MODE 6 EXE 7.89 SHIFT MODE 4 EXE	8.76666667
Result displayed in degrees 47.3°+82.5 rad= 4774.20181	MODE 4 EXE 47.3 ± 82.5 SHIFT MODE 5 EXE	4774.20181
12.4°+8.3 rad-1.8 gra= 486.33497	12.4 ± 8.3 SHIFT MODE 5 = 1.8 SHIFT MODE 6 EXE	486.33497
Result displayed in radians 24°6'31"+85.34 rad= 85.76077464	MODE 5 EXE 24 6 31 SHIFT MODE 4 + 85.34 EXE	85.76077464
Result displayed in grads 36.9°+41.2 rad= 2663.873462	MODE 6 EXE 36.9 SHIFT MODE 4 ± 41.2 SHIFT MODE 5 EXE	2663.873462

^{*} 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

■ 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
sin 63°52'41"= 0.897859012	MODE 4 EXE sin 63 52 41 EXE	0.897859012
$\cos\left(\frac{\pi}{3}\operatorname{rad}\right)=0.5$	MODE 5 EXE COS (SHIFT $\pi \div 3$) EXE	0.5
$\tan (-35 \text{ gra}) = -0.6128007881$	MODE 6 EXE tan (-) 35 EXE	-0.6128007881
$2 \cdot \sin 45^{\circ} \times \cos 65^{\circ} = 0.5976724775$	MODE 4 EXE 2 Sin 45 Cos 65 EXE Can be omitted.	0.5976724775
$\sin^{-1} 0.5=30^{\circ}$ (Determine the value of x when $\sin x=0.5$.)	SHIFT sin 0.5 EXE Can be entered as .5	30.
$\cos^{-1} \frac{\sqrt{2}}{2} = 0.7853981634 \text{ rad}$ $= \frac{\pi}{4} \text{rad}$	MODE 5 EXE SHIFT cos ⁻¹ (\(\tau \) 2 \(\tau \) 2 EXE SHIFT \(\pi \) EXE	0.7853981634 0.25
tan ⁻¹ 0.741= 36.53844577° =36°32'18.4" * If the total number of digits for degrees/minutes/ seconds exceeds eleven digits, the high-order values (degrees and minutes) are given display priority, and any lower-order values are not dis- played. However, the entire value is stored within the unit as a decimal value.		36.53844577 36°32′18.4″
$2.5 \times (\sin^{-1}0.8 - \cos^{-1}0.9)$ = 68°13'13.53"	2.5 × (SHIFT sin ⁻¹ 0.8 - SHIFT cos ⁻¹ 0.9) EXE SHIFT	68°13′13.53″
sin18°×cos0.25rad= 0.2994104044 * The above is computed as sin 18 SHIFT MODE 4	in radians, and is the same	0.2994104044

■ Logarithmic and exponential functions

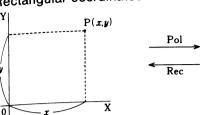
Example	Operation	Display
$\log 1.23(\log_{10}1.23) = 0.08990511144$	log 1.23 EXE	0.08990511144
$ \ln 90(\log e 90) = $ 4.49980967	In 90 EXE	4.49980967
log 456÷In 456= 0.4342944819 (log/In ratio=constant M)	log 456 In 456 EXE	0.4342944819
10 ^{1,23} =16.98243652 (To obtain the antilogarithm of common logarithm 1.23)	10 ^x 1.23 EXE	16.98243652
e^{45} =90.0171313 (To obtain the antilogarithm of natural logarithm 4.5)	e ^z 4.5 EXE	90.0171313
$10^{4} \cdot e^{-4} + 1.2 \cdot 10^{23} = 422.5878667$	10 ^x 4 × e ^x (-) 4 ± 1.2 × 10 ^x 2.3 EXE	422.5878667
5.6 ²³ =52.58143837	5.6 xy 2.3 EXE	52.58143837
$\sqrt[7]{123} (=123^{\frac{1}{7}}) =$ 1.988647795	7 123 EXE	1.988647795
$(78-23)^{-12}$ = 1.305111829×10 ⁻²¹	(78 – 23) xy (-) 12 EXE	1.305111829 _E -21
$2+3\times\sqrt[3]{64}-4=10$ * x' and \(\forall \) given compute:	2 ⊞ 3 ⊠ 3 ₹ 64 ⊡ 4 EXE Itation priority over X and	10.
2×3.4 ^(5+6.7) =3306232.001	2 × 3.4 x (5 ± 6.7)	3306232.001

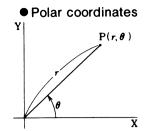
■ Hyperbolic functions and inverse hyperbolic functions

Example	Operation	Display
sinh 3.6=18.28545536	hyp sin 3.6 EXE	18.28545536
cosh 1.23=1.856761057	hyp cos 1.23 EXE	1.856761057
tanh 2.5=0.9866142982	hyp tan 2.5 EXE	0.9866142982
cosh 1.5-sinh 1.5= 0.2231301601 $= e^{-1.5}$	hyp cos 1.5 — hyp sin 1.5 EXE (Continuing) In Ans EXE	0.2231301601 -1.5
(Proof of cosh x $\pm \sinh x = e^{\pm x}$) $\sinh^{-1}30 = 4.094622224$	SHIFT hyp sin-1 30 EXE	4.094622224
$\cosh^{-1}\left(\frac{20}{15}\right) = 0.7953654612$	SHIFT hyp cos ⁻¹ (20 ÷ 15	0.7953654612
Determine the value of x when $tanh 4 x=0.88$		
$x = \frac{\tanh^{-1}0.88}{4} = 0.3439419141$	SHIFT hyp tan-1 0.88 ÷ 4	0.3439419141
sinh ⁻¹ 2×cosh ⁻¹ 1.5= 1.389388923	SHIFT hyp sin ⁻¹ 2 × SHIFT hyp cos ⁻¹ 1.5 EXE	1.389388923
$\sinh^{-1}\left(\frac{2}{3}\right) + \tanh^{-1}\left(\frac{4}{5}\right) =$ 1.723757406	SHIFT hyp sin ⁻¹ (2 ÷ 3) + SHIFT hyp tan ⁻¹ (4 ÷ 5	
1.723737400) EXE	1.723757406

■ Coordinate transformation

Rectangular coordinates





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

Pol
$$\rightarrow$$
I= r , J= θ

$$Rec \rightarrow I = x, J = y$$

• With polar coordinates, θ can be computed within a range of $-180^{\circ} < \theta \le 180^{\circ}$. (The computation range is the same with radians or grads.)

Operation	Display
MODE 4 EXE SHIFT POI(14 SHIFT 20.7)	
(Continuing) ALPHA J EXE	24.98979792(7)
SHIFT O	55°55'42.2"(θ)
MODE 5 EXE SHIFT POI(7.5 SHIFT) (-)	12.5(7)
(Continuing) ALPHA J EXE	-Ø.927295218(8)
MODE 4 EXE SHIFT Rec(25 SHIFT . 56)	
(Continuing) ALPHA J EXE	13.97982259(x) 20.72593931(y)
MODE 5 EXE	
÷ 3 ⋈ SHIFT □ □ □ EXE (Continuing) ALPHA ■ EXE	-2.25(x) 3.897114317(y)
	MODE 4 EXE SHIFT PO(14 SHIFT . 20.7) EXE (Continuing) ALPHA EXE SHIFT PO(7.5 SHIFT . (-) 10) EXE (Continuing) ALPHA EXE MODE 4 EXE SHIFT Rec(25 SHIFT . 56) EXE (Continuing) ALPHA EXE MODE 5 EXE (Continuing) ALPHA EXE MODE 5 EXE SHIFT Rec(4.5 SHIFT . (2

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

Example	Operation	Display
$\sqrt{2} + \sqrt{5} = 3.65028154$	2 ± 5 EXE	3.65028154
$2^2+3^2+4^2+5^2=54$	$2 x^{2} + 3 x^{2} + 4 x^{2}$ $+ 5 x^{2} EXE$	54.
$\frac{1}{\frac{1}{3} - \frac{1}{4}} = 12$	$(3x^{-1}-4x^{-1})x^{-1}$ EXE	12.
8!(=1×2×3×···×8)= 40320	8 SHIFT x! EXE	40320.
$\sqrt[3]{36\times42\times49} = 42$	SHIFT 3√ (36 × 42 × 49) EXE	42.
Random number genera- tion (pseudorandom num- ber from 0.000 to 0.999)	SHIFT Ran# EXE	(Ex) 0.792
$\sqrt{13^2 - 5^2} + \sqrt{3^2 + 4^2} = 17$	13 x ² - 5 x ²) + 7 (3 x ² + 4 x ²) EXE	17.
$\sqrt{1-\sin^2 40^\circ} = 0.7660444431 = \cos 40$	MODE 4 EXE 1 1 - (sin 40) 2 2) EXE	0.7660444431
$(\text{Proof of } \cos \theta = \sqrt{1-\sin^2 \theta})$	(Continuing) SHIFT [cos ⁻¹] Ans	40.
$\frac{1}{2!} + \frac{1}{4!} + \frac{1}{6!} + \frac{1}{8!} = 0.543080357$	~/ \-1 \+ 6 \SHIFT \-r/	0.5430803571
What is the absolute value of the common logarithm of $\frac{3}{4}$? $\left \log \frac{3}{4}\right = 0.1249387366$	SHIFT Abs log (3 ÷ 4)	0.1249387366

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

2-4 BINARY, OCTAL, DECIMAL, HEXADECI-MAL COMPUTATIONS

- Binary, octal, decimal and hexadecimal computations, conversions and logical operations are performed in the Base-n mode (press MODE □).
- The number system (2, 8, 10, 16) is set by respectively pressing Bin, Oct, Dec or Hex, followed by EXE.
- Number systems are specified for specific values by pressing SHFT, then the number system designator (ⓑ, ⊙, ⓓ or ♄), 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.
- Octal, decimal and hexadecimal computations can be handled up to 32 bits, while binary can be handled up to 16 bits.

Binary Up to 16 digits
Octal Up to 11 digits
Decimal Up to 10 digits
Hexadecimal Up to 8 digits

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

Valid values

Binary 0, 1

Octal 0, 1, 2, 3, 4, 5, 6, 7 Decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

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

- Negative numbers in binary, octal and hexadecimal are expressed as two's complements.
- To distinguish the A, B, C, D, E, 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: 111111111111111 $\ge x \ge 0$ Negative: 1111111111111111 $\ge x$

Decimal Positive: $2147483647 \ge x \ge 0$

Negative: $-1 \ge x \ge -2147483648$

Hexadecimal Positive: $7FFFFFFF \ge x \ge 0$

Negative: FFFFFFF $\geq x \geq 80000000$

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
	MODE -	
What are the decimal values for $2A_{16}$ and 274_8 ?	Dec EXE SHIFT h 2A EXE SHIFT o 274 EXE	42. 188.
What are the hexadecimal values for 123 ₁₀ and 1010 ₂ ?	Hex EXE SHIFT d 123 EXE SHIFT b 1010 EXE	0000007B 0000000A
What are the octal values for 15_{16} and 1100_2 ?	Oct EXE SHIFT h 15 EXE SHIFT b 1100 EXE	00000000025 00000000014
What are the binary values for 36 ₁₀ and 3B7 ₁₆ ?	Bin EXE SHIFT d 36 EXE SHIFT h 3B7 EXE	0000000000100100

■ Negative expressions

Example	Operation	Display
	MODE -	
How is 110010 ₂ expressed as a negative?	Bin EXE Neg 110010 EXE	1111111111001110
How is 72 ₈ expressed as a negative?	Oct EXE Neg 72 EXE	3777777706
How is 3A ₁₆ expressed as a negative?	Hex EXE Neg 3A EXE	FFFFFC6

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

Example	Operation	Display
	MODE —	
10111 ₂ +11010 ₂ =110001 ₂	Bin EXE 10111	0000000000110001
B47 ₁₆ -DF ₁₆ =A68 ₁₆	Hex EXE	
	B47 - DF EXE	00000A68
$123_8 \times ABC_{16} = 37AF4_{16}$ = 228084_{10}	SHIFT 0 123 X ABC EXE Dec EXE	00037AF4 228084
$1F2D_{16} - 100_{10} = 7881_{10} \\ = 1EC9_{16}$	SHIFT IN 1F2D - 100 EXE HEX EXE	7881 00001EC9
$7654_8 \div 12_{10}$ = 334.3333333 ₁₀ = 516 ₈ * Computation results are portion cut off.	Dec EXE SHIFT 0 7654 ÷ 12 EXE Oct EXE displayed with the decimal	334 00000000516
1234+1EF ₁₆ ÷24 ₈ =2352 ₈ =1258 ₁₀ * For mixed basic arithme tion and division are give over addition and subtra	en computation priority	00000002352 1258

■ Logical operations

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

Example	Operation	Display
	MODE —	
19 ₁₆ AND 1A ₁₆ =18 ₁₆	Hex EXE 19 and 1A EXE	00000018
1110 ₂ AND 36 ₈ =1110 ₂	Bin EXE 1110 and SHIFT 0 36 EXE	000000000001110
23 ₈ OR 61 ₈ =63 ₈	Oct EXE 23 or 61 EXE	00000000063
120_{16} OR $1101_2 = 12D_{16}$	Hex EXE 120 or SHIFT b 1101 EXE	0000012lD
1010_2 AND (A ₁₆ OR 7 ₁₆)= 1010_2	Bin EXE 1010 and (SHIFT h A or SHIFT h 7) EXE	00000000000001010
5 ₁₆ XOR 3 ₁₆ =6 ₁₆	Hex EXE 5 SHIFT xor 3 EXE	00000006
42 ₁₀ XOR B ₁₆ =33 ₁₀	Dec EXE 42 SHIFT xor SHIFT h B EXE	33
Negation of 1234 ₈	Oct EXE Not 1234 EXE	37777776543
Negation of 2FFFED ₁₆	Hex EXE Not 2FFFED EXE	FFD00012

2-5 STATISTICAL COMPUTATIONS

■ Standard deviation

- Standard deviation computations are performed in the SD1 mode. (Press MODE ⊠.)
- Before beginning computations, the statistical memories are cleared by pressing SHIFT followed by ScI (AC key) and then EXE.
- Individual data is input using DT (▼key).
- Multiple data of the same value can be input either by repeatedly pressing DT or by entering the data, pressing SHIFT, followed by :, that represents the number of times the data is repeated, and then DT.
- Standard deviation

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

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

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

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

Example	Э	Operation	Display
Data 55, 54, 51, 5 54, 52	55, 53, 53,	MODE X SHIFT SCI EXE (Memory clear) 55 DT 54 DT 51 DT 55 DT 53 DT DT 54 DT 52 DT	52.
* Results can b	e obtained	in any order desired.	
11002		$(\text{Standard deviation } \sigma_n) \\ \text{SHIFT} x\sigma_n \text{EXE} \\ (\text{Standard deviation } \sigma_{n-1}) \\ \text{SHIFT} x\sigma_{n-1} \text{EXE} \\ (\text{Mean } \bar{x}) \text{SHIFT} \bar{x} \text{EXE}$	1.316956719
		(Number of data n) ALPHA W EXE (Sum total Σx) ALPHA V EXE (Sum of squares Σx^2)	1
		ALPHA U EXE	22805.
What is deviation unbiased variance ference between datum and the nather above data?	ce, the dif- n each nean of	(Continuing) SHIFT xan-1 x ² EXE 55 — SHIFT x EXE 54 — SHIFT x EXE 51 — SHIFT x EXE :	1.982142857 1.625 0.625 -2.375
What is \bar{x} and x the following tab	ole?	SHIFT SCI EXE 110 SHIFT ; 10 DT 130 SHIFT ; 31 DT	110. 130.
1 110	quoney	150 SHIFT ; 24 DT	150.
2 130		170 DT DT DT 190 DT DT	170.
3 150	24	ALPHA W EXE	70.
4 170	2	SHIFT \$\overline{x}\$ EXE	137.7142857
5 190	3	SHIFT X on-1 EXE	18.42898069

- * Erroneous data clearing/correction I (correct data operation: 51 DT)
- 1) If 50 DT is entered, enter correct data after pressing CL (x key).
- ② If 49 DT was input a number of entries previously, enter correct data after pressing 49 CL.

- * Erroneous data clearing/correction II (correct data operation: 130 SHIFT 31 DT)
- 1) If 120 SHIFT : is entered, enter correct data after pressing AC.
- 2 If 120 SHIFT : 31 is entered, enter correct data after pressing AC.
- 3 If 120 SHIFT : 30 DT is entered, enter correct data after pressing CL.
- (4) If 120 SHIFT : 30 DT was entered previously, enter correct data after pressing 120 SHIFT : 30 CL.

■ Regression computation

- Regression computations are performed in the LR1 mode. (Press MODE ⊕.)
- Before beginning computations, the tabulation memories are cleared by pressing SHIFT followed by ScI and then EXE.
- Individual data are entered as x data SHIFT y data DT.
- Multiple data of the same value can be entered by repeatedly pressing \overline{DT} . This operation can also be performed by entering x data \overline{SHFT} y data \overline{SHFT} f followed by a value representing the number of times the data is repeated, and then \overline{DT} .
- If only x data is repeated (x data having the same value), enter y data y
- If only y data is repeated (y data having the same value), enter x data \Box or x data \Box followed by a value representing the total number of times the data is repeated, and then \Box \Box .
- The regression formula is y = A + Bx, and constant term A and regression coefficient B are computed using the following formulas:

Regression coefficient of regression formula

Constant term of regression formula

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

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

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

$$r = \frac{n \cdot \sum xy - \sum x \cdot \sum y}{\sqrt{\left|n \cdot \sum x^2 - (\sum x)^2\right| \cdot \left|n \cdot \sum y^2 - (\sum y)^2\right|}}$$

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

Linear regression

Exa	ample	Operation	Display
Temperate length of a	ure and the	MODE ÷	
Temp.	Length 1003mm	clear) 10 SHIFT . 1003 DT	10.
15 20 25	1005 1010 1011 1014	15 SHIFT : 1005 DT 20 SHIFT : 1010 DT 25 SHIFT : 1011 DT 30 SHIFT : 1014 DT	15. 20. 25. 30.
Using this ta	able the re-	(Constant term A) SHIFT A EXE	997.4
obtained. Ba	fficient can be ased on the ormula, the	(Regression coefficient B) SHIFT B EXE	0.56
18°C and the	e steel bar at e temperature can be esti-	(Correlation coefficient r) SHIFT T EXE	0.9826073689
	e, the critical (r²) and covar-	(Length at 18°C) 18 SHIFT 1 EXE	1007.48
iance	$\left(\frac{\bar{z}\cdot ar{y}}{z}\right)$ can also	(Temperature at 1000mm) 1000[SHIFT] I EXE	4.642857142
be compute	ed.	(Critical coefficient) SHIFT T x2 EXE	0.9655172414
		(Covariance) (ALPHA R = ALPHA W X SHIFT X SHIFT	
		Ū) → (ALPHA W - 1) EXE	35.

