CASIO FX-9000P

OPERATION MANUAL

CASIO®

英

PREFACE

This manual covers the basic configuration of the CASIO **FX-9000P** Personal Computer. This volume provides information on the basic configuration of the **FX-9000P**. It will serve as an introduction for people who have never used a computer.

- The **FX-9000P** can be operated conversationally via a keyboard and video display screen. Therefore, it can be used easily by people without any computer experience.
- One-key commands can be input from the keyboard to assure efficient key-stroke operation.
- The **FX-9000P** is equipped with a graphic function. Graphs and geometric patterns can be displayed for use by businessmen and scientists.

The following calculations can be performed by the **FX-9000P**.

1. **Manual calculation, in the command mode.**
2. **Program calculation during program execution.**

This personal computer, therefore, combines two useful functions: an advanced program calculation function which is unique to electronic computers, and a handy manual calculation function which has long been performed by conventional calculators.

CASIO sincerely hopes that **FX-9000P** users, with the information provided by this manual, can effectively solve their problems by utilizing the machine's built-in functions.

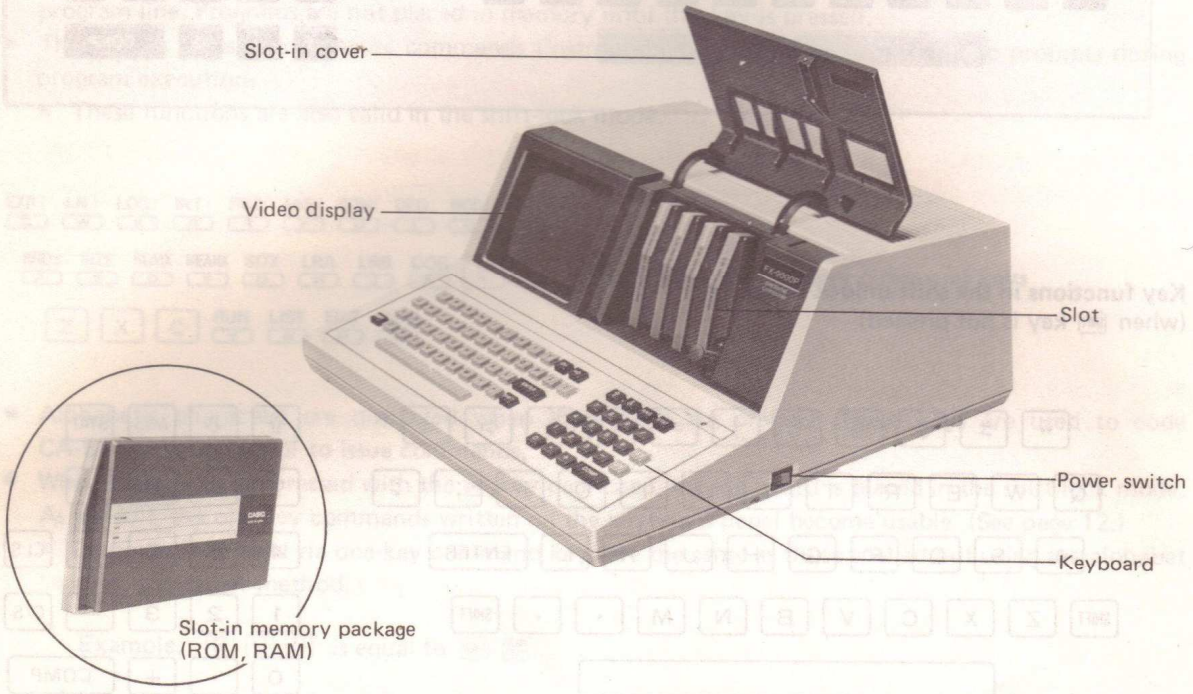
CONTENTS

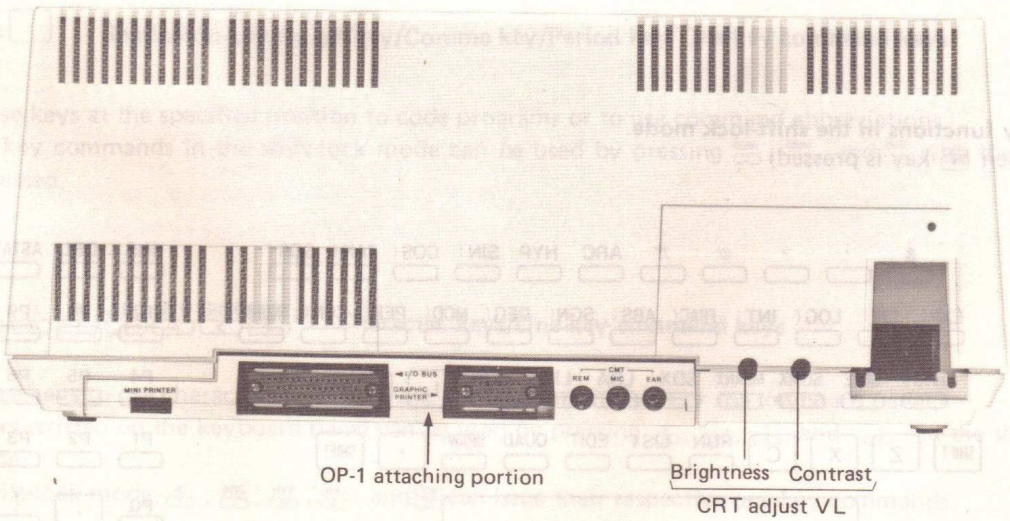
PRECAUTIONS

- Never attempt to disassemble the computer. It is constructed of very delicate electronic components.
- Do not use the computer with its ventilation holes covered, or in a poorly-ventilated area. The ventilation holes in the case are provided to prevent rapid temperature increases. Do not store or use this computer in a place where the temperature is extremely high or low, or subject to dramatic change.
- Do not store or use the computer in a place exposed to the direct rays of the sun, or near a heater or in an extremely humid or dusty place.
- Be careful not to allow water or other liquids, or metal fragments, to enter the computer.
- Plug the computer securely into a power outlet. do not use an outlet which is overloaded.
- Turn the power switch OFF only when no program is being executed.
- The power switch must be OFF when this computer is not in use.
- The power cord should be disconnected from the wall outlet when the computer is not in use for long periods of time.
- Use of this computer near radios, television sets, or similar appliances may cause radio frequency interference. Conversely, the computer may be affected by equipment which generates strong magnetic fields.
- Do not use volatile liquids to clean the computer. Instead, wipe it clean with a dry, soft cloth, or with cloth tightly wrung after being soaked in a neutral liquid detergent.
- For servicing contact your retailer or authorized dealer.