^{*} Erroneous data clearing/correction (correct data operation: 10 SHIFT ... 1003 DT)

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

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

⁽³⁾ If 11 SHIFT I 1003 DT was entered previously, enter correct data after pressing 11 SHIFT I 1003 CL.

● Logarithmic regression

- The regression formula is $y = A + B \cdot \ln x$. Enter the x data as the logarithm (In) 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} , in x shift \hat{y} EXE is used, and to obtain estimated value \hat{x} , y shift \hat{y} EXE e^x Ans EXE is used.

Furthermore, Σx , Σx^2 , and Σxy are obtained as $\Sigma \ln x$, $\Sigma (\ln x)^2$, and $\Sigma \ln xy$ respectively.

Ex	ample	Operation	Display
29 50 74 103 118	y, 1.6 23.5 38.0 46.4 48.9	MODE : SHIFT SCI EXE IN 29 SHIFT 1.6 DT IN 50 SHIFT 23.5 DT IN 74 SHIFT 38.0 DT IN 103 SHIFT 46.4 DT IN 118 SHIFT 48.9 DT	3.36729583 3.912023005 4.304065093 4.634728988 4.770684624
Through logarithmic regression of the above data, the regression formula and correlation coefficient are obtained.		(Constant term A) SHIFT A EXE (Regression coefficient B) SHIFT B EXE (Correlation coefficient r)	-111.1283976 34.0201475
estimated v	e, respective alues \hat{y} and \hat{x} lined for xi =80 using the	SHIFT \vec{r} EXE $(\hat{y} \text{ when } xi = 80)$	0.9940139466 37.94879482
regression f	ormula.	$(\hat{x} \text{ when } yi=73) 73 \text{ SHIFT } \hat{x}$ EXE e' Ans EXE	224.1541313

Exponential regression

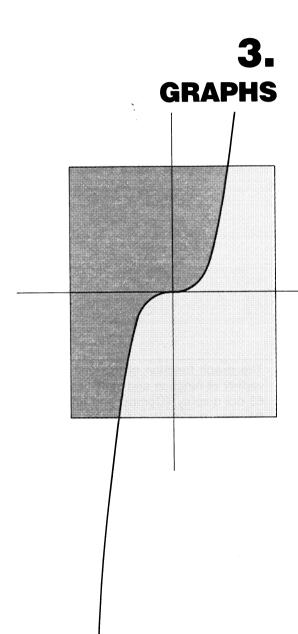
- The regression formula is $y = A \cdot e^{B \cdot x} (\ln y = \ln A + B x)$. Enter the y data as the logarithm of $y(\ln)$, and the x data the same as that for linear regression.
- Correction is performed the same as in linear regression. Constant term A is obtained by $\underbrace{e^x}$ SHIFT A EXE,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

	Example		Operation	Display
	x _i 6.9 12.9 19.8 26.7 35.1	y _i 21.4 15.7 12.1 8.5 5.2	MODE : SHIFT SCI EXE 6.9 SHIFT , In 21.4 DT 12.9 SHIFT , In 15.7 DT 19.8 SHIFT , In 12.1 DT 26.7 SHIFT , In 8.5 DT 35.1 SHIFT , In 5.2 DT	6.9 12.9 19.8 26.7 35.1
gr da mi	Through exponential regression of the above data, the regression formula and correlation coefficient are obtained. Furthermore, the regres-		(Constant term A) (Constant term A) (Regression coefficient B) (SHIFT B EXE	30.49758743
sion formula is used to obtain the respective estimated values \hat{y} and \hat{x} when $xi=16$ and $yi=20$.		espective eses \hat{y} and \hat{x}	ti- $(\hat{y} \text{ when } xi=16) \text{ 16 SHIFT } \hat{y}$	-0.997247352 13.87915739
			(\hat{x} when yi =20)	8.574868046

▶ Power regression

- The regression formula is $y = A \cdot x^B (\ln y = \ln A + B \ln x)$. Enter both data x and y as logarithms (In).
- Correction is performed the same as in linear regression. Constant term A is obtained by $\stackrel{e^{\bot}}{=}$ SHIFT $\stackrel{\frown}{=}$ EXE, estimated value \hat{y} is obtained by $\stackrel{[]}{=}$ In x SHIFT $\stackrel{\frown}{=}$ EXE $\stackrel{e^{\bot}}{=}$ Ans EXE, and estimated value \hat{x} is obtained by $\stackrel{[]}{=}$ In x SHIFT $\stackrel{\frown}{=}$ EXE $\stackrel{e^{\bot}}{=}$ Ans EXE. $\sum x$, $\sum x^2$, $\sum y$, $\sum y^2$ and $\sum xy$ are obtained by $\sum \ln x$, $\sum (\ln x)^2$, $\sum \ln y$, $\sum (\ln y)^2$, and $\sum \ln x \cdot \ln y$ respectively.

	Example	Operation	Display
x _i 28	<i>y</i> , 2410	MODE :- SHIFT Scl EXE	
30 33 35	3033 3895 4491	In 28 SHIFT , In 2410 DT In 30 SHIFT , In 3033 DT	3.33220451
38	5717	in 33 SHIFT , in 3895 DT In 35 SHIFT , in 4491	3.496507561
of the abo	power regression ove data, the re- formula and cor-	In 38 SHIFT , In 5717 DT (Constant term A)	3.555348061
relation contained.	oefficient are ore, the regres-	(Regression coefficient B)	0.2388010724
obtain the	ula is used to respective esti- lues \hat{x} and \hat{y}	(Correlation coefficient r) SHIFT [EXE	2.771866153 0.9989062542
yi=1000.		(\hat{y} when xi =40) In 40 SHIFT \hat{y} EXE	0.000002042
		(\hat{x} When $yi=1000$) In 1000 SHIFT \hat{x} EXE	6587.67458
		cr Ans EXE	20.2622568



The graph function of this unit makes it possible to produce a wide variety of function and statistical graphs quickly and easily on a 95 \times 63 dot display. (Upmost and leftmost lines are not used.)

Besides the built-in function graphs, a generous selection of functions can also be input for graphic representation.

Graph commands can be used manually or in programs, but here all examples will be centered around manual operations. Programmed graphs are identical to those produced manually, and details can be found on page 132.

* Some of keys used for the operation examples in this manual show alphabetic character key markings. On the actual unit, alphabetic characters are marked under the keys by which they are represented.

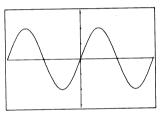
3-1 BUILT-IN FUNCTION GRAPHS

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

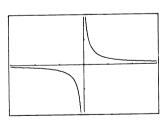
•
$$\sin$$
 • \cos • \tan • \sin^{-1} • \cos^{-1} • \tan^{-1}
• \sinh • \cosh • \tanh • \sinh^{-1} • \cosh^{-1} • \tanh^{-1}
• x^2 • \log • \ln • 10^x • e^x

Any time a built-in graph is executed, the ranges (see page 59) are automatically set to their optimum values, and any graph previously on the display is cleared.

Ex. 1) Sine curve



Ex. 2)
$$y = \frac{1}{x}$$
 graph



■ Overwriting built-in function graphs

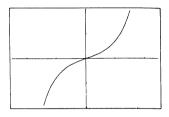
Two or more different built-in function graphs can be written together on the same display. Since the range for the first graph is automatically set, all subsequent graphs on the same display are produced according to the range of the first graph.

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

Subsequent graphs are produced using the variable X in the operation ${\tt Graph}$ [function key] ALPHA ${\tt X}$ [EXE] (${\tt X}$: ${\tt H}$ key). By inputting ALPHA ${\tt X}$ after the function key, the range is unchanged and the next graph is produced without clearing the existing display. (See page 65 for details.)

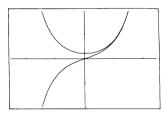
Ex. Overwrite the graph for $y = \cosh x$ on the graph for $y = \sinh x$. First, draw the graph for $y = \sinh x$.

Graph hyp sin EXE



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

Graph hyp cos ALPHA X EXE



⟨Note⟩

Built-in function graphs cannot be used in multistatements (see page 36) and cannot be written into programs.

3-2 USER GENERATED GRAPHS

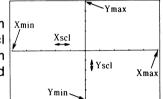
Built-in function graphs can also be used in combination with each other. Graphing a formula such as $y=2x^2+3x-5$ makes it possible to visually represent the solution.

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

Ranges

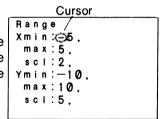
The ranges of the x and y-axes, as well as the scale (distance between points) for both axes can be set or checked using the Range key.

●Ranges contents
Ranges consist of Xmin (x-axis minimum value), Xmax (x-axis maximum value), Xscl (x-axis scale), Ymin (y-axis minimum value), Ymax (y-axis maximum value), and Yscl (y-axis scale).



• Range display

Ranges are displayed as shown on the right when the Range key is pressed. The range value at the cursor position can be changed.



* Values shown here are only an example. Actual values may differ.

Range setting

Range settings are made from the current cursor position and proceed in the order of Xmin→Xmax→Xscl→Ymin→Ymax→Yscl. Input a numeric value at the cursor position and then press EXE. Any value input while the cursor is at the first (extreme left) digit of the displayed value will replace the displayed value when EXE is pressed.

If the \(\bullet \) key is used to move the cursor to the second or subsequent digit of the displayed value, only the portion of the displayed value starting from the cursor position will be affected by the new input when \(\bullet \text{EXE} \) is pressed.

Here, let's try changing the currently set range values to those listed below:

① Input 0 for Xmin.

0 EXE

Range Xmin: 0 max: 55. scl: 2. Ymin: -10. max: 10 scl: 5.

2) The Xmax value is the same, so simply press EXE.

EXE (we can also be used.)

Range

Xmin: 0

max: 5.

scl: 2.

Ymin: -10.

max: 10

scl: 5.

③ Input 1 for Xscl.

1 EXE

Range
Xmin: 0
max: 5.
scl: 1
Ymin: 0.
max: 10
scl: 5.

4 To change Ymin to -5, use the \triangleright key to move the cursor one digit to the right and input 5.

D 5 EXE

Range Xmin: 0 max: 5. scl: 1 Ymin: -5 max: 10. scl: 5. ⑤ To change Ymax to 15, use the D key to move the cursor one digit to the right and input 5.

□ 5 EXE

Range

Xmin: 0

max: 5.

scl: 1

Ymin: -5

max: 15

scl: 5.

6 The YscI value is the same, so simply press EXE.

EXE

Once all settings are complete, the display that was shown before pressing the Regge key is retrieved.

Press the Range key again to confirm whether settings are correct.

Range

Range
Xmin: 0
max: 5
sci:1.
Ymin: -5.
max: 15.
sci:5.

The \triangle and ∇ keys can be used to move the cursor from line to line in the range display without affecting the range values. The cursor can only be moved upwards as far as Xmin, and downwards as far as Yscl. Press to return to the display that was shown before entering the range display.

- * The input range for graph ranges is $-9.9999_{\rm E} + 98$ through $9.99999_{\rm E} + 98$.
- * Only numeric value keys from ① through ⑨, ①, EXP, ①, ④, ♠, ♠, ♠, and Range can be used during range display. Other key operation is ignored.

(Use the (-) key for negative value input.)

* To completely change an existing range setting, ensure that the cursor is located at the first digit (all the way to the left) of the displayed value. If the cursor has been moved to another digit of the value, only the portion of the value from the cursor position (to the right) will be changed. The portion of the value to the left of the cursor will remain unchanged.

Ex.

 \triangleright

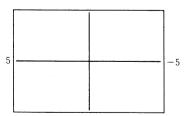
3 EXE **-**25 -35 -3

- * Values up to nine significant digits can be input.

 Values less than 10⁻² and equal to or greater than 10⁸ are displayed with a 6-digit mantissa (including negative sign) and a 2-digit exponent.
- * If input is improper (outside the allowable calculation range or inputting only a negative sign), the existing value will remain unchanged. (The improper input, however, will be temporarily displayed.)
- * Inputting 0 for XscI or YscI does not set any scale.
- * Inputting a maximum value that is less than the minimum value will reverse the respective axis.

Ex. Xmin: 5

Xmax: -5



- * If the maximum and minimum values of an axis are equal, an error (Ma ERROR) will be generated when an attempt is made to produce a graph.
- *When a range setting is used that does not allow display of the axes, the scale for the y-axis is indicated on either the left or right edge of the display, while that for the x-axis is indicated on either the top or bottom edge. (In both cases, the location of the scale is the edge which is closest to the origin (0, 0)).
- * When range values are changed (reset), the graph display is cleared and the newly set axes only are displayed.
- * Range settings may cause irregular scale spacing.
- * If the range is set too wide, the graph produced may not fit on the dis-
- * Points of deflection sometimes exceed the capabilities of the display with graphs that change drastically as they approach the point of deflection.
- * An Ma ERROR may be generated when a range value is specified that exceeds the allowable range.

Ex. Xmin $9.\epsilon99$ Xmax $9.9\epsilon99$ Xscl $1.\epsilon99 \Rightarrow$ Falls outside of range.

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

Range values are reset to their initial values by pressing SHIFT DEL during range display.

Range (Not required when range display is already being shown.)

SHIFT DEL

Range
Xmin:-4.7
max:4.7
sci:1.
Ymin:-3.1
max:3.1
sci:1.

(Reference)

Range settings are performed within programs using the following format:

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

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

■ User generated function graphs

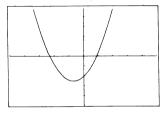
After performing range settings, user generated graphs can be drawn simply by entering the function (formula) after pressing Graph.

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

Set the ranges to the values shown below.

Input the functional formula after pressing the Graph key.

Graph 2 ALPHA
$$\times$$
 x^2 $+$ 3 ALPHA \times $-$ 4 EXE



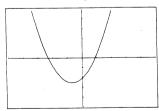
The result produces a visual representation of the formula.

■ Function graph overwrite

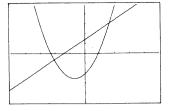
Two or more function graphs can be overwritten which makes it easy to determine intersection points and solutions that satisfy all the equations.

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

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



Next, overwrite the graph for y=2x+3.



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

* Be sure to input variable X (ALPHA \square) into the function when using built-in graphs for overwrite. If variable X is not included in the second formula, the second graph is produced after clearing the first graph.

■ Trace function

The pointer (blinking dot) can be moved using the cursor keys (\bigcirc \triangleright) to determine the x and y coordinates of any point on a graph. After a graph is produced on the display, press \bigcirc and the point will appear at the extreme left plot of the graph. The x-coordinate value (X=. . . .) will appear on the bottom line of the display. The pointer can be moved using the \bigcirc and \bigcirc cursor keys, and the x-coordinate value changes as the pointer moves. To change from the x-coordinate to the y-coordinate value, press \bigcirc \bigcirc \bigcirc The displayed coordinate switches between x and y with each press of \bigcirc \bigcirc

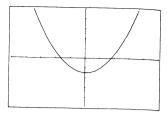
Ex. Determine the points of intersection of the graphs for $y = x^2 - 3$ and y = -x + 2.

The range values should be set as follows:

```
Range
Xmin: -5.
max: 5.
sc!: 1.
Ymin: -10.
max: 10.
sc!: 2.
```

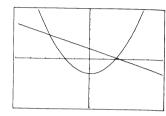
First, draw the graph for $y=x^2-3$.

Graph ALPHA $X x^2 - 3$ EXE



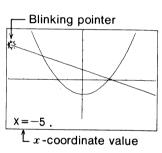
Next, draw the graph for y=-x+2.

Graph (-) ALPHA X + 2 EXE



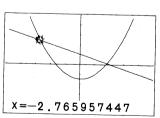
Finally, let's use the trace function.

Trace



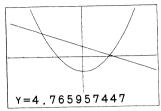
The pointer appears at the extreme left plot of the graph. The key moves the pointer to the right along the graph. Each press of moves the pointer one point, while holding it down causes continuous movement.

~(Hold down)



Hold ightharpoonup down until the pointer reaches the intersection of the two graphs. Note the x-coordinate value, and then press ightharpoonup for the y-coordinate value.

SHIFT X↔Y

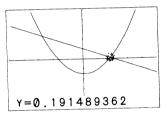


In this way, it can be determined that the coordinates of the first intersection are x=-2.765957447 and y=4.765957447.

* The pointer does not move at the fixed distance because the distance is located along the dots of the display. Therefore, the x-y coordinates for the point of intersection are approximate values.

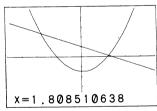
Similarly, press \(\bar{\text{D}} \) to move the pointer to the next point of intersec-

tion.



This time, press $\overline{\text{SHIFT}}$ $\overline{\text{X} \leftarrow \text{Y}}$ to display the x-coordinate value.

SHIFT X++Y



Using the operations outlined above, the approximate x-y coordinates for points along graphs can be obtained.

- * The trace function can only be used immediately after a graph is drawn. This function cannot be used if other calculations or operations (except [MDISS], [Range], or [GIT]) have been employed after a graph has been drawn.
- * The x-y coordinate values at the bottom of the display consist of a 10-digit mantissa or a 5-digit mantissa plus a 2-digit exponent.

- * The trace function cannot be written into a program.
- * The trace function can be used during a "-DISP-" display.
- * When the format Graph formula Graph formula EXE is executed and a graph is drawn by pressing EXE directly after executing the trace function during halt status, the previous coordinate value remains on the display. After the trace function is executed and the text display is brought up using the G-T key, pressing EXE causes the next graph to appear and the coordinate value to clear.

Examine the above using Graph ALPHA X 12 SHIFT Graph 2 ALPHA X + 5

■ Plot function

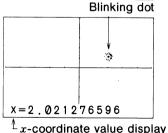
The plot function is used to mark a point on the screen of a graph display. The point can be moved left, right, up and down using the cursor keys, and the coordinates for the graph displayed can be read. Two points can also be connected by a straight line (see Line function, page 71).

Press \underline{Plot} and specify the x and y-coordinates after the "Plot" message.

Ex. Plot a point at x=2 and y=2 on the axes created by the following range values:

Range
Xmin: -5.
max: 5.
scl: 1.
Ymin: -10.
max: 10.
scl: 2.

Plot 2 SHIFT , 2 EXE

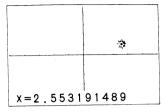


The blinking pointer is positioned at the specified coordinates.

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

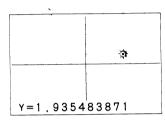
The pointer can be moved left, right, up, and down using the cursor keys. The current position of the pointer is always shown at the bottom of the display.

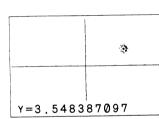




To find the y-coordinate value:

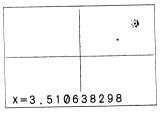
SHIFT X-Y





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

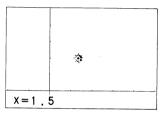
Plot 3.5 SHIFT , 6.5 EXE



If x-y coordinates are not specified for the plot function, the pointer appears at the center of the screen. Set the following range values:

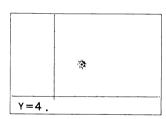
Range Xmin:-2. max:5. scl:1. Ymin:-2. max:10. scl:2.

Plot EXE



To find the Y-coordinate value:

SHIFT X↔Y



- * Attempting to plot a point outside of the preset range is disregarded.
- * The x and y-coordinates of the pointer used in the plot function are respectively stored in the X memory and Y memory.
- * A blinking pointer becomes a fixed point (not blinking) when a new pointer is created.

■ Line function

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

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

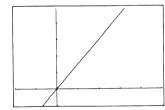
The range values for the graph are as follows:

Range
Xmin:-2.
max:5.
sc!:1.
Ymin:-2.
max:10.
sc!:2.

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

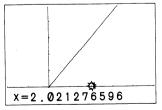
CIS EXE

Graph 3 ALPHA X EXE



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

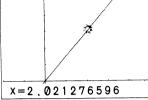
Plot 2 SHIFT , 0 EXE



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

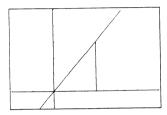
Plot 2 SHIFT , 0 EXE

(Move the pointer up to the point on the graph for y=3x.)



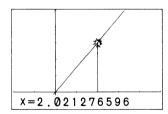
Draw a line using the line function.

Line EXE



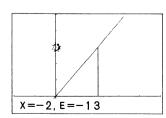
Next, a perpendicular will be drawn from the same point on the graph to the y-axis. First, plot the point on the graph and use the cursor key (\bigcirc) to move the pointer to the y-axis. This can be accomplished using Plot X, Y since the x-y coordinates of the point on the graph are stored in the X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



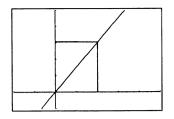


△ ~ □

(Move the pointer to the y-axis.)







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

■ Factor function

The factor function is used to magnify or reduce the range of a graph centered around the blinking pointer provided with the plot function or trace function.

For magnification, the minimum value and maximum value of the range are multiplied by 1/n. For reduction, they are multiplied by n.

Operation

Factor m SHIFT \cdot n EXE $\cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot x$ is magnified m times and y is magnified n times centered around the pointer.

centered around the pointer.

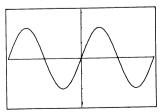
The graph display is cleared when the factor function is executed because of changes in the range values.

Ex. After setting the range values specified below, magnify the graph for $y=\sin x$ centered on the origin.

Range
Xmin:-360.
max:360.
sci:180.
Ymin:-1.6
max:1.6
sci:0.5

Draw the graph for $y=\sin x$ after setting the range values.

Graph sin ALPHA X EXE

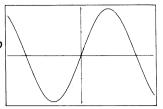


Now use the plot function to blink the pointer at the origin of the graph and then use the factor function to magnify the graph 1.5 times.

Plot : Factor 1.5 :

Graph Sin ALPHA X EXE

* The multistatement function is used to produce the graph in a single step.



The following shows the resulting range values:

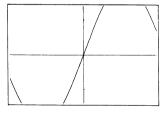
Range Xmin:-240. max:240. sci:180. Ymin:-1.06666667 max:1.06666667 sci:0.5

This indicates that the range values for the x and y-axes are equal to 1/1.5 of their original values.

Now let's try magnifying the graph another 1.5 times.

This time, it is not necessary to input any further commands. The existing graph is magnified by simply pressing **EXE**. Since the original magnification was accomplished using the multistatement function, the replay function becomes operational.

EXE



Now the graph is so large that little of it remains on the display. Let's try to reduce the graph to half its present size to make it more manageable.

The replay function is used to change the magnification value from 1.5 to 0.5.

 \triangleright

Plot :Factor 1.5 :Graph Y=sin X

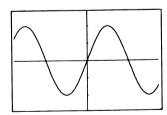
Plot:Factor 1.5 :Graph Y=sin X

0

Plot :Factor 0_5 :Graph Y=sin X

Now execute the function.

EXE



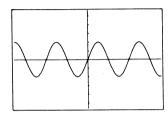
The following display shows the new range values:

Range

Range Xmin: -320. max: 320. scl: 180. Ymin: -1.42222223 max: 1.42222221 scl: 0.5

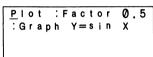
To reduce the graph by half again:

EXE



Now let's double the x-axis and increase the y-axis by 1.5 times.

 \Box



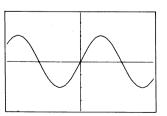
2 SHIFT ,

INS 1

Plot:Factor 2,1 _5:Graph Y=sin X

Now execute the function.

EXE



Using the operations outlined in this section, graphs can be magnified or reduced. In the examples given here, the graphs were magnified and reduced centered around the origin, but any pointer on the display can be used as a central point for magnification and reduction.

■ Instant factor function

The instant factor function can be used to quickly magnify the size of a graph by 2^n or reduce it by $1/2^n$. The change in size is centered at the pointer when it is displayed, and at the center of the graph when the pointer is not displayed.

Operation

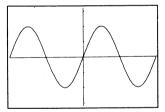
SHIFT \boxtimes ... 2X magnification in both x and y directions. Pressing \boxtimes again magnifies by 2^2 or 4X, and a third press magnifies by 2^3 or 8X.

SHIFT \oplus ... 1/2 reduction in both x and y directions. Pressing SHIFT \oplus again reduces by $1/2^2$ or 1/4, and a third press reduces by $1/2^3$ or 1/8.

Since range contents are switched to their inverse proportions, the graphic display is cleared each time the instant factor function is executed.

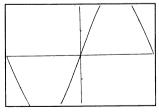
Ex. Graph $y = \cos x$ using the built-in function, and change the size by 2X and 1/2.

Graph sin EXE



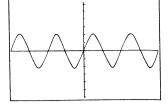
Now magnify the graph 2 X at the center of the display.

SHIFT X



Next, reduce the original $y=\cos x$ graph by 1/2.

SHIFT ÷



In the above examples, the changes in the graph size were performed at the center of the display. If the pointer is shown on the display, the magnification/reduction is applied centered at the pointer.

3-3 GRAPH FUNCTION APPLICATIONS

Even complex equations can be graphically represented. A number of graphs for the equations will be presented in this section.

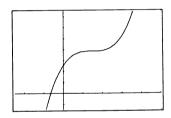
Ex. 1) Draw the graph for the third degree equation, $y=x^3-9x^2+27x+50$.

The range values for the graph are given on the right.

Operation

CIS EXE

Graph ALPHA
$$\times$$
 x^y 3 $-$ 9 ALPHA \times x^2 $+$ 27 ALPHA \times $+$ 50 EXE

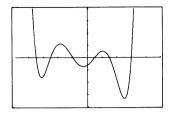


Ex. 2) Draw the graph for the polynomial equation, $y=x^6+4x^5-54x^4-160x^3+641x^2+828x-1260$.

The range values for the graph are given on the right.

```
Range
Xmin:-10.
max:10.
scl:2.
Ymin:-8000.
max:8000.
scl:2000.
```

Operation

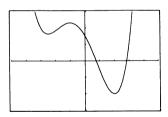


Ex. 3) Find the maximum and minimum for the equation, $y=x^4+4x^3-36x^2-160x+300$.

* If this equation is graphed, the minimum and maximum can be easily understood without differentiation.

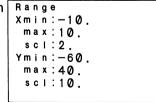
The range values for the graph are given Range xmin:

Operation



Ex. 4) Determine whether the two graphs for equations, $y=x^3-3x^2-6x-16$ and y=3x-11 have a point of tangency.

The range values for the graphs are given Range xmin:



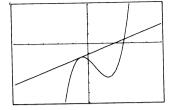
Operation

Cis

Graph ALPHA X xy 3 — 3 ALPHA X x² —

6 ALPHA X - 16 EXE

Graph 3 ALPHA X - 11 EXE



3-4 SINGLE VARIABLE STATISTICAL GRAPHS

- Single variable statistical graphs are drawn in the SD2 mode (SHIFT MODE 区).
- Bar graphs, line graphs, and normal distribution curves can be produced as single variable statistical graphs.
- Function graphs are also possible in the SD2 mode, so graphs of theoretical values and graphs of actual values can be overwritten.
 - * Abs and [₹] cannot be used in the SD2 mode.
- Number of data is determined by expanding memories.
- ullet Graphs are drawn with the x-coordinate as the data range and the y-coordinate as the number of items (frequency) of each data.
- The DT key (☑) is used for data input.
- The CL key (xy) is used for data correction.

■ Drawing single variable statistical graphs

- Procedure
- ① Specify the SD2 mode (SHIFT MODE ⋈).
- 2 Set the range values (Range).
- 3 Expand the memory in accordance with the number of bars ($\underbrace{\text{MODE}}$ \cdot n
- 4 Clear the statistical memories (SHIFT SCI EXE).
- 5 Input data (Data DT ()).
- 6 Draw the graph.

 - Line graph Graph Line EXE

 - * Data input method in step 5 is the same as that for standard deviation computations (see page 48).

Ex. Use the following data to draw a ranked graph.

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

Perform graph preparation in accordance with the following procedure:

- 1) Specify the SD2 mode (SHIFT MODE X).
- 2 Set the range values.

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

Ymax value is set to 20 for the y-axis because the maximum frequency is 15.

Range
Xmin:0.
max:110.
scl:10.
Ymin:0.
max:20.
scl:2.

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

MODE · 11 EXE

** Defm **

Program : 0

Memory : 37

3918 Bytes Free

4 Clear the statistical memory.

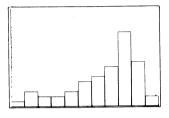
SHIFT ScI EXE

(5) Input the data.

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

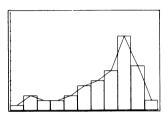
6 First, draw a bar graph.

Graph EXE



Next, overwrite a line graph.

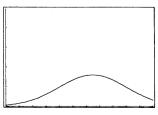
Graph Line EXE



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

Graph Line 1 EXE

Inputting the number 1 causes a normal distribution curve to be drawn.



(Summary)

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

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

- * Keyboard designation of σ is $x \sigma n$. m is \bar{x} .
- The following must be true in the case of range settings: Xmin< Xmax.
- After a bar or line graph is executed, "done" is displayed in the text display.

3-5 PAIRED VARIABLE STATISTICAL GRAPHS

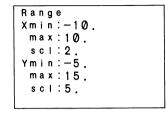
- Paired variable graphs are drawn in the LR2 mode (SHIFT MODE :).
- Paired variable graphs can be drawn as regression lines.
- Standard function graphs can also be drawn in the LR2 mode, so theoretical graphs, data distribution and regression line graphs can be overwritten.
- After data input in the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

Drawing paired variable statistical graphs

- Procedure
- 1) Specify the LR2 mode (SHIFT MODE :).
- 2 Set the range values (Range).
- (SHIFT Sci EXE).
- 4 Input data (x data SHIFT : y data SHIFT : frequency DT).
- 5 Draw the graph (Graph Line 1 EXE).
 - * Data input method in step 4 is the same as that for Regression computation (Page 50).
 - Ex. Perform linear regression on the following data and draw a regression line graph.

x_i	\boldsymbol{y}_i
-9	-2
- 5	-1
-3	2
1	2 3 5 8
4	5
7	8

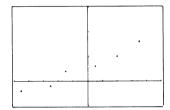
- $\widecheck{\ensuremath{ \begin{tabular}{l} \ensuremath{ \begin{tabula$



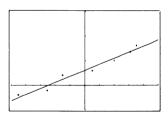
- * According to the general rule of the x-axis range values, the values for x are: $-10 \le x < 10$.
- ③ Clear the statistical memories.

4 Input the data.



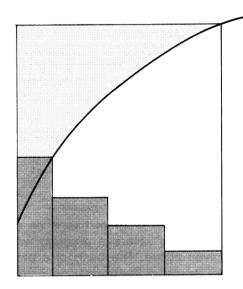


⑤ Draw the graph.



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

4. PROGRAM COMPUTATIONS



4-1 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)	Surfac	e area (S)	Volu	ıme (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 are defined as:

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

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 \(\subseteq 3 \) Numeric value A \(\textit{x}^2 \) \(\textit{EXE} \)

Volume (V): ✓ 2 ÷ 3 ⋈ Numeric value A 🗷 3 EXE

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

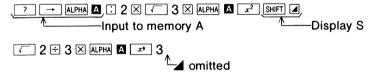
Numeric value A
$$\rightarrow$$
 ALPHA A EXE 2 \boxtimes 7 3 \boxtimes ALPHA A x^2 EXE \boxtimes 8 \boxtimes 2 \boxtimes 3 \boxtimes ALPHA A \boxtimes 9 3 EXE \boxtimes V

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 " \blacktriangle ".

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

When "?" is displayed, calculation commands and numeric values can be input within 111 steps.

The "\[\rightarrow\]" command causes program execution to stop temporarily and the latest formula result or alphanumeric characters and symbols (see page 129) 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. However, if the Base-n mode is specified for base conversion during a program, do not omit the final "\[\rightarrow\]". Here these two commands will be used in the previously presented pro-



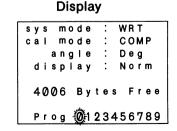
Now the program is complete.

3 Program storage

cedure:

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

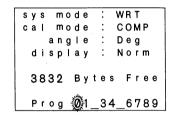




When MODE 2 are pressed, the system mode changes to the WRT mode. Then, the number of remaining steps (see page 106) is indicated. The number of remaining steps is decreased when programs are input or when memories are expanded. If no programs have been input and the number of memories equals 26 (the number of memories at initialization), the number of usable steps should equal 4.006.

The larger figures located below indicate the program areas (see page 108). 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.



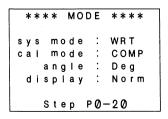
Here the previously mentioned program will be stored to program area P0 (indicated by the blinking zero):

Operation Display EXE (Start storage) ? \rightarrow ALPHA \triangle : 2 × \bigcirc 3 × ALPHA \triangle x² SHIFT \triangle ? \rightarrow A : 2 × \bigcirc 3 × A 2 \triangle \bigcirc 2 \div 3 × A 2 \triangle \bigcirc 2 \div 3 × A 2 \triangle \bigcirc 2 \div 3 × A x y 3 \bigcirc

After these operations are complete, the program is stored.

* The system display appears only while the be key is pressed.

Moisp (Displayed while pressed)



^{*} After the program is stored, press MODE 11 to return to the RUN mode.

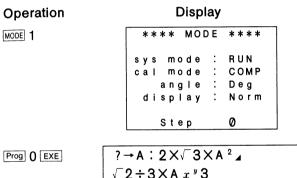
(4) Program execution

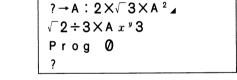
Programs are executed in the RUN mode (MODE 1). The program area to be executed is specified using the Program key.

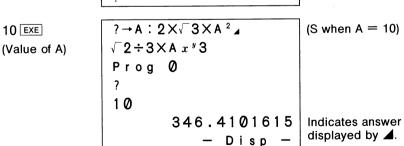
To execute P0: Prog 0 EXE
To execute P3: Prog 3 EXE
To execute P8: Prog 8 EXE

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

Length of one side (A)	Suface area (S)	Volume (V)
10cm	(346.4101615)cm ²	(471.4045208)cm ³
7	(169.7409791)	(161.6917506)
15	(779.4228634)	(1590.990258)







? \rightarrow A: $2 \times \sqrt{3} \times A^2$ EXE (V when A = 10) $\sqrt{2 \div 3 \times A} x^{y} 3$ Prog 0 ? 10 346.4101615 471.4045208 Prog () EXE $\sqrt{2 \div 3 \times 4} x^{y} 3$ Proq 0 ? 10 346.4101615 471.4045208 Prog 0 7 EXE (Value of A) 10 (S when A = 7) 346.4101615 471.4045208 Prog 0 7 169.7409791 - Disp -10 (V when A = 7) EXE 346.4101615 471.4045208 Prog 0 ? 7 169.7409791

161.6917506

471.4045208 Prog 0 EXE Prog 0 ? 7 169.7409791 161.6917506 Prog 0 ? (S when A = 15) 15 EXE 169.7409791 (Value of A) 161.6917506 Prog 15 779.4228634 - Disp -(V when A = 15) EXE 169.7409791 161.6917506 Prog 0 15 779.4228634 1590.990258

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

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

Operation

96

```
Prog 0 EXE (P0 program execution)

10 EXE (Input 10 for A)

EXE (Display V when A = 10)

EXE (Reexecute)

7 EXE (Input 7 for A)

EXE (Display V when A = 7)

:
```

4-2 PROGRAM CHECKING AND EDITING (CORRECTION, ADDITION, DELETION)

Recalling a stored program can be performed in order to verify its contents. After specifying the desired program area using \triangleleft or \triangleright in the WRT mode (MODE 2), the program contents will be displayed by pressing the EXE key. Once the program is displayed, the \triangleright (or \triangleleft , \triangleright) 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)	Surfac	e area (S)	Vol	ume (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 are defined as:

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

2 Programming

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

Numeric value A → ALPHA A EXE	
3 X ALPHA A x² EXE ·····	S
2 ÷ 12 × ALPHA A xy 3 EXE	٧

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

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

3 Program editing

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

Octahedron: ? \longrightarrow ALPHA A : 2 \boxtimes \checkmark 3 \boxtimes ALPHA A \checkmark SHIFT \checkmark 2 \div 3 \boxtimes ALPHA A \checkmark 3 \boxtimes ALPHA A \checkmark SHIFT \checkmark 2 \div 12 \boxtimes ALPHA A \checkmark 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

MODE 2

sys mode: WRT
cal mode: COMP
angle: Deg
display: Norm
3986 Bytes Free
Prog __123456789

EXE

 $? \rightarrow A : 2 \times \sqrt{3} \times A^{2} \boxed{}
\sqrt{2 \div 3} \times A x^{y} 3$

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

DEL DEL

 $? \rightarrow A : \sqrt{3} \times A^{2} \checkmark$ $\sqrt{2 \div 3} \times A \times x^{y} 3$

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

? → A : $\sqrt{3}$ × A 2 $\sqrt{2}$ ÷ 1 2[3]× A x^{y} 3

Insert two characters.