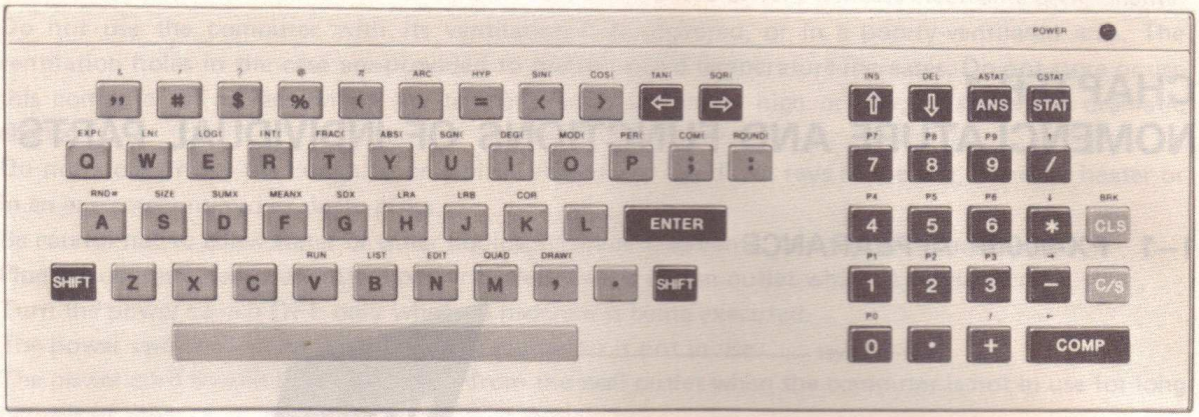
CHAPTER 1
NOMENCLATURE AND FUNCTIONS OF INDIVIDUAL PARTS

1–1 FX-9000P APPEARANCE

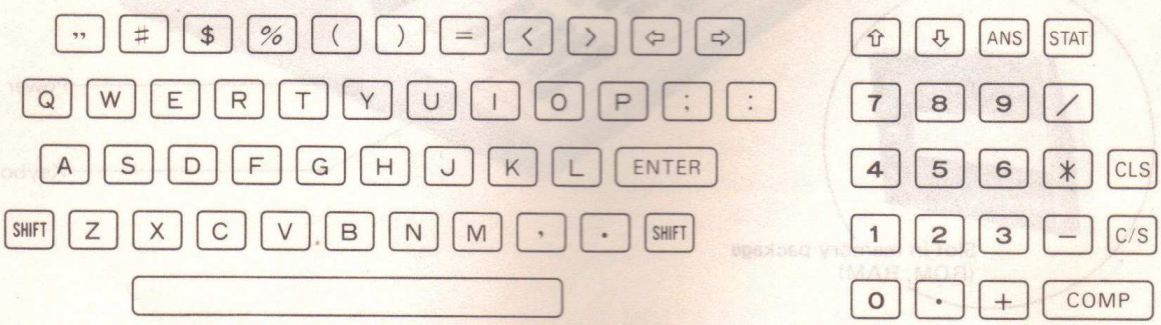




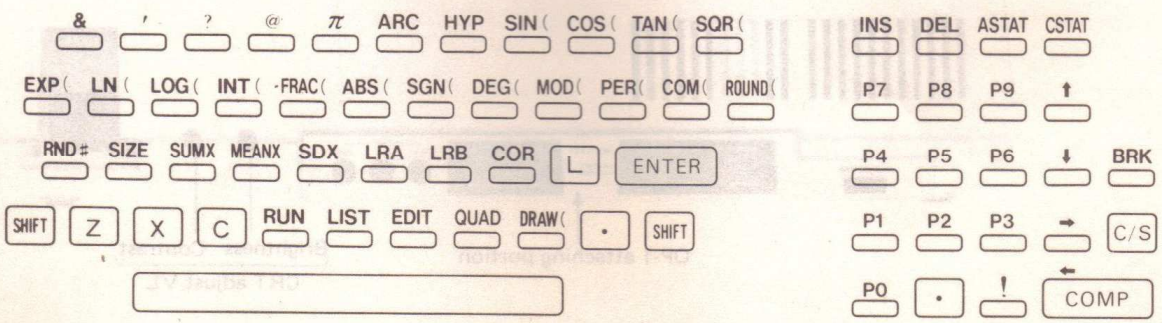
1–2 KEYBOARD LAYOUT



Key functions in the shift-unlock mode
(when SHIFT key is not pressed)



Key functions in the shift-lock mode
(when SHIFT key is pressed)



1-3 NOMENCLATURE

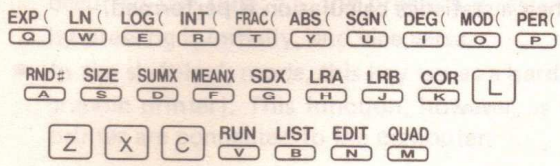
[SHIFT] Shift key

When this key is continually pressed, the keyboard is placed in the shift-lock mode. The one-key commands and symbols written on the keyboard panel can be used in this mode.

★ This key is located at each side of the alphabet keys.

[ENTER] Enter key

- When programs are written, press this key to write to (or store in) the computer memory each program line. Programs are not placed in memory until this key is pressed.
- This key is pressed to execute commands (instructions) manually or to respond to prompts during program execution.
  - ★ These functions are also valid in the shift-lock mode.



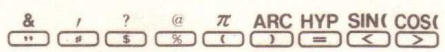
Alphabet keys/One-key command keys

- Alphabetic characters are displayed when these keys are pressed. These keys are used to code CA-BASIC programs or to issue commands.
- When these keys are pressed with the [SHIFT] key depressed, the keyboard is placed in the shift-lock mode. As a result, the one-key commands written on the keyboard panel become usable. (See page 12.)
  - ★ Commands entered via one-key command keys are the same as those spelled out using the alphabet keys. Use either method.