DEL

 $? \rightarrow A : \sqrt{3} \times A^{2} \checkmark$ $\sqrt{2 \div 1} 2 \times A x^{y} 3$

Delete unnecessary 3.

MODE 1

**** MODE ****

sys mode : RUN
cal mode : COMP
angle : Deg
display : Norm

Step 0

Editing cmoplete. 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)cm ²	(117.8511302)cm ³
7.5	(97.42785793)	(49.71844555)
20	(692.820323)	(942.8090416)

Operation

MODE 1

Display

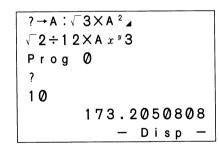
**** MODE ****

sys mode : RUN
cal mode : COMP
angle : Deg
display : Norm

Prog () EXE

?→A:√3×A² ⊿	
$\sqrt{2 \div 12 \times Ax^{9}3}$	
Prog Ø	
?	

10 EXE



EXE

?
$$\rightarrow$$
 A: $\sqrt{3} \times$ A² $\sqrt{2} \div 12 \times$ A x⁹ 3

Prog 0
?
10
173.2050808
117.8511302

Prog 0 EXE	$\sqrt{2 \div 12 \times A} x^y 3$
	Prog Ø
	?
	10
	173.2050808
	117.8511302
	Prog Ø
	?
7.5 EXE	10
1.0 [[
	173.2050808
	117.8511302
	Prog 0
	7.5
	97.42785793
	— Disp —
EXE	1 0
	173.2050808
	117.8511302
	Prog 0
	?
	7.5
	97.42785793
	49.71844555
Prog 0 EXE	117.8511302
	Prog Ø
	?
	7.5
	97.42785793
	49.71844555
	Prog Ø

```
7.5

97.42785793

49.71844555

Prog 0

?

20

692.820323

— Disp —
```

20 EXE

EXE

7.5 97.42785793 49.71844555 Prog 0 ? 20 692.820323 942.8090416

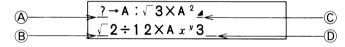
Summary

	Operation	Keys used
Program	 ●WRT mode specification 	MODE 2
check	 Program area specification (Omitted if P0) 	4 b
	Start verification	EXE
	◆ Verification of contents	$\boxed{ \bigcirc } \qquad \boxed{ \boxed } \qquad \boxed$
Correction	 Move the cursor to the position to be corrected. 	4 1 4
	Press correct keys.	
Deletion	Move the cursor to the position to be deleted.	4 D A D
	● Delete	DEL
Insertion	 Move the cursor to the position to be in- 	
	serted into.	
	Specify the insert mode.	INS
	Press desired keys.	

⟨Reference⟩

Cursor movement

Pressing the cursor keys (\triangleleft , \triangleright , \triangle , ∇) causes the cursor to move as follows:



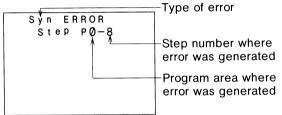
Cursor position	4	D	Δ	abla
A	Invalid	1 position right	Invalid	1 line down (B)
B	1 position left (©)	1 position right	1 line up (A)	End of line (D)
©	1 position left	1 position right (B)	Beginning of line (A)	1 line down (©)
D	1 position left	Invalid	1 line up (©)	Invalid

4-3 PROGRAM DEBUGGING (CORRECTING ERRORS)

After a program has been created and input, it will sometimes generate error message 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 seven error messages.

- ① Syn ERROR (Syntax error) Indicates a mistake in the formula or a misuse of program commands.
- 2 Ma ERROR (Mathematical error)
 Indicates the computation result of a numeric expression exceeds 10¹⁰⁰, 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 113), or that the program area (see page 108) for the Prog command (see page 120) does not contain a program.

- 4 Ne ERROR (Nesting error) Indicates a subroutine nesting overflow by the Prog command.
- Stk ERROR (Stack error) Indicates the computation performed exceeds the capacity of the stack for numeric values or for commands (see page 16).
- 6 Mem ERROR (Memory error) Indicates the attempt to use a memory name such as Z [5] without having expanded memories.
- 7 Arg ERROR (Argument error) Indicates the argument of a command or specification in a program exceeds the input range (i.e. Sci 10, Goto 11).

Further operation will become impossible when an error message is displayed. Press AC, \triangleleft , or \triangleright to cancel the error.

Pressing AC cancels the error and new key input becomes possible. With this operation, the RUN mode is maintained.

Pressing or cancels the error and changes the system mode to the WRT mode. The cursor is positioned at the location where the error was generated to allow modification of the program to eliminate the error.

■ Checkpoints for each type of error

The following are checkpoints for each type of error:

① Syn ERROR

Verify again that there are no errors in the program.

2 Ma ERROR

For 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 P n has been correctly input when Prog n is used.

4 Ne ERROR

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

(5) Stk ERROR

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

6 Mem ERROR

Check to see that memories were properly expanded using " $\underline{\text{MODE}} \ \underline{\text{D}} \ n$ [EXE]" (Defm). When using array-type memories (see page 124), check to see that the subscripts are correct.

(7) Arg ERROR

Check whether values specified by $\boxed{0}$ (Sci) or $\boxed{0}$ (Fix) are within the range of $0 \sim 9$. Also check whether values specified by Goto, LbI, or Prog commands are within 0-9. Also ensure that memory expansion using $\boxed{0}$ (Defm) is performed within the remaining number of steps and that the value used for expansion is not negative.

4-4 COUNTING THE NUMBER OF STEPS

The program capacity of this unit consists of a total of 4,006 steps.

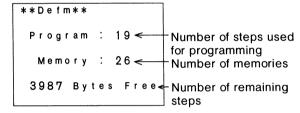
The number of steps indicates the amount of storage space available for programs, and it will decrease as programs are input. The number of remaining steps will also be decreased when steps are converted to memories. (See page 24).

There are two methods to determine the current number of remaining steps:

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

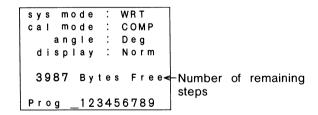
Example:





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





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

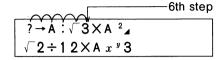
- One function/one step: sin, cos, tan, log, (,), :, A, B, 1, 2, 3, etc.
- One function/two steps: Lbl 1, Goto 2, Prog 8, etc.

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

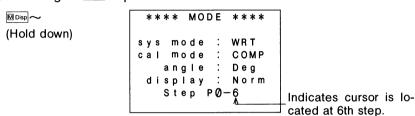
Example:

Present cursor position
$$\rightarrow \boxed{ ? \rightarrow A : \sqrt{3} \times A^2 }_{\sqrt{2} \div 1 \ 2 \times A \ x^9 \ 3}$$

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



The display will show at what step of the program the cursor is currently located as long as [MDISS] is pressed.



4-5 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 4,006 maximum.

Specification of a program area is performed as follows:

RUN mode: Press any key from 0 through 9 after pressing the Prog key.

Then press EXE.

Example:

P 0 Prog O EXE

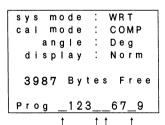
P 8 Prog 8 EXE

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

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

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

Example:



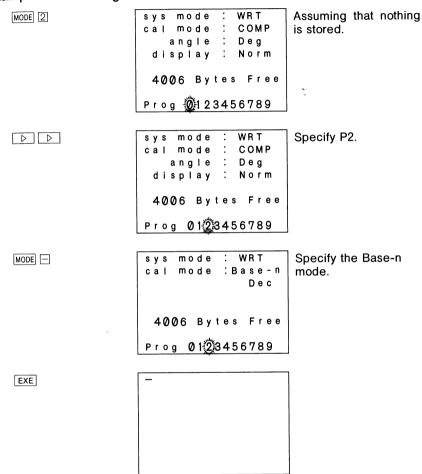
Programs already stored in these program areas.

■ 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.



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

■ Cautions concerning the computation modes

All key operations 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.
- Be sure to include a "▲" at the final result output to return to the previous computation mode when a program execution is terminated. Failure to do so may result in a decimal display or an error.

SD1, SD2 mode

- Among the functions, Abs and ³√ cannot be used.
- Among the program commands, Dsz, > and < cannot be used.

LR1, LR2 mode

- Among the functions, Abs and √ cannot be used.
- Among the program commands, \Rightarrow , =, \pm , Isz, \geq , \leq , Dsz, > and < cannot be used.

4-6 ERASING PROGRAMS

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

■ Erasing a single program

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

Example: Erase the program in P3 only.

Operation	Display	
MODE 3	sys mode: PCL cal mode: COMP angle: Deg display: Norm	P0, P3 and P9 already contain programs.
	3908 Bytes Free	
	Prog	
D D D	sys mode : PCL cal mode : COMP angle : Deg display : Norm	Align cursor with P3.
	3908 Bytes Free	
	Prog _12 45678_	
AC	sys mode : PCL cal mode : COMP angle : Deg display : Norm	Number 3 appears after deletion.
	3951 Bytes Free	

MODE 1

*** MODE ****

sys mode : RUN
cal mode : COMP
angle : Deg
display : Norm

Return to RUN mode

Erasing all programs

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

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

Operation

MODE 3

Display

sys mode: PCL
cal mode: COMP
angle: Deg
display: Norm
3879 Bytes Free
Prog 123 567

SHIFT DEL

S	у	s		m	0	d	е		:		Ρ	С	L		
С	а	١		m	0	d	е		:		С	0	М	Ρ	
			а	n	g	١	е		:		D	е	g		
	d	į	s	р	Ī	а	у		:		N	0	r	m	
	4	0	0	6		В	у	t	е	s		F	r	e e	
		Ī	Ī	-			٠								
	Р	r	0	g		Ö	1	2	3	4	5	6	7	89	
					_	تبوت	<u> </u>			_					-

MODE O

sys m		:	RUN
cal m			
	o d e	:	COMP
a n	gle	:	Deg
disp	iay	:	

4-7 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 commands are used to change the flow of program execution. Programs are executed in the order that they are input (from the lowest step number first) until the end of the program is reached. This system is not very convenient when there are repeat computations to be performed or when it is desirable to transfer execution to another formula. It is in these cases, however, that the jumps commands are very effective. There are three types of jump commands: a simple unconditional jump to a branch destination, conditional jumps that decide the branch destination by whether a certain condition is true or not, and count jumps that increase or decrease a specific memory by one and then decide the branch destination after checking whether the value stored equals zero or not.

Unconditional jump

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

?,
$$\rightarrow$$
, A, :, $\sqrt{}$, 3, \times , A, x^2 , \checkmark , \checkmark , 2, \div , 1, 2, \times , 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 "LbI 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, :, ?,
$$\rightarrow$$
, A, :, $\sqrt{}$, 3, \times , A, x^2 , \checkmark , $\sqrt{}$, 2, \div , 1, 2, \times , A, x^y , 3, \checkmark , Goto, 1

25 steps

Now let's try executing this program.

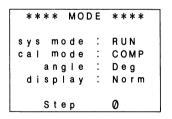
* For details on inputting programs and editing programs, see sections 4-1 and 4-2.

*Henceforth, the displays will only show computation result output.

Operation		Display	
Prog O EXE	?		Stored in P0.
10 EXE		173.2050808	The length of the side=10
EXE		117.8511302	side— iu
EXE	?		
7.5 EXE		97.42785793	The length of the side=7.5
EXE		49.71844555	side—7.5
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

?,
$$\rightarrow$$
, A, $:$, ?, \rightarrow , B, $:$, $\underline{Lbl, 1}$, $:$, ?, \rightarrow , X, $:$, A, \times , X, $+$, B, \blacktriangle , $\underline{Goto, 1}$ 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 "I bl n" to correspond to a "Goto n", an error (Go ERROR) is generated.

Conditional jumps

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

Conditional jumps take on the following form:

Left	Relational	Right	State-	∫ " (^	State-
side	operator	side	ment	ا نا	ment

- ★ represents carriage return function (see page 122).
- * Anyone can be used.

One memory name (alphabetic character from A through Z), constant numeric values or computation formulas ($A \times 2$, 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: =, \neq , \geq , \leq , >, <.

Left side = right side (left side equals right side)

Left side \neq right side (left side does not equal right side)

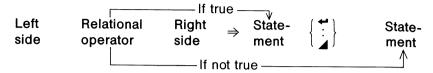
Left side ≥ right side (left side is greater than or equal to right side)

Left side \leq 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 " \Rightarrow " is displayed when $\fill \fill \fill$



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

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

Lbl, 1,
$$\vdots$$
, ?, \rightarrow , A, \vdots , A, \geq , \emptyset , \Rightarrow , $\sqrt{}$, A, \blacktriangleleft , Goto, 1

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

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

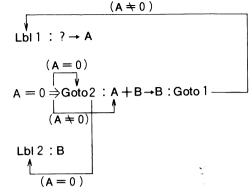
Program

$$0, \rightarrow$$
, B, :,
LbI, 1, :, ?, \rightarrow , A, :, A, =, $0, \Rightarrow$, Goto, 2, :,
A, +, B, \rightarrow , B, :, Goto, 1, :,
LbI, 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 "? \rightarrow A" is stored in memory A by "A=0 \Rightarrow " 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 \rightarrow B which follows ":" is executed, and then Goto 1 returns execution to Lbl 1.

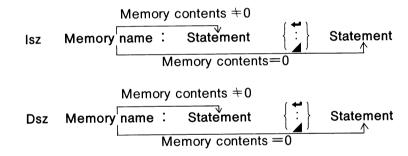
Execution from Lbl 2 will display the sum that has been stored in memory B. Actually, the display command "\(\rightarrow\)" is inserted following B, but here it can be omitted.

The following illustration shows the flow of the program:



Count jumps

The count jumps cause the value in a specified memory to be increased or decreased by 1. If the value does equal 0, the following statement is skipped, and the statement following the next "\(\neq\)", ":" or "\(\neq\)" 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,
$$\emptyset$$
, \rightarrow , A, \vdots , \emptyset , \rightarrow , C, \vdots ,
Lbl, 1, \vdots , ?, \rightarrow , B, \vdots , B, $+$, C, \rightarrow , C, \vdots ,
Dsz, A, \vdots , Goto, 1, \vdots , C, \div , 1, \emptyset

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 \rightarrow 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 \div 10" is executed.

Example: Determine the altitude ϵt 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=Vsin θ t $-\frac{1}{2}$ gt², with g=9.8, with the effects of air resistance being disregarded.

Program

Deg, :,
$$\emptyset$$
, \rightarrow , T, :, ?, \rightarrow , V, :, ?, \rightarrow , S, :,
Lbl, 1, :, Isz, T, :, V, \times , sin, S, \times , T, $-$,
9, •, 8, \times , T, x^2 , \div , 2, \blacktriangleleft , 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.

LbI 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 \rightarrow 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 11 are pressed to terminate the program.

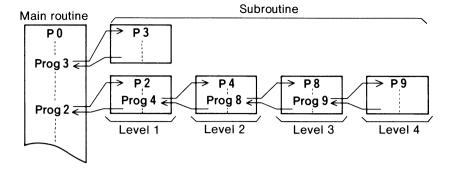
(Summary)

Command	Formula	Operation
Unconditional	LbI n	Performs unconditional jump to
jump	Goto n ($n=$ natural number	LbI n corresponding to Goto n .
	from 0 through 9)	
Conditional	Left Relational Right ⇒	Left and right sides are com-
jumps	side operator side ´	pared. If the conditional expres-
•	(4-)	sion is true, the statement after
	Statement { : Statement	⇒ is executed.
	A ,	If not true, execution jumps to
,	(Relational operators: $=$, \neq ,	the statement following the next
	 >, <, ≥, ≤)	← , : or ⊿ .
		Statements include numeric ex-
		pressions, Goto commands, etc.
Count jumps	Isz Memory name:	Numeric value stored in memory
	Statement { Statement }	is increased (lsz) or decreased (Dsz) by one. If result equals 0, a
	Dsz Memory name:	jump is performed to the state- ment following the next ←, : or ◢.
	Statement { : } Statement	Statements include numeric expressions, Goto commands, etc.
	(Memory name consists of	
	single character from A	
	through Z, A [], 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 "Prog" followed by a number from 0 through 9 which indicates the program area.

Example: Prog 0 ······Jump to program area 0 Prog 2 ······Jump to program area 2

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

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

Example: Simultaneously execute the two previously presented programs to 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, :, ?, \rightarrow , A, :, 2, \times , $\sqrt{}$, 3, \times , A, x^2 , \checkmark , 2, \div , 3, \times , A, x^y , 3 23 steps

Regular tetrahedron

P1 Fix, 3, :, ?,
$$\rightarrow$$
, A, :, $\sqrt{}$, 3, \times , A, x^2 , \checkmark , 2, \div , 1, 2, \times , 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: $\sqrt{}$, 2, \div , 3, \times , A, x^y , 3, \div , 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, :, ?,
$$\rightarrow$$
, A, :, $\sqrt{\ }$, 3, \times , A, x^2

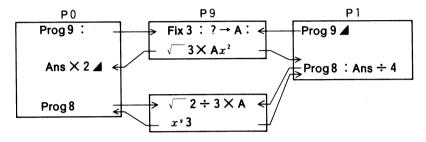
12 steps
P8 $\sqrt{\ }$, 2, \div , 3, \times , A, x^y , 3

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

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 " $2\times$ " of the original octahedron formula was omitted in P9, so when execution returns to P0, "Ans \times 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.

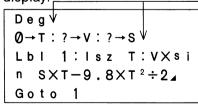


By isolating the common portions of the two original programs and storing them in separate program areas, steps are shortened and programs take on a clear configuration.

■ Carriage return function

With the carriage return function, **EXE** is used in place of **(I)** to separate commands to produce easy-to-read displays.

Deg:
$$0 \rightarrow T$$
: $? \rightarrow V$: $? \rightarrow S$:
Lb! 1: Isz T: VXsi
n SXT-9.8XT²÷2 $_{\perp}$
Goto 1



these two locations. Nothing is displayed at the point where EXE is pressed, and the display advances to the next line.

This makes angle unit setting and looped operations, etc. easier to follow.

Operation procedure

ODE 4 EXE (Press in place of :)	
→ ALPHA T : ? → ALPHA V : ? → ALPHA S	EXE
HIFT LbI 1 : ·····	

* To include the carriage return function in a program that has already been input, first press INS to specify the insert mode and then press EXE. Then, delete the ":".

Deg:
$$0 \rightarrow T$$
: $? \rightarrow V$: $? \rightarrow S$:
Lb! 1: Isz T: VXsi
n SXT-9.8XT²÷2₄
Goto 1

Align the cursor with the ":" following "Deg" and press INS EXE.

Deg

$$[:]0 \rightarrow T : ? \rightarrow V : ? \rightarrow S : Lb I$$

$$1 : Isz T : V \times s in S$$

$$\times T - 9 \cdot 8 \times T^{2} \div 2 \checkmark$$

$$Goto 1$$

Delete the ":".

DEL

Deg
$$[0] \rightarrow T : ? \rightarrow V : ? \rightarrow S : L b I$$

1 : I s z T : V × s i n S × T - 9 . 8 × T $^2 \div 2$ 4 Got o 1

Align the cursor with the " \vdots " following "? \rightarrow S". As above, first insert [EXE] and then delete the " \vdots ".

Deg

$$0 \rightarrow T : ? \rightarrow V : ? \rightarrow S$$

 $\downarrow b \mid 1 : l s z \quad T : V \times s i$
 $n \quad S \times T - 9 \cdot 8 \times T^2 \div 2 \checkmark$
Goto 1

^{*} Carriage return can be used in manual operations by pressing SHIFT EXE.

4-8 ARRAY-TYPE MEMORIES

■ Using array-type memories

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

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

^{*} Brackets are input by ALPHA : and ALPHA EXP.

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

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

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

Using standard memories

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

Using array-type memories

$$\emptyset$$
, \to , Z, $:$, LbI, 1, $:$, Z, $+$, 1, \to , A, $[$, Z, $]$, $:$, Isz, Z, $:$, Z, $<$, 1, \emptyset , \Rightarrow , 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, :, ?,
$$\rightarrow$$
, Z, :,
Z, =, 1, \Rightarrow , A, \triangle , Z, =, 2, \Rightarrow , B, \triangle ,
Z, =, 3, \Rightarrow , C, \triangle , Z, =, 4, \Rightarrow , D, \triangle ,
Z, =, 5, \Rightarrow , E, \triangle , Z, =, 6, \Rightarrow , F, \triangle ,
Z, =, 7, \Rightarrow , G, \triangle , Z, =, 8, \Rightarrow , H, \triangle ,
Z, =, 9, \Rightarrow , I, \triangle , Z, =, 1, \bigcirc 0, \Rightarrow , J, \triangle ,
Goto. 1

Using array-type memories

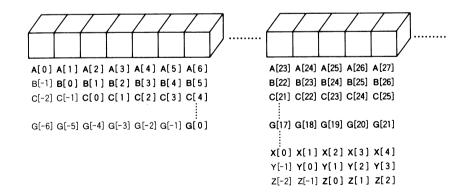
The difference is readily apparent. When using the standard memories, the input value is compared one by one with the value assigned to each memory (i.e. A=1, B=2, . .).

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:



70 steps

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.

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

EXE

Prog (EXE
EXE	
EXE	
EXE	

	1.
	0.
	3.
	4.
1	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.

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,
$$\rightarrow$$
, A, \vdots , Defm, 7, \vdots ,
Lbl, 1, \vdots , ?, \rightarrow , C, [, A,], \vdots ,
?, \rightarrow , C, [, A, +, 1, 5,], \vdots ,
lsz, A, \vdots , A, =, 1, 6, \Rightarrow , Goto, 2, \vdots , Goto, 1, \vdots ,
Lbl, 2, \vdots , 1, 5, \rightarrow , A, \vdots , ?, \rightarrow , B, \vdots ,
B, =, \emptyset , \Rightarrow , Goto, 5, \vdots ,
Lbl, 3, \vdots , B, =, C, [, A,], \Rightarrow , Goto, 4, \vdots ,
Dsz, A, \vdots , Goto, 3, \vdots , Goto, 2, \vdots ,
Lbl, 4, \vdots , C, [, A, +, 1, 5,], \blacktriangleleft , Goto, 2, \vdots ,
Lbl, 5

In this program, memories are used as follows:

x data

Z(1) Z(2) Z(3) Z(4) Z(5) Z(6) Z(7)

Example program 2

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

1,
$$\rightarrow$$
, A, $:$, Defm, 7, $:$,
Lbl, 1, $:$, ?, \rightarrow , C, [, A,], $:$,
?, \rightarrow , R, [, A,], $:$,
lsz, A, $:$, A, $=$, 1, 6, \Rightarrow , Goto, 2, $:$, Goto, 1, $:$,
Lbl, 2, $:$, 1, 5, \rightarrow , A, $:$, ?, \rightarrow , B, $:$,
B, $=$, \emptyset , \Rightarrow , Goto, 5, $:$,
Lbl, 3, $:$, B, $=$, C, [, A,], \Rightarrow , Goto, 4, $:$,
Dsz, A, $:$, Goto, 3, $:$, Goto, 2, $:$,
Lbl, 4, $:$, R, [, A,], \blacktriangleleft , Goto, 2, $:$,
Lbl, 5

Memories are used as follows:

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.

4-9 DISPLAYING ALPHA-NUMERIC CHARACTERS AND SYMBOLS

Alphabetic characters, numbers, computation command symbols, etc. can be displayed as messages. They are enclosed in quotation marks ([MPM]]).

- Alpha-numeric characters and symbols
- Characters and symbols displayed when pressed following ALPHA:
 [,], k, m, μ, n, p, f, space,
 A, B, C, D, E, F, G, H, I, J, K, L, M, N,
 O, P, Q, R, S, T, U, V, W, X, Y, Z
- Other numbers, symbols, calculation commands, program commands

$$0$$
, 1, 2, 3, 4, 5, 6, 7, 8, 9, (,), $\sqrt{}$, ϵ , +, -, \times , \div , ... sin, cos, tan, log, ln, ... =, \pm , \geq , \leq , $>$, $<$, ... A, B, C, D, E, F, d, h, b, o Neg, Not, and, or, xor \bar{x} , \bar{y} , $x\sigma_n$, $x\sigma_{n-1}$, ... $^{\circ}$ ([SHIFT] MODE [4]), $^{\circ}$ ([SHIFT] MODE [5]), $^{\circ}$ ([SHIFT] MODE [6])

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

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

Lbl, 1, :, ?,
$$\rightarrow$$
, X, :, ?, \rightarrow , Y, :, ...

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

Lbl, 1, :, ",
$$X$$
, =, ", ?, \rightarrow , X , :, ", Y , =, ", Y , :, ...

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

Messages are also convenient when displaying result in program computations.

Example:

Lbl,
$$\emptyset$$
, :, ", N, =, ", ?, \rightarrow , B, \sim , C, :, \emptyset , \rightarrow , A, :, Lbl, 1, :, C, \div , 2, \rightarrow , C, :, Frac, C, \pm , \emptyset , \Rightarrow , Goto, 3, :, Isz, A, :, C, =, 1, \Rightarrow , Goto, 2, :, Goto, 1, :, Lbl, 2, :, ", X, =, ", \triangle , A, \triangle , Goto, \emptyset , :, Lbl, 3, :, ", N, O, ", \triangle , Goto, \emptyset 70 steps

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

* Always follow a message with a **\(\Lambda \)** whenever a formula follows the message.

Assuming that the program is stored in P2:

Prog 2 EXE
4096 EXE
EXE
EXE
3124 EXE
EXE
512 EXE
EXE

N = ?	
X =	
	12.
N = ?	
NO	
N = ?	
X =	
	9.

Strings longer than 16 characters are displayed in two lines. When alphabetic characters are displayed at the end of the bottom line, the entire display shifts upwards and the uppermost line disappears from the display.

Prog 0

123+45	168.
852-87	
968+125-65	765.
	1028.
Prog [©] 0	

EXE

123+45	1.00
852-87	168.
0001105 05	765.
968+125-65	1028.
Prog Ø	
ABCDEFGHIJK	LMNOP

↓ After a while

168.
852-87
765.
968+125-65 1028.
Prog 0
ABCDEFGHIJKLMNOP
QRSTUVWXYZ

4-10 USING THE GRAPH FUNCTION IN PROGRAMS

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

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

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

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

 $y=10x-30$

The range values are as follows:

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

Range, (-), 1,
$$\emptyset$$
, , 1, \emptyset , , 2, , (-), 1, 2, \emptyset , , 1, 5, \emptyset , , 5, \emptyset

Next, program the equation for the first graph.

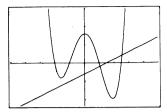
Graph, X,
$$x^y$$
, 4, -, X, x^y , 3, -, 2, 4, X, x^2 , +, 4, X, +, 8, \emptyset

Finally, program the equation for the second graph.

Total 49 steps

When inputting this program, press $\begin{tabular}{l} \end{tabular}$ after input of the ranges and the first equation.

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

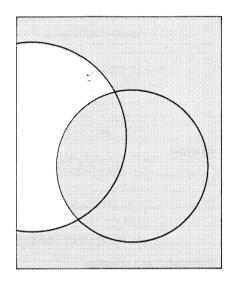


A "\[\begin{align*} \]" can be input in place of the \[\begin{align*} \begin{al

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

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

PROGRAM LIBRARY



⟨Prior to use⟩

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

Then start programming by pressing EXE.

PROGRAM SHEET

Program for Prime factor analysis No. 1

Description

Prime factors of arbitrary positive integers are produced.

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

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

⟨Overview⟩

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

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

Example

<1>

$$119 = 7 \times 17$$

(2)

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

⟨3⟩

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

Preparation and operation

• Store the program written on the next page.

• Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prog O EXE	M ?	11	EXE	3803.
2	119 EXE	7.	12	EXE	END
3	EXE	17.	13	EXE	м?
4	EXE	END	14	987654321 EXE	3.
5	EXE	M ?	15	EXE	3.
6	1234567890 EXE	2.	16	EXE	17.
7	EXE	3.	17	EXE	17.
8	EXE	3.	18	EXE	(After 10)379721.
9	EXE	5.	19	EXE	END
10	EXE	(After 57) 3607.	20		

_				Program												T T	Missante
Line	M	ODE [2]				Р	rogr	am							Notes	Number of steps
1	McI	:															2
2	Lbl	0	:	"	М	"	?	→	Α	:	Goto	2	:				15
3	Lbl	1	:	2	4	Α	÷	2	-	Α	:	Α	=	1	⇒		30
4	Goto	9	:														33
5	Lbl	2	:	Frac	(Α	÷	2)	=	0	⇒	Goto	1	:		48
6	3	-	В	:													52
7	Lbl	3	:	√ <u></u>	Α	+	1	→	С	:							62
8	Lbl	4	:	В	≥	С	⇒	Goto	8	:	Frac	(Α	÷	В		77
9)	=	0	→	Goto	6	:										84
10	Lbl	5	:	В	+	2	→	В	:	Goto	4	:					96
11	Lbl	6	:	Α	÷	В	×	В	_	Α	=	0	⇒	Goto	7		111
12	:	Goto	5	:													115
13	Lbl	7	:	В	4	Α	÷	В	-	Α	:	Goto	3	:			129
14	Lbl	8	:	Α	4											,	134
15	Lbl	9	:	"	Е	N	D	,,	4	Goto	0						145
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28	;										;						
	A		m_i		I	ł				0					V		
nts	В		d]	[P					w		
onte	С		$\sqrt{m_i}$ +	-1	J					Q					X		
ŏ	D				ŀ	(R					Y		
Memory contents	Е				I	_			***************************************	s					z		
Μe	F				N	1				T							
	G		M N						U								

PROGRAM SHEET

Program for Greatest common measure No. 2

Description

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

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

⟨Overview⟩

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

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

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

$$k = 2, 3 \dots$$

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

Example

$$\langle 1 \rangle$$
 $a = 238$

$$a = 23345$$

$$a = 522952$$

$$b = 374$$

$$b = 9135$$

$$b = 3208137866$$

$$c = 34$$

$$c = 1015$$

$$c = 998$$

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	Prog O EXE	A ?	11		
2	238 EXE	В?	12		
3	374 EXE	34.	13		
4	EXE	A ?	14		
5	23345 EXE	В?	15		
6	9135 EXE	1015.	16		
7	EXE	A ?	17		
8	522952 EXE	В?	18		
9	3208137866 EXE	998.	19		
10			20		

													'	NO.		2	
Line	M	ODE	2				Р	rogi	am				•			Notes	Number of steps
1	Lbl	1	:	"	Α	"	?	-	Α	:	,,	В	"	?	-		15
2	В	:														,	17
3	Abs	Α	-	Α	:	Abs	В	→	В	:							27
4	В	<	Α	\Rightarrow	Goto	2	:										34
5	Α	→	С	:	В	→	Α	:	С	-	В	:					46
6	Lbl	2	:	(-)	(Int	(Α	÷	В)	×	В	_	Α		61
7)	→	С	:			****										65
8	С	_	0	⇒	Goto	3	:							2			72
9	В	→	Α	:	С	→	В	:	Goto	2	:						83
10	Lbl	3	:	В	1	Goto	1										90
11								:				<u> </u>	-				
12								-					-				
13			<u> </u>														
14			-											,			
15			-														
16			<u> </u>														
17			-														
18																	
19																	
20																	
21																	
22 23																	
23 24																	
2 4 25																	
26														-	+		
27																	
28																	
	A		a, n	0		H		'		То				·	v		L
ıts	В		b, n		-	I				P					w		
nten	С		n_k			J				Q	 				X		
Memory contents	D		·•R			K				R					Y		
nor	E					L				S	-				Z		
Men	F				-	M				T	-				-		
-	G					N				U					\vdash		
	J					•				10	l						

No.

3

Program for

Definite integrals using Simpson's rule

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

Example

$$\langle 1 \rangle \ a = 0, b = 1, 2m = 10$$

$$I = \int_0^1 \frac{1}{x^2 + 1} d_x = 0.7853981537$$
 $\langle 2 \rangle \ a = 2, b = 5, 2m = 20$

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

Preparation and operation

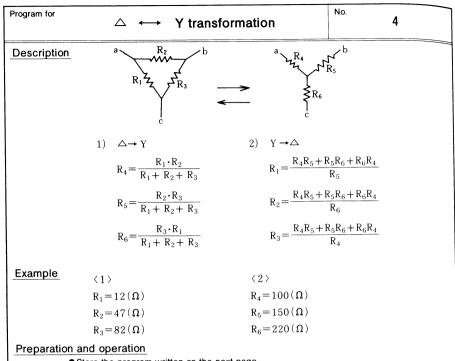
• Store the program written on the next page.

• Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prog O EXE	A ?	11		4
2	0 EXE	В?	12		
3	1 EXE	2 M ?	13		
4	10 EXE	0.7853981535	14		
5	EXE	A ?	15		
6	2 EXE	В?	16		
7	5 EXE	2 M ?	17		
8	20 EXE	0.2662526769	18		
9			19		
10			20		

																_	
Line		DDE [2]				Р	rogr	am							Notes	Number of steps
1	Р0																
2	Lbl	1	:	McI	:		<u> </u>										5
3	"	Α	"	?	→	Α	:	"	В	"	?	-	В	:	,,		20
4	2	М	"	?	→	М	:		!			<u> </u>					27
5	Α	-	G	:	Prog	1	:	Р	-	1	:	(В	_	Α		42
6)	÷	М	→	D	:	М	÷	2	-	0	:	<u> </u>				54
7	Lbl	2	:	G	+	D	-	G	:	Prog	1	:	1	+	Р		69
8	Χ	4	→	1	:							1		-			74
9	G	+	D	-	G	:	Prog	1	:	1	+	Р	X	2	-		89
10	1	:	0	_	1	→	0	:									97
11	0	+	0	⇒	Goto	2	:		<u> </u>								104
12	В	→	G	:	Prog	1	:	1	<u> </u>	Р	-	1	:				117
13	D	X	١	÷	3	4											123
14	Goto	1															125
15																	
16	Р1																
17	1	÷	(G	×	G	+	1)	-	Р						11
18								, , ,									
19								! ! !								Total 136	steps
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
	A		a		I	H				0	m (Number	of repe	titions)	v		
nts	В		b			I		I		P					w		
ntei	С				-+	ı				Q				·	X		
8	D	h	<u>_</u>	<u>-a</u>		ζ				R				******	Y		
Memory contents	E		2	m	-+					S					Z		
Jen					-+	+					-			****	L		
	F		187000			И				Т							
	G		x		ı	1				U							

PROGRAM SHEET



 \bullet Store the program written on the next page.

• Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1	Prog 0 EXE	D→Y:1,Y→D:2?	11	EXE	D→Y:1,Y→D:2?
2	1 EXE	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	M	ODE 2	2				Р	rogr	am							Notes	Number of steps
1	Lbl	1	:	"	D	-	Υ	:	1	,	Υ	-	D	:	2		15
2	"	?	→	N	:												20
3	N	=	2	\Rightarrow	Goto	2	:	N	+	1	→	Goto	1	:			34
4	,,	R	1	=	"	?	-	Α	:								43
5	"	R	2	=	"	?	-	В	:								52
6	,,	R	3	=	,,	?	-	С	:								61
7	Α	+	В	+	С	→	D	:									69
8	,,	R	4	=	"	4	Α	×	В	÷	D	4		-			81
9	"	R	5	=	"	4	В	×	С	÷	D	1		:	:		93
10	"	R	6	=	,,,	1	Α	X	С	÷	D	1					105
11	Goto	1	:		-	-		-									108
12	Lbl	2	:														111
13	"	R	4	=	"	?	-	E	:								120
14	"	R	5	=		?	→	F									129
15	"	R	6	_	,,,	?	→	G	: :								138
16	Е	X	F	+	F	×	G	+	G	×	E	-	Н	<u> </u>			152
17	"	R	1	=	"	1	Н	÷	F	4							162
18	"	R	2	=	,,	1	Н	÷	G					!	:		172
19	"	R	3	=	,,	1	Н	÷	Е	1							182
20	Goto	1			-		1	-						-			184
21						-	-				-				-		
22							-							-			
23					-	-	-							<u> </u>			
24					<u> </u>	_	-										
25							-	-									
26					-	-	-	-						-			
27					-	-	<u> </u>	-			<u> </u>			-	:		
28					<u>:</u>		<u>;</u>	, D.D.	D D	T_	<u>.</u>			<u>:</u>	177		
,	A		R ₁			H	R ₄ R ₅ +	K ₅ K ₆	⊢ K ₆ K ₄	0					V		
ent	В		R ₂		_	I				P Q	-				W		
Memory contents	С		R ₃		-	J					-				X		
J.	D	R ₁	+ R ₂	$+ R_3$		K									Y		
eme	Е		R ₄			L	<u> </u>								Z		
ž	F		R_5			М	[
	G		R ₆			N	For judgement										

No.

Δ

PROGRAM SHEET

Program for

Minimum loss matching

5

Description

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

$$Z_0 \longrightarrow R_1 \longrightarrow Z_1$$

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

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

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 11).

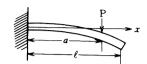
Step	Key operation	Display	Step	Key operation	Display
1	Prog 0 EXE	Z 0= ?	11		
2	500 EXE	Z 1= ?	12		
3	200 EXE	R 1=	13		
4	EXE	387.2983346	14		
5	EXE	R 2=	15		
6	EXE	258.1988897	16		
7	EXE	LMIN =	17		
8	EXE	8.961393328	18		
9			19		
10			20		

Line	N	IODE [Р	rogr	am							Notes	Number of steps
1	,,	Z	0	=	"	?	-	Υ	:								9
2	,,	z	1	=	,,	?	-	Z	:					:			18
3	1	(1	-	Z	÷	Υ)	-	Α	:						29
4	Υ	×	Α	-	R	:	z	÷	Α	-	S	:	Υ	÷	Z		44
5	-	В	:	2	0	×	log	($\sqrt{}$	В	+	$\sqrt{}$	(В	_		59
6	1))	-	Т	:											65
7	"	R	1	=	"	4	R	4				:					73
8	"	R	2	=	"	1	s	4	:								81
9	,,	L	М	1	N	=	"	:	Т								90
10																	
11																	
12		<u>i</u>															
13		<u> </u>															
14																	
15				<u> </u>		<u> </u>						-					
16		!	!			_						-					
17					-							-					
18														<u> </u>			
19		<u> </u>	-		-												
20					-												
21		-					<u> </u>		<u> </u>			<u> </u>					
22			<u> </u>		-							-		<u> </u>			
23		 	<u> </u>	ļ	-	<u> </u>	<u> </u>		<u> </u>								
24		<u> </u>	<u> </u>	<u> </u>		-	-		-	-		<u> </u>		-			
25		-	-		-	-	-			-		!		-			
26		-	-	-			<u> </u>		!	-		:		<u> </u>			
27		-	-	-		-	-		<u> </u>	-		-		<u> </u>			
28		!		71		:	<u> </u>		<u> </u>	<u> </u>	<u> </u>			<u> </u>			
	A		√ 1-	$\frac{z_1}{z_0}$		Н				0	<u> </u>				V		
ents	В		$\frac{z_0}{z_1}$			I				P					W		
Memory contents	С					J				Q					X		
<u>ح</u>	D					K				R		R	1		Y	Z_0	
ΩŪ	Е					L				S		R	2		Z	Z_1	
Σ	F					M				T		Lı	nin				
	G				-+	N				U	1						
					. 1					10	1				1		

PROGRAM SHEET

Program for Cantilever under concentrated load No. 6

Description



E: Young's modulus (kg/mm²)

I : Geometrical moment of inertia [mm4]

a : Distance of concentrated load
 from support [mm]

P: Load [kg]

x : Distance of point of interest from
the support (mm)

Deflection y (mm), Angle of deflection s (°), Bending moment M (kg · mm)

①
$$\ell > x > a$$

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

②
$$x \leq a$$

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

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

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

M = 0 (shearing load Ws = 0)

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

Example

$$E = 4000 \text{ kg/mm}^2$$
 $I = 5 \text{ mm}^4$
 $a = 30 \text{ mm}$

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

$P = 2 \ kg \label{eq:P}$ Preparation and operation

• Store the program written on the next page.

• Execute the program as shown below in the RUN mode (MODE 1)

Step	Key operation	Display	Step	Key operation	Display
1	Prog 0 EXE	E = ?	11	EXE	—10 .
2	4000 EXE	1=?	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	Υ =	16	EXE	-2.57657183
7	EXE	-0 .677 0 833333	17	EXE	M =
8	EXE	S =	18	EXE	0.
9	EXE	-2.505092867	19	Repeat from	step 5
10	EXE	M =	20		

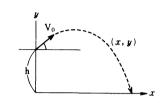
Line	MO	DDE 2	2]				P	rogr	am							Notes	Number of steps
1	Deg	:	,,	Ε	=	,,	?	-	Ε	:	,,	1	=	"	?		15
2	-	1	:	"	Α	=	,,	?	-	Α	:	,,,	Р	=	"		30
3	?	→	Р	:													34
4	Lbl	1	:	"	Х	=	"	?	→	Х	:						45
5	Χ	≤	Α	\Rightarrow	Goto	2	:										52
6	,,	Υ	=	"	4	Р	×	Α	x²	÷	(2	X	E	×		67
7	I)	Χ	(Α	÷	3	-	Х)	4						78
8	,,	S	=	"	4	tan ⁻¹	((-)	Р	×	Α	x^2	÷	· (2		93
9	X	E	X	1))	4	,,	М	=	,,	4	0	4			107
10	Goto	1	:														110
11	Lbl	2	:														113
12	"	Υ	=	"	4	Р	×	Х	x ²	÷	(2	×	E	×		129
13	1)	X	(Х	÷	3	_	Α)	4	!					139
14	,,	S	=	,,	4	tan 1	(Р	X	Χ	÷	(2	×	E		154
15	×	1)	X	(Х	_	2	X	Α))	4				167
16	,,	М	=	,,	4	Р	×	(Х	_	Α)	4				180
17	Goto	1															182
18													;				
19																	
20																	
21																	
22													-				
23					<u> </u>	-						-	-	-			
24													-				
25						-		-					-				
26					!	:	-	:				:	:	-	-		
27					-	-						-					
28	.				: ,	:						1	<u> </u>	:	<u> </u>		L
	A		a			Н				0					V		
ents	В					I		I		P			P		W		
onte	С					J				Q					X	<i>x</i>	
ry c	D					K				R					Y		
Memory contents	Е		Е			L				S					Z		
Me	F					M				Т							
	G					N				U	 				+		



PROGRAM SHEET

	Program for Parabolic movement	No. 7
1		

Description



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

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

$$g = 9.8 \left(\text{m/s}^2 \right)$$

 $V_0 (m/s)$

a (°)

Δ t (sec.)

h (m)

Example

Initial velocity $V_0 = 130 \, (\text{m/sec.})$

Initial angle a = 25 (°)

Height $h = 0 \ (m)$

 $\Delta t = 0.5 (sec.)$

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	Prog O EXE	∨ 0= ?	11	EXE	T =
2	130 EXE	A = ?	12	EXE	0.5
3	25 EXE	H = ?	13	EXE	X =
4	O 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	Repeat from	step 11
8	EXE	0.	18		
9	EXE	Υ =	19		
10	EXE	0.	20		

														110.		/	
Line	MC	DDE [2	2]				Р	rogr	am							Notes	Number of steps
1	Deg	:	0	→	s	:											6
2	"	٧	0	=	,,,	?	→	V	:	,,	Α	=	"	?	→		21
3	Α	:	"	Н	=	"	?	→	Н	:	"	Т	=	"	?		36
4	→	Т	:														39
5	Lbl	1	:	٧	×	cos	Α	×	Ś	-	Х	:	٧	×	sin		54
6	Α	X	S	_	9		8	×	s	x²	÷	2	+	Н	→		69
7	Υ	:															71
8	"	Т	=	"	4	s	4	s	+	Т	→	s	: `				84
9	"	Χ	=	"	4	Х	4	"	Υ	=	"	4	Υ	4			98
10	Y	≥	0	⇒	Goto	1		<u> </u>				<u> </u>	<u> </u>	<u> </u>			104
11																	
12																	
13																	
14																	
15													!	:			
16					-	-	-						!				
17						-			-			-		<u> </u>	1		
18						-											
19																	
20				<u> </u>													
21																	
22						-		<u> </u>				-	!	<u> </u>	:		
23				<u> </u>	<u> </u>	-	<u> </u>	-	-						.		
24				-	-	-	-	<u>. </u>	-			-	-	-	 		
25				-	<u> </u>	<u> </u>		<u> </u>	-	-		-	-	-			
26				-	!	-			-	<u>:</u>		:	-	-	1		
27				<u> </u>	!	! 		-	<u> </u>			<u>; </u>	-	<u> </u>			
28				<u> </u>	: -	<u>:</u>	<u> </u>	<u> </u>	!	<u> </u>	_	:	!	<u> </u>			
,	A		a			H		h		0	-				V	V ₀	
ents	В					I				P					W		
ont	С					J				Q					X		
ry c	D				T	K				R					Y		
Memory contents	Е					L				S					Z		
Σ	F					М				Т			 ∆ t				
	G					N				U	1				++		

PROGRAM SHEET

Program for Normal distribution 8

Description

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

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

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



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

P = 0.2316419

 $C_3 = 1.78147937$

 $C_1 = 0.31938153$

 $C_4 = -1.821255978$

 $C_2 = -0.356563782$

 $C_5 = 1.330274429$

Example

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

Preparation and operation

• Store the program written on the next page.

• Execute the program as shown below in the RUN mode (MODE 1).

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

													1			U	i
Line	N	ODE 2	2]				Р	rogr	am							Notes	Number of steps
1	,,	Х	=	,,	?	→	Х	:									8
2	1	÷	(1	+	0		2	3	1	6	4	1	9	×		23
3	Х)	→	Т	:	1	÷	$\sqrt{}$	(2	X	π)	×	e ^x		38
4	((-)	Х	x2	÷	2)	→	Q	:							48
5	"	Р	Х	=	"	4	1	_	Q	×	(0		3	1		63
6	9	3	8	1	5	3	×	Т	+	(-)	0		3	5	6		78
7	5	6	3	7	8	2	×	Т	x ²	+	1		7	8	1		93
8	4	7	9	3	7	×	Т	x y	3	+	(-)	1		8	2		108
9	1	2	5	5	9	7	8	×	Т	x y	4	+	1		3		123
10	3	0	2	7	4	4	2	9	×	Т	x y	5)				136
11														-			
12																	
13														-			
14								_						:			
15						<u> </u>				-							
16 17										-				-			
18		+				<u> </u>				_							
19						<u> </u>	<u> </u>			<u> </u>							
20										-							
21				_						-			-				
22										-							
23						-				<u> </u>							
24														-			
25										-							
26											:						
27											:						
28																	
	A					Н				0					v		
nts	В					I				P					w		
nter	С					J				Q			ø t		X	x	
00 /	D					K				R		•			Y		
Memory contents	Е					L				S	+-				Z		
Mer	F					M				T	-		,		-		
	G					N N				U			t		++		
	G					IN											

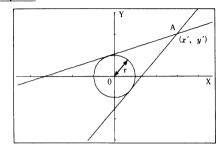
PROGRAM SHEET

Program for

Circle and points of tangency

9

Description



 $Circle\ formula$

 $x^2+y^2=r^2$

Formula for tangent lines passing through point A (x', y')

$$y-y'=m(x-x')$$

No.

* m is the tangent line slope

Draw a line from point A (x', y') to a circle with radius r, and determine the slope m and intercept b (=y'-mx'). Also, read the coordinates of the tangent using the trace function, and use the factor function to magnify the graph.

Example

(NOTE)

• r=x' generates an Ma ERROR.

Preparation and operation

 \bullet Store the program written on the next page.

	A	Н	0	V	
nts	В	I	P	W	
contents	С	J	Q	X	
	D	K	R	Y	
Memory	Е	L	S	Z	
Me	F	М	Т		
	G	N	U		

No.

9

Line	MC	DDE 2]				Р	rogr	am							Notes	Number of steps
1	Р0	;															
2	Prog	1	•														3
3	"	Х	x ²	+	Υ	x ²	=	R	x ²	44							13
4	R	=	"	?	→	R	4										20
5	Prog	2	4														23
6	"	(Х	,	Υ)	4										30
7	Х	= :	"	?	→	Α	4										37
8	"	Υ	=	,,	→	В	4										45
9	Plot	Α	,	В	4												50
10	R	x2	(Α	x ²	+	В	x ²	_	R	<i>x</i> ²)	-	Р	44		65
11	(√	Р		Α	В)	(R	x ²	_	Α	<i>x</i> ²)	x ⁻¹		80
12	-	М	44														83
13	Lbl	6	4														86
14	Graph	М	(Х	÷	Α)	+	В	4							96
15	"	М	=	"	4	М	4										103
16	"	В	=	"	4	В	_	М	Α	4							113
17	Lbl	0	44	-													116
18	"	Т	R	Α	С	Ε	?	44									124
19	Υ	Е	S	→	1	44											130
20	N	0	⇒	0	"	:	?	-	Z	44							140
21	1	→	S	:	z	=	1	→	Goto	1	4						151
22	Z	=	0	⇒	Goto	2	:	Goto	0	*							161
23	LbI	2	44														164
24	((-)	Α	В	_	√	Р)	(R	x ²	_	Α	x ²)		179
25	x-1	→	N	**									:		.		183
26	Graph	N	(Х	-	Α)	+	В	4			:				193
27	"	М	=	"	4	N	4										200
28	"	В	=	"	4	В	_	N	Α	4							210
29	LbI	5	4														213
30	"	Т	R	Α	С	Ε	?	*					:				221
31	Υ	Ε	S	→	1	4											227
32	N	0	→	0	, ,,	:	?	-	Z	4							237
33	2	-	s	:	Z	=	1	⇒	Goto	1	44		:				248
34	Z	=	0	⇒	Goto	3	:	Goto	5	+							258
35	LbI	1	+														261
36	"	Т	R	Α	С	Е	"	4									269

Line	М	ODE [2	2]			•	Р	rogr	am							Notes	Number of steps
1	"	Factor	N	:	N	=	,,	?	→	F	:	Factor	F	*			283
2	Prog	2	:	s	=	1	⇒	Goto	9	4							293
3	S	=	2	\Rightarrow	Graph	М	(Х	_	Α)	+	В	4			307
4	Graph	N	(Х	_	Α)	+	В	4							317
5	Goto	3	4														320
6	LbI	9	+														323
7	Graph	М	(Х	_	Α)	+	В	4							333
8	Prog	1	:	Prog	2	:	Goto	6	4								342
9	LbI	3	4														345
10	"	E	N	D	"												350
11				-													
12	P1			-			-										
13		(-)			7	,	4	٠	7	,	1	,	(-)	3			15
14	1	,	3		1	,	1										22
15																	
16	P2			-								-					
17	Graph	$\sqrt{}$	(R	x ²	_	X	x ²)	4		-					10
18	Graph	(-)	√	(R	x ²	_	Х	x^2)							20
19				-			-					-					
20				-			-					-				Total 39	2 steps
21				 			-					:					
23				-			-	-				-			,		
24				-			-					-					
25				-			!			-					-		
26				-			-			-		<u>:</u>	:				
27				:			-					-					
28				!			:					:					
29				-			-			:		-					
30				:			-										
31				!			-	-									
32				:			-										
33												:		-	-		
34				:			:	:				-					
35				:			-	:			-	-	:				
36				:								!	:				
100			<u> </u>									:				L	

	Circle and points of tang	gency 9
Step	Key operation	Display
1	Prog 0 EXE	Prog 0 X ² +Y ² =R ² R=?
2	1 EXE	
3	EXE	Prog 0 X ² +Y ² =R ² R=? 1 done (X, Y) X=?
4	3 EXE 2 EXE	*

Program	Circle and points of tangenc	y No. 9
Step	Key operation	Display
5	EXE	
6	EXE	3 Y=? 2 done done M= 0.3169872981 - Disp -
7	EXE	2 done done B= 0.3169872981 B= 1.049038106 - Disp -
8	EXE	M=

Prograi	Circle and points of tangend	ey ^{No.} 9
Step	Key operation	Display
9	0 EXE	
10	EXE	YES⇒1 NO⇒0; ? 0 done M= 1.183012702 — Disp —
11	EXE	? 0 done M= 1.183012702 B= -1.549038106 - Disp -
12	EXE	M= 1.183012702 B= -1.549038106 TRACE? YES⇒1 NO⇒0 ?

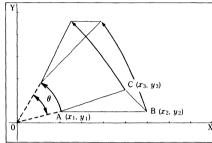
Prograr	Circle and points of tangend	y No. 9
Step	Key operation	Display
13	1 EXE	-1.549038106 TRACE? YES⇒1 NO⇒0 ? 1 TRACE - Disp -
14	Trace	x=-1.3
15		x=0.8
16	SHIFT X-Y	Y=-0.6026279442

Prograi	Circle and points of tangenc	y No. 9						
Step	Key operation	Display						
17	EXE	-1.549038106 TRACE? YES⇒1 NO⇒0 ? 1 TRACE Factor N:N=?						
18	4 EXE							
19	EXE	NO⇒0 ? 1 TRACE Factor N:N=? 4 done END						
20								

PROGRAM SHEET

Program for Rotation of figures No. 10

Description



Coordinate conversion formula $(x, y) \rightarrow (x', y')$ $x' = x \cos \theta - y \sin \theta$ $y' = x \sin \theta + y \cos \theta$

Draw a figure that represents a degree rotation of a triangle.

Example

Draw the figure of the triangle (A (2, 0.5), B (6, 0.5), C (5, 1.5)) rotated 45°

$\langle NOTE \rangle$

- The blinking point can be moved using the cursor keys.
- To terminate the program, press the AC key during graph display.
- A triangle cannot be drawn if the converted coordinates (E' (set the value of x to 5.)) exceed the preset range values.

Preparation and operation

• Store the program written on the next page.

	A	x_1	Н	y'_1	0		V	
nts	В	<i>y</i> 1	I	x'_2	P		W	
contents	С	x_2	J	y'_2	Q	θ	X	
1 .	D	y 2	K	x'_3	R		Y	
Memory	Е	x_3	L	y' ₃	S		Z	
ž	F	y 3	M		Т			
	G	x'_1	N		U			

Line	М	ODE 2	2]				Р	rogr	am							Notes	Number of steps
1	Range	(-)	0		4	,	9	,	1	,	(-)	0		8	,		15
2	5		4	,	1	:	Deg	**					:				23
3	,,	(Χ	1	,	Υ	1)	4								32
4	Χ	1	=	"	?	-	Α	4									40
5	"	Υ	1	=	"	?		В	4								49
6	Plot	Α	,	В	4												54
7	Х	→	Α	:	Υ	-	В	4									62
8	,,	(Х	2	,	Υ	2)	4	,							71
9	Χ	2	=	,,	?	-	С	4									79
10	,,	Υ	2	=	"	?	-	D	+								88
11	Plot	С	,	D	4												93
12	Х	→	С	:	Υ	-	В	44							;		101
13	,,	(Х	3	,	Υ	3)	4								110
14	Х	3	=	,,	?	-	Е	44									118
15	"	Υ	3	=	"	?	-	F	4								127
16	Plot	Е	,	F	4												132
17	Х	-	E	:	Υ	-	F	4									140
18	Lbl	1	4														143
19	Line	:	Plot	Α	,	В	:	Line	:	Plot	С	,	D	:	Line		158
20	4					:											159
21	,,	Α	N	G	L	Е	:	Deg	"	?		Q	-				172
22	Α	cos	Q	_	В	sin	Q	→	G	-							182
23	Α	sin	Q	+	В	cos	Q	→	Н	-							192
24	Plot	G	,	Н	4												197
25	С	cos	Q	_	D	sin	Q	→	1	-							207
26	С	sin	Q	+	D	cos	Q	→	J	-							217
27	Plot	1	,	J	:	Line	4										224
28	Ε	cos	Q	_	F	sin	Q	→	K	-							234
29	Ε	sin	Q	+	F	cos	Q	-	L	4							244
30	Plot	K	,	L	:	Line	*										251
31	Plot	G	,	Н	:	Line	4										258
32	Cls	:	Plot	С	,	D	:	Plot	E	,	F	:	Goto	1			272
33																	
34																Total 272	steps
35																	
36																	

Prograi	Rotation of figures	No. 10
Step	Key operation	Display
1	Prog 0 EXE	Prog Ø (X1, Y1) X1=?
2	2 EXE 0.5 EXE	X=2.
3	EXE	(X1,Y1) X1=? 2 Y1=? 0.5 done (X2,Y2) X2=?
4	6 EXE 0.5 EXE	X=6.

Program	Rotation of figures	No. 10						
Step	Key operation	Display						
5	EXE	(X2, X2=? 6 Y2=? 0.5 (X3, X3=?	done					
6	4.5 EXE 1.5 EXE	x=4.						
7	\triangleright ~ (Set the value of x to 5.)	X=5.						
8	EXE							

Prograi	Rotation of figures		No.	10
Step	Key operation		Displa	ау
9	EXE	(X3, X3=? 4.5 Y3=? 1.5	Y 3) E : D e g	done done ?
10	45 EXE			
11	Repeat above procedure from step 8.			
12				

Program for	Rotation of figures	No. 10
Step	Key operation	Display
13		ī.
14		
15		
16		

PROGRAM SHEET

Program for

Graph variation by parameters

11

Description

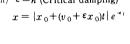
Damped vibration

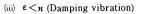
(i) $\varepsilon > n$ (Overdamping)

$$\begin{aligned} & P_{1} = -\epsilon + \sqrt{\epsilon^{2} - n^{2}} , & P_{2} = -\epsilon - \sqrt{\epsilon^{2} - n^{2}} \\ & x = \frac{v_{0} - x_{0} P_{2}}{P_{1} - P_{2}} e^{P_{11}} - \frac{v_{0} - x_{0} P_{1}}{P_{1} - P_{2}} e^{P_{21}} \end{aligned}$$

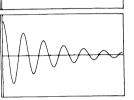


(ii) $\epsilon = n$ (Critical damping)





$$x = e^{-\epsilon_1} |x_0 \cos \sqrt{n^2 - \epsilon^2} t + \frac{v_0 + \epsilon x_0}{\sqrt{n^2 - \epsilon^2}} \cdot \sin \sqrt{n^2 - \epsilon^2} t|$$



Example

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

(1)
$$\varepsilon = 0.1$$

(2)
$$\varepsilon = 0.2$$

$$(3) \epsilon = 0.2$$

$$n = 1.5$$

$$x_0 = 2.5$$

$$n = 0.2$$

$$n = 0.18$$

$$x_0 = z$$

$$x_0 = 2.5$$
 $x_0 = 2$ $x_0 = -2$

$$v_0 = 1$$

$$v_0 = 1$$
 $v_0 = 0.6$

$$v_0 = 1.5$$

Preparation and operation

• Store the program written on the next page.

	Α	x_0	Н	0)		V	
nts	В	<i>v</i> ₀	I	P	2	$P_1 = -\epsilon + \sqrt{\epsilon^2 - n^2}$	W	
contents	С	$\sqrt{n^2-\epsilon^2}$	J	Q	Ş	$P_2 = -\epsilon - \sqrt{\epsilon^2 - n^2}$	X	t
	D		K	R	₹		Y	x
Memory	E	ε	L	S	S		Z	
Σ	F		M	Т	Γ			
	G		N	n U	J			

														No.		11	
Line	M	ODE [2]				P	rog	ram							Notes	Number of steps
1	Rad	+						:	:				:	:	1		2
2	Range	0	,	2	5	,	5		(-)	3	,	3	,	1	+		17
3	"	E	Р	s	1	L	0	N	=	"	?	-	Ε	4			31
4	"	N	=	"	?	→	N	+									39
5	"	Х	0	=	"	?	-	Α	-								48
6	"	٧	0	=	"	?	-	В	-								57
7	E	>	N	⇒	Goto	1	44										64
8	Е	=	N	→	Goto	2	44		:								71
9	\	(N	x^2	_	Ε	x^2)	-	С	4		:				82
10	Graph	e x	((-)	E	Х)	(Α	cos	(С	Х)	+		97
11	(В	+	Е	Α)	С	x-1	sin	(С	Х))	44		112
12	Goto	0	4									:					115
13	LbI	1	44														118
14	(-)	E	+	$\sqrt{}$	(E	x^2	-	N	x ²)	-	Р	44			132
15	(-)	E	_	√	(Ε	x^2	-	N	x^2)	→	Q	44			146
16	Graph	(В	-	Α	Q)	(Р	_	Q)	x-1	e x	(161
17	Р	Х)	_	(В	-	Α	Р)	(Р	-	Q)		176
18	x-1	e x	(Q	Х)	**										183
19	Goto	0	4														186
20	LbI	2	4														189
21	Graph	(Α	+	(В	+	E	Α)	Х)	e x	((-)		204
22	Ε	Х)	4													208
23	LbI	0															210
24																	
25																Total 21	0 steps
26												:					
27																	
28																	
29																	
30																	

Nο

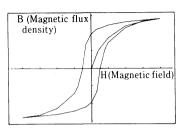
Program	Graph variation by parameter	rs No.						
Step	Key operation	Display						
	Prog O EXE 0.1 EXE	Prog 0 EPSILON=? 0.1 N=?						
1	1.5 EXE 2.5 EXE	$N = ?$ 1.5 $X \emptyset = ?$ 2.5						
		∨ Ø = ?						
2	1 EXE							
3	O.2 EXE 0.2 EXE 2 EXE	Prog 0 EPSILON=? 0.2 N=? 0.2 X0=? 2 V0=?						
4	0.6 EXE							

Prograi	Graph variation by paramete	rs No.
Step	Key operation	Display
	Prog 0 EXE 0.2 EXE	Prog 0 EPSILON=? 0.2
5	0.18 EXE	N = ? 0.18
	(-) 2 EXE	X 0 = ? -2 V 0 = ?
6	1.5 EXE	
7		
8		

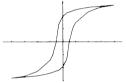
PROGRAM SHEET

Program for Hysteresis loop No. 12

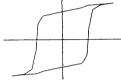
Description



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



Soft magnetic substance



Ferromagnetic substance

Example

Hysteresis curve of soft magnetic material

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

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

	10	11	12	13	14	15	16	17
Н	0	-0.3	-0.5	-0.8	-1.0	-2.0	-3.0	-4.0
В	0.96	0.66	0	-0.53	-0.72	-1.15	-1.33	-1.4

* Within 20 data items.

Preparation and operation

• Store the program written on the next page.

	A	Number of data items	Н	0		V	
nts	В	Number of data items in the main loop	I	P		W	
contents	С		J	Q		X	
	D		K	R		Y	
Memory	Е		L	S		Z	
₹	F		M	T			$Z(1) \sim Z(20) B$
	G	F(1)~F(20) H	N	U			

Line		ODE [Р	rogr	am							Notes	Number of steps
1	Range	(-)	4		7	,	4		7	,	1	,	(-)	1			15
2	5	5	,	1		5	5	,	0		5	44					27
3	Defm	2	0	4													31
4	,,	N	0		SPACE	0	F	SPACE	D	Α	Т	Α	,,	?	→		46
5	Α	4	Lbl	9	44												51
6	,,	М	Α	1	N	SPACE	L	0	0	Р	*						62
7	N	0		SPACE	0	F	SPACE	D	Α	Т	Α	"	?	-	В		77
8	44																78
9	В	>	2	0	\Rightarrow	Goto	9	4									86
10	1	→	С	:	Plot	0	,	0	44								95
11	LbI	0	:	,,	Н	=	,,	?	-	F	(С)	44			109
12	,,	В	=	"	?	→	Z	(С)	4				:		120
13	Plot	F	(С)	,	Z	(С)	:	Line	4				133
14	С	+	1	-	С	4		:			:			:			139
15	С	+	Α	+	1	\Rightarrow	Goto	0	+								148
16	Α	_	В	+	1	-	D	-									156
17	Lbl	1	:	Plot	(-)	F	(D)	,	(-)	Z	(D)		171
18	:	Line	44					:		:							174
19	D	+	1	-	D	+		:		:	:						180
20	D	+	Α	+	1	⇒	Goto	1	4	:	:			:	:		189
21	,,	E	N	D	"									:			194
22																	
23																	
24				:			:	:		1	:			Mei	mory	20×8=160	
25				:	:		:	:			:			:			
26															:	Total 35	4 steps
27																	
28				:													
29				:						:							
30					!			:			:				:		
31					:		:			:	:			:			
32				:	:						:			:			
33														:			
34				:				:			:				:		
35							:			:	:				_		
36					<u> </u>		<u> </u>	!			-	·		:	:		

Progran	Hysteresis loop	No. 12
Step	Key operation	Display
1	Prog 0 EXE	Prog 0 NO. OF DATA?
2	17 EXE	Prog Ø NO. OF DATA? 17 MAIN LOOP NO. OF DATA?
3	12 EXE	Prog Ø NO. OF DATA? 17 MAIN LOOP NO. OF DATA? 12 H=?
4	0.4 EXE 0.5 EXE	

Prograi	Hysteresis loop	No. 12
Step	Key operation	Display
5	1.0 EXE 0.86 EXE	
6	Input data in order. : : :	
7	EXE	-1.33 done H=? -4 B=? -1.4 done
8	G++T	

No. 13

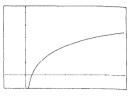
Program for		N	lo.
	Regression curve		13

Description

i Logarithmic regression curve ${\it Regression formula:} \quad y = {\it A} + {\it Bln} x$

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

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

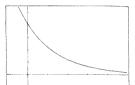


ii Exponential regression curve

Regression formula: $y = A \cdot e^{Br}$

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

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

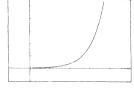


iii Power regression curve

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

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

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



* See page 176 for an example.

Preparation and operation

• Store the program written on the next page.

	A	A or ln A	Н	$\sum (\operatorname{In} x)^2$	0		V	$\sum x$
ents	В	В	I		Р	$\sum oldsymbol{y}^2$	W	п
contents	С	$\Sigma \ln x$	J		Q	$\sum y$	X	x data
	D	$\Sigma \ln y$	K		R	$\sum xy$	Y	y data
Memory	Е	XΣln y	L		S	For selection of 1~3	Z	
ž	F	$Y \sum \ln x$	M		Т			
	G	$\Sigma (\ln x \cdot \ln y)$	N		U	Σx^2		

Line		ODE [Р	rogr	am							Notes	Number of steps
1	Р0	SHIFT			→	LR	2										
2	Scl	:	Cls	:	0	→	С	~	Н	4							10
3	,,	Range	0	K	?	,,	4										17
4	,,	D	Α	Т	Α	SPACE	ı	N	~	Ε	N	D	-	4			31
5	Α	С	→	Prog	1	SPACE	Е	Х	E	"	44						42
6	Lbl	1	4														45
7	"	Х	:	,,	?	→	X	44									53
8	,,	Υ	:	,,	?	-	Υ	4									61
9	In	Х	+	С	→	С	:	In	Υ	+	D	-+	D	:	Х		76
10	In	Υ	+	Ε	-	Е	:	Υ	ln	Χ	+	F	→	F	:		91
11	In	Х	×	In	Υ	+	G	→	G	:	(ln	Х)	x^2		106
12	+	н	→	Н	4												111
13	Χ	,	Υ	DT	1												116
14	Goto	1															118
15																	
16	P1	MODE	\oplus	→	СОМР												
17	,,	Υ	=	Α	+	В	In	Х	SPACE	-	1	4					12
18	Υ	=	Α	×	e x	(В	Х)	SPACE	→	2	**				25
19	Υ	=	Α	×	Х	x y	В	SPACE	SPACE	→	3	44					37
20	1	~	3	_ :	,,	?	-	S	44								46
21	S	=	1	\Rightarrow	Prog	7	+										53
22	S	=	2	\Rightarrow	Prog	8	4										60
23	S	=	3	\Rightarrow	Prog	9	4										67
24	,,	Ε	N	D	,,												72
25																	
26	` P7	SHIFT	MODE	\oplus	-	LR	2										
27	(W	F	_	С	Q)	(W	Н	_	С	x^2)	x ⁻¹		15
28	→	В	:	(Q	_	В	С)	W	x^{-1}	→	Α	4			29
29	Graph	Α	+	В	In	Х	A										36
30	"	Α	:	,,	4	Α	4										43
31	,,	В	:	,,	4	В	4										50
32																	
33																	
34																	
35																	
36																	

PROGRAM SHEET

Program for Regression curve No. 13

Example

Perform exponential regression of the following data:

x i	2.2	5.6	9.5	13.8	18.0	23.2	29.9	37.8
y i	35.6	28.1	23.0	17.9	12.9	10.2	6.2	4.0

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

Range values:

Preparation and operation

• Store the program written on the next page.

	A	Н	0	V
alts	В	I	P	w
contents	С	J	Q	X
	D	K	R	Y
Memory	E	L	S	Z
ž	F	M	Т	
	G	N	U	

Line	М	MODE 2 Program											Notes	Number of steps			
1	P8	SHIFT	MODE	÷	-	LR	2	:	:			;	:				•
2	(W	Е	_	٧	D)	(w	U	_	٧	x ²)	x-1		15
3	-	В	:	(D	-	В	٧)	W	x-1	-	Α	4			29
4	Graph	e*	Α	X	e ^r	В	Х	4									37
5	"	Α	:	"	4	e z	Α	4									45
6	,,	В		,,	4	В	4										52
7																	
8	P 9		MODE	÷	-	LR	2										
9	(W	G	_	С	D)	(W	Н	_	С	x ²)	x-1		15
10	→	В	:	(D	_	В	С)	W	x -1	-	Α	+			29
11	Graph	e*	Α	X	Х	x y	В	4									37
12	,,	Α	:	"	4	e*	Α	4					<u> </u>				45
13	"	В	:	"	4	В	4										52
14																	
15																Total 34	4 steps
16										-		-	<u>;</u>				
17					-					-		-	<u> </u>				
18		-			-			-	-	-	-	-	<u> </u>	-			
19					;			-	-	-	-	-	-				
20								-	-	-		-	-	-			
21		-	:		-				:		-	-					
22		-	-		-			-	-			-	-				
23					-			-	!	-		:	-	-	-		
24					-				-	-		-	-	-			
25 26		:						-	:	-			-	-			_
27					:			:	:			-	-				
28		:			:			:	:	:	:	:	-	:			
26 29		-			-								-				
30		:		_	-					-		-		-			
31								-	-	-		-	-		\vdash		
32		-						_	-	<u> </u>	-	-	-	<u> </u>			
33					:			-	:	:	-	:	:	-		77, 174,54	
34		:			:			:	:	:	-	:	:	-	:		
35					-				-		-	-	 	-			-
36	-	!		_	:			-	<u> </u>	_		-	<u> </u>				
J0								i					:				

No.

13

Progran	Regression curve	No. 13
Step	Key operation	Display
1	(Range setting check)	Prog Ø Range OK? — Disp —
2	Set range values. Range (-) 10 EXE 50 EXE 10 EXE (-) 10 EXE 55 EXE 10	Range Xmin:-10 max:50 sc:10 Ymin:-10 max:55 sc:10
3	After data input is complete, press the AC key and execute the program in Prog 1.	Prog Ø Range OK? DATA IN ~END→ AC→Prog 1 EXE X:?
4	2.2 EXE 35.6 EXE	DATA IN ~END→ AC→Prog 1 EXE X:? 2.2 Y:? 35.6 2.2 — Disp —

Prograi	Regression curve	No. 13						
Step	Key operation	Display						
5	EXE	DATA IN ~END→ AC→Prog 1 EXE X:? 2.2 Y:? 35.6 2.2 X:?						
6	Input data in order.							
7	4.0 EXE	6.2 29.9 X:? 37.8 Y:? 4.0 37.8 — Disp.—						
8	GT							

Prograi	Regression curve	No. 13
Step	Key operation	Display
9	Prog 1 EXE	Prog 1 Y=A+B n X \rightarrow 1 Y=A×e (BX) \rightarrow 2 Y=A×X x^y B \rightarrow 3 1~3:?
10	2 EXE (Select exponential regression).	
11	Trace	x=-4.893617021
12	Move pointer to X=20	x=20.

Progran	Regression curve	No. 13
Step	Key operation	Display
13	SHIFT X-Y	Y=11.86149086
14	EXE	Y=A×e (BX) $\rightarrow 2$ Y=A×X x^y B $\rightarrow 3$ 1~3:? 2 done A: 40.68214077 — Disp —
15	EXE	1~3:? 2 done A: 40.68214077 B: -0.06162460519 - Disp -
16	EXE	1~3:? 2 done A: 40.68214077 B: -0.06162460519 END

CASIO PR

PROGRAM SHEET

14

Program for Parade diagram

Description

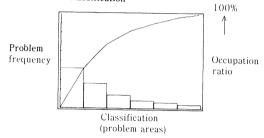
One example of a parade diagram application is problem solving in QC activities. The problem is quantitatively analyzed based on actual data concerning its extent, and the main points demanding attention are determined.

Horizontal axis: Problem classification

(Item 6 in this example)

Vertical axis: (Right) Occupation ratio

(Left) Problem extent in each classification



Example

Create a parade diagram using the data on the right.

Problem areas	Frequency
A	105
В	65
С	35
D	20
E	15
Others	10

Preparation and operation

• Store the program written on the next page.

	A	Input data	Н	О		V	
ents	В		I	Р		W	n
contents	С		J	Q		X	Count of data
1	D		K	R		Y	
Memory	Е		L	S	Display count	Z	Sum of data
Σ	F		M	Т			$Z(1) \sim Z(6)$
	G		N	U			

Line	L	ODE [rogr	am							Notes	Number of steps
1	Р0	SHIFT	MODE	X	-	SD2									1		
2	ScI	:	McI	:	Defm	6	4										7
3	Range	0	,	6	,	1	,	0	,	5	0	0	,	5	0		22
4	44	:						:									23
5	LbI	1	-														26
6	,,	D	Α	Т	Α .	"	?	→	Α	4				1	:		36
7	Х	;	Α	DT	-												41
8	Х	+	1	→	Х	:	Χ	_ ≤	5	\Rightarrow	Goto	1	4				54
9	Range	,	,	,	,	w	,	W	÷	1	0	4					66
10	Graph	4													-		68
11	Plot	0	,	0	-												73
12	1	-	S	+													77
13	LbI	2	-														80
14	Z	(S)	+	Z	→	Z	-				:		1		89
15	Plot	s	,	Z	:	Line	4								:		96
16	S	+	1	-	s	:	S	≤	6	\Rightarrow	Goto	,2	+				109
17	Graph	W													:		111
18		:						:	:						:		
19		:							:						vemo	ry 6×8=48	
20		:							:						:		
21															:	Total 159	steps
22																	
23																	
24															:		
25		-							:	:							
26															:		
27														:	:		
28															-		
29																	
30														:	:		
31			:												:		
32		1	:		1									:	:		
33										:					1		
34			:										:				
35			-												-		
36		:	-						:	!					1		

Progran	Parade diagram	No. 14
Step	Key operation	Display
1	Prog 0 EXE	Prog Ø DATA?
2	105 EXE	Prog Ø DATA? 105 DATA?
3	65 EXE	Prog Ø DATA? 105 DATA? 65 DATA?
4	Input data in order.	

Program	Parade diagram	No. 14
Step	Key operation	Display
5	10 EXE (Bar graph display)	
6	(Parade diagram display)	
7		
8		

CASIO PROGRAM SHEET

Drogram	Program for No.										
Progran	II IOI			'							
Desc	Description										
Exam	nple										
Prep	aration and operation	<u>n</u>									
LT						Ď:I					
Step	Key operation	Display	Step	Key operat	ion	Display					
1			11								

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		

Line	[MODE 2			—— Р	rogr	am							Notes	Number of steps
1				:		;				:					
2				-	-	1				:					
3				:											
4															
5															
6				<u> </u>											
7										-					
8															
9				<u>:</u>	-					-					
10					-	-				-					
11				-		-				-	 	-			
12				-		-	:	<u> </u>		<u> </u>					
13				-	-	-	: :			!	-			1	
14				<u> </u>	-	!	-			:		-			
15		+		-	-	!	-			-					
16				+	-	 				1					
17				-	-	 	-			<u> </u>	:				
18				+	-	:	-			-	:				
19	<u> </u>			+	-	-	-			:	-			-	
20 21	-		-		-	:				:	:		-		
22	_	+ + + +	-	+-	-	-	1			:	-		-		
23	<u> </u>		-	+		:			-	-					
24	-		+		-	!	-			-	-	-			
25	<u> </u>				-	:			-	1	:		-		
26	\vdash		-	:	-	:	:			:	-				
27	 		- 			:	:			:	-	-			
28	 		-	-	-	:				:	!				
	A			Н		•		0	Π				v	1	
ıts	В			I				P					w		
inte	С			J				Q	T				X		
00 >	D			K				R				***************************************	Y		
Memory contents	E			L				S					Z		-
Σ	F			М				Т	1				П		
	G			N				U	-				\Box		
													4		

CASIO PROGRAM SHEET

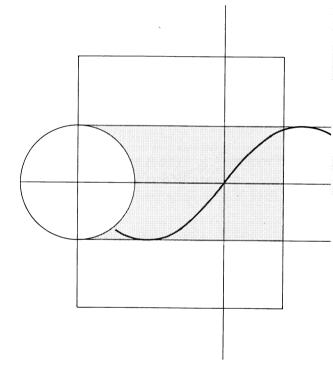
Prog	gram	for			No.	
Ex	amp	ole				
			*			
İ						
PI	repa	aration and operation				
\vdash	Α	Н	()	V	
S	В	I		P	w	
Memory contents	С	J		2	X	
l io	D	K		R	Y	
٥	_	L		S	Z	
eŭ	E			Т	L	
Σ	F	M				
	G	N		U		

	No.		
1			

Line	MODE 2	Prog	ram						Notes	Number of steps
1			-							
2										
3		:	-							
4		į								
5										
6										
7		:								
8			-			:				
9										
10			-							
11				<u> </u>		:		 :		
12			-	<u> </u>				 		
13										
14			-							
15		-	<u> </u>							
16			<u> </u>		-					
17			-							
18			-							
19			-	-						
20										
21			<u> </u>							
22			<u> </u>							
23				<u>:</u>						
24			-		-					
25			<u> </u>							
26			-	-	;		:	-		
27			-	<u> </u>	:		:	 -		
28			-	-		:	:	-		
29			-	!	-		-	-		
30			<u> </u>	<u> </u>	-					
31				<u> </u>	-			-		
32			-	:		:		-		
33			-	<u> </u>	-	;	-	:		
34					-	<u> </u>				-
35						<u> </u>		 <u>; </u>		
36					<u> </u>	<u> </u>		<u>:</u>		

Progran	n for	No.
Step	Key operation	Display

REFERENCE MATERIAL



■ Manual computations

Mode speci- fication	COMP mode (MODE +)	Four arithmetic computations and function computations.
	Base-n mode (MODE □)	Binary, octal, decimal, hexadecimal conversions and computations, logical operations.
	SD1 mode (MODE ⋈)	Standard deviation computations (1-variable statistical computations).
	LR1 mode (MODE ⊕)	Regression computations (paired variable statistical computations).
	SD2 mode (SHIFT MODE ⊠)	For production of single variable statistical graphs. (Bar graphs, line graphs, normal distribution curves)
	LR2 mode (SHIFT MODE ÷)	For production of paired variable statistical graphs. (Regression lines)
Functions	Type A func- tions	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 , $\sqrt{}$, $3\sqrt{}$, 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 $x \sim A$ (A to the 1/Bth power), Pol (A,B), Reç (A,B) * A and B are numeric values.
	Immediately executed functions	Displayed value changed with each press of a key. [ENG, ENG, (5, 7, 7)]

Binary, octal, decimal, hex- adecimal computations	Setting num- ber system	Decimal \cdots Dec EXE (Dec = $\sqrt{\ }$) Hexadecimal \cdots Hex EXE (Hex = x^2) Binary \cdots Bin EXE (Bin = log) Octal \cdots Oct EXE (Oct = \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 \cdots SHIFT d ($d = \sqrt{\ }$) Hexadecimal \cdots SHIFT b ($b = \log$) Octal \cdots SHIFT b ($b = \log$)
	Logical opera- tions	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 xor Exclusive logical sum of each bit
Standard de-	Data clear	SHIFT SCI EXE (Sci = AC)
viation com- putations	Data input	Data (;frequency) DT (DT = T) * Frequency can be omitted.
	Data deletion	Data (;frequency) CL (CL = x²) * Frequency can be omitted.
	Result display	Number of data (n)

Regression	Data clear	SHIFT ScI EXE (ScI = AC)
computations		, , , , , , , , , , , , , , , , , , , ,
	Data input	x data, y data (; frequency) \Box
		$(DT = \sqrt{})$
		* Frequency can be omitted.
	Data deletion	x data, y data (;frequency) $\boxed{\mathtt{CL}}$
		$(CL = x^y)$
		* Frequency can be omitted.
	Result display	Number of data (n)
		$\cdots\cdots\cdots \qquad \text{ALPHA} n \text{EXE} (n = 3)$
	-	Sum of x (Σx)
		Sum of y (Σy)
		Sum of squares of x ($\sum x^2$)
		$\begin{array}{cccc} & & & & & & \\ & & & & & \\ & & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & $
		Sum of squares of y ($\sum y^2$)
		Sum of products of x and y ($\sum xy$)
		Mean of $x(\bar{x})$ SHIFT \bar{x} EXE $(\bar{x} = 1)$
		Mean of y (\bar{y}) \cdots SHIFT \bar{y} EXE $(\bar{y} = 4)$
		Population standard deviation of x
		(x_{σ_n}) SHIFT x_{σ_n} EXE $(x_{\sigma_n} = 2)$
		Population standard deviation of y
		(y_{σ_n}) SHIFT y_{σ_n} EXE $(y_{\sigma_n} = 5)$
		Sample standard deviation of x
		$(x\sigma_{n-1})$ ····· SHIFT $x\sigma_{n-1}$ EXE $(x\sigma_{n-1})$ = 3)
		Sample standard deviation of y
		$(y_{\sigma_{n-1}}) \cdots $ SHIFT $y_{\sigma_{n-1}}$ EXE $(y_{\sigma_{n-1}} = 6)$ Constant term of regression formula
		(A) $\cdots \cdots \cdots \circ $ SHIFT $\land \in XE (\land = 7)$
		Regression coefficient (B)
		Correlation coefficient (r)
		$\cdots\cdots\cdots\cdots\cdots$
1		Estimated value of x (\hat{x})
		$y \text{ data } \text{SHIFT } \hat{x} \text{ EXE } (\hat{x} = X)$
		Estimated value of $y(\hat{y})$
		$x ext{ data SHIFT } \hat{y} ext{ EXE } (\hat{y} = \div)$

Special func- iions	Ans function	The latest result obtained in manual or program computations is stored in memory. It is recalled by pressing Ans. * Mantissa of numeric value is 10 digits.
	Replay function	 After computation results are obtained, the computation formula can be recalled by pressing either or b. If an error is generated, pressing either or b 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 "\(\rightarrow\)", the computation result to that point is displayed.
	Memory ex- pansion	The Number of memories can be expanded from the standard 26. Memories can be expanded in units of one up to 500 (for a total of 526). Eight steps are required for one memory expansion. MODE In number of memories to be expanded EXE.

Graph function	Range function	Graph range settings XmaxMaximum value of x XminMinimum value of x XsclScale of X-axis (space between points) YmaxMaximum value of y YminMinimum value of y YsclScale of Y-axis (space between points)
	Trace function	Moves pointer (blinking dot) on graph. $x-y$ coordinates can be read.
	Plot function	Marks pointer (blinking dot) at any coordinate on the graph display.
	Line function	Connects with a straight line two points created with plot function.
	Factor function	Magnifies or reduces a graph using pointer (blinking dot) as center.

■ Program computations

Program	Input mode	WRT mode (MODE 2)
input	Computation mode	Mode that conforms with program specified by: MODE ⊕, MODE ⊕, MODE ⋈, MODE ⊕.
	Program area specification	Cursor is moved to the desired program area number (P0 through P9) using and , and EXE is pressed.
Program ex- ecution	Execution mode	RUN mode (MODE 11)
	Program area specification	Execution starts with Prog program area No. EXE. Program area No.: 0~9
Program	Input mode	WRT mode (MODE 2)
editing	Program area specification	Cursor is moved to the desired program area number (P0 through P9) using or , and exe are pressed.
	Editing	Cursor is moved to position to be edited using , , , a or . • Press correct key for corrections. • Press DEL for deletions. • Press INS to specify insert mode for insertion.
Program	Erase mode	PCL mode (MODE 3)
erasing	Erasing a program in a single program area	Cusor 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 McI (McI = DEL).

Program commands	Unconditional jump	Program execution jumps to the LbI n which corresponds to Goto n . * $n = 0$ through 9
	Conditional jumps	If conditional expression is true, the statement after "⇒" is executed. If not true, execution jumps to the statement following next "→", ":" or "△". True F: Formula R: Relational operator S: Statement * The relational operator is: =, ‡, >, <, ≥, ≤.
·	Count jumps	The value in a memory is increased or decreased. If the value does not equal 0, the next statement is executed. If it is 0, a jump is performed to the statement following the next "+", ": " or "4". Increase When V \(\) \
		When V = 0 S: Statement V: Value in memory
	Subroutines	Program execution jumps from main routine to subroutine indicated by Prog n ($n = 0$ through 9). After execution of the subroutine, execution returns to the point following Prog n in the original program area.

■ Error messages

Message	Meaning	Countermeasure
Syn ERROR	①Computation formula contains an error. ②Formula in a program contains an error.	①Use dor to display the point where the error was generated and correct it. ②Use dor to display the point where the error was generated, press AC and then correct the program in the WRT mode.
Ma ERROR	 Computation result exceeds computation range. Computation is performed outside the input range of a function. Illogical operation (division by zero, etc.) 	① ② ③ Check the input numeric value and correct it. When using memories, check that the numeric values stored in memories are correct.
Go ERROR	 No corresponding LbI n to Goto n. No program stored in program area P n which corresponds to Prog n. 	 Correctly input a Lbl n to correspond to the Goto n, or delete the Goto n if not required. Store a program in program area P n to correspond to Prog n, or delete the Prog n if not required.
Ne ERROR	Nesting of subroutines by Prog n exceeds 10 levels.	 Ensure that Prog n is not used to return from subroutines to main routine. If used, delete any unnecessary Prog n. Trace the subroutine jump destinations and ensure that no jumps are made back to the original program area. Ensure that returns are made correctly.

Stk ERROR	 Execution of 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 ex- panded.	 Expand memories using MODE
Arg ERROR	Incorrect argument specification for a command that requires an argument.	Correct the argument. • Sci n, Fix n: n= natural number from 0 through 9. • Goto n, Lbl n, Prog n: n = natural number from 0 through 9. • Defm n: n = natural number between 0 to the number of remaining steps.

■ Input range of functions (general principles)

Function name	Input range
$\sin x$, $\cos x$, $\tan x$	$ x \le 9 \times 10^9$ degree
	$ x \leq 5 \times 10^7 \text{m}$ rad
	$ x = 3 \times 10^{-36} \text{ fad}$
$\sin^{-1}x$, $\cos^{-1}x$	$ x \leq 1$
$tan^{-1}x$	x = 1
e^x	$-10^{100} < x \le 230.2585092$
$\sinh x$, $\cosh x$	$ x \le 230.2585092$
tanhx	$ x \le 200.2505092$
$\sinh^{-1}x$	$ x < 5 \times 10^{99}$
$\cosh^{-1}x$	$1 \le x < 5 \times 10^{99}$
$tanh^{-1}x$	$ x \leq x < 0$
$\log x$, $\ln x$	$10^{-99} \le x < 10^{100}$
10 ^x	$-10^{100} \le x < 100$
\sqrt{x}	$0 \le x < 10^{100}$
x^2	$ x < 10^{50}$
$x^{-1}(^{1}/x)$	$ x < 10^{100}, x \neq 0$
$\sqrt[3]{x}$	$ x < 10^{100}$
x!	$0 \le x \le 69$ (x is an integer.)
x^{y}	When $x < 0$, $y = n$, $1/2n + 1$ (n is an integer)
	$x=0 \rightarrow y > 0$
	When $x > 0$, $-10^{100} < y \log x < 100$
$y\sqrt{x}(x^{1/y})$	$x \ge 0, y \ne 0$
Poi(x, y)	$\sqrt{x^2+y^2} < 10^{100}$
$\operatorname{Rec}(r, \theta)$	$ r < 10^{100}, \theta \le 9 \times 10^9 \text{ degree}$
, , ,	$ \theta \le 5 \times 10^7 \mathrm{m}$ rad
	$ \theta < 10^{10}$ gra
	1° 1 ` 10 9:u

Binary number	(Positive) 111111111111111 $\ge x \ge 0$
	(Negative) 1111111111111111 $\ge x \ge$
	100000000000000
Octal number	(Positive) $1777777777777777777777777777777777777$
	(Negative) $3777777777777777777777777777777777777$
Hexadecimal	(Positive) 7 FFFFFFF $\geq x \geq 0$
number	(Negative) FFFFFFFF $\geq x \geq 80000000$
Decimal→	$ x \leq$ 9999999. 999. If degrees, minutes and
sexagesimal	seconds exceed a total of 11 digits, the higher
	(degrees, minutes) values will be given priority,
	and displayed in 11 digits.
Statistical com-	$ x < 10^{50}, y < 10^{50}, n < 10^{100}$
putation	

- * 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!, $\sqrt[3]{x}$, and accuracy is sometimes affected.
- * In $\tan x$, $|x| \neq 90^{\circ} \times (2n+1)$, $|x| \neq \pi/2 \text{rad} \times (2n+1)$. $|x| \neq 100$ gra (2n+1), (n is an integer.)
- * With $\sinh x$ and $\tanh x$, when x = 0, errors are cumulative and accuracy is affected.

SPECIFICATIONS

Model: fx-7500G Computations

Basic computation functions:

Negative numbers, exponents, parenthetical addition/subtraction/multiplication/division(with priority sequence judgement function-true al-

gebraic 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, 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 y, sum of squares of x, 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.

26 standard (526 maximum)

Computation

range:

 $\pm 1 \times 10^{-99} \sim \pm 9.99999999 \times 10^{99}$ and 0. Internal operation uses 13-digit mantissa.

Rounding:

Memories:

Performed according to the specified number of

significant digits or the number of specified de-

cimal places.

Programs

Number of steps:

4,006 maximum

Jump function:

Unconditional jump (Goto), 10 maximum

Conditional jump (=, \pm , >, <, \ge , \le)

Count jumps (Isz, Dsz)

Subroutines:

9 levels

Number of stored programs:

10 maximum (P0 to P9)

Check function:

Program checking, debugging, deletion, addi-

tion, etc.

Graph function

Built-in function graphs:

(20 types) sin, cos, tan, sin-1, cos-1, tan-1, sinh, cosh, tanh, sinh⁻¹, cosh⁻¹, tanh⁻¹, log, ln, 10^s, e^s,

 x^{2} , $\sqrt{}$, $\sqrt{}$, x^{-1}

Graph commands: Graphs:

Graph, Range, Plot, Trace, Factor, Line, X → Y User generated functions, statistical graphs (bar

graphs, line graphs, normal distribution curves,

regression lines), Instant Factor

Common section

Power supply:

Three lithium batteries (CR2025)

Power consump-

0.06 W

tion: Battery life:

Approximately 70 hours on type CR2025.

Auto power off:

Power is automatically switched off approx-

imately 6 minutes after last operation.

Ambient tempera-

0°C-40°C(32°F-104°F)

ture range:

Dimensions:

Unfolded: $7H \times 126W \times 145mmD$

 $(^{1}/_{4}"H \times 5"W \times 5^{3}/_{4}"D)$

14H × 126W × 74mmD Folded: $(^{1}/_{2}"H \times 5"W \times 2^{7}/_{8}"D)$

Weight:

140 g (4.9 oz) icluding batteries