  Example: [R] [U] [N] is equal to [SHIFT] [RUN]

Semicolon key/Colon key/Comma key/Period key/One-key command keys

- Press these keys at the specified position to code programs or to use command abbreviations.
- The one-key commands in the shift-lock mode can be used by pressing [COM(], [ROUND(], and [DRAW(] with the [SHIFT] key depressed.



Character keys/One-key command keys

- Press these keys to use characters (symbols, etc.) written on the keys or the keyboard panel.
- Characters written on the keyboard panel can be used by pressing [&], ['], [?], and [@] in the shift-lock mode.
- In the shift-lock mode, [π], [ARC], [HYP], [SIN(], and [COS(] can issue their respective one-key commands.

TAN(SQR(INS DEL

Cursor move keys/One-key command keys/Insert key/ Delete key

- The blinking cursor which indicates that the computer is ready for input (see page 13), can be moved to the right, to the left, upward, or downward by pressing these keys.
- In the shift-lock mode,
  TAN(and SQR(: One-key commands
  INS : Insert instruction to provide spaces for insertion of characters.
  DEL : Delete instruction by which a character or characters can be deleted.

CSTAT STAT

STAT key/Clear STAT key

- Pressed as a data input key when a statistics calculation is performed.

  Examples: Standard deviation: x STAT
  Regression calculation: x, y STAT

- In the shift-lock mode, pressed as a data delete key when a statistics calculation is performed.

ASTAT ANS

ANS key/ANS STAT key

- Pressed in manual calculation to call back the result of the previous calculation.
- In the shift-lock mode, pressed to display the results of a statistics calculation.
  ★ CSTAT STAT and ASTAT ANS can be used ONLY in manual operation.

P7 P8 P9
P4 P5 P6
P1 P2 P3
P0 •

Numerical value input keys/Program No. keys

- Pressed to input numeric values to the computer.
  • is pressed at the position of the decimal point.
- A program number is specified by pressing any of 0, and 1 to 9 with the SHIFT key depressed. At this time, the corresponding program begins execution if it is already written. (This operation is represented by SHIFT P0 hereafter. This representation applies for every key.) If no program currently resides in the specified program area, one can now be written to and stored in that area.

Arithmetic calculation keys/COMP key/Character keys

- Press [+], [−], [*], and [/] at their respective positions to perform arithmetic calculation. [COMP] is pressed in place of " = " to obtain manual calculation answers.

  Example: $12+34-56\times78\div96=0.5$

  12 [+] 34 [−] 56 [*] 78 [/] 96 [COMP]

- In the shift-lock mode, pressed to use the characters written on the keyboard panel.
- [+] issues an instruction to obtain factorials in the shift-lock mode.

[C/S] CONT stop key/Hard copy key

- Pressed to suspend program execution or restart a program which has been halted by the [C/S] key or STOP statement. When this key is pressed, the resulting function is opposite to the function that occurred last time: i.e. the program is restarted when this key is pressed after the program was halted by pressing it initially, and vice versa.
- In the shift-lock mode, this key issues a hard copy instruction (The whole screen is printed out by the graphic printer). This function, however, is not available unless the option box OP-1 and the graphic printer are connected to the computer.

[BRK / CLS] Clear DISP key/Break key

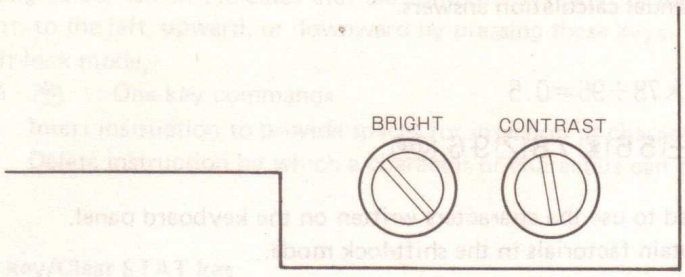
- Pressed to clear the display screen.
  Regardless of the data type on the screen, all the items written on it are erased and the cursor is returned to the home position (the upper left corner on the screen). This key is ineffective during program execution.
- In the shift-lock mode, this key is pressed to abort program execution. Unlike the [C/S] key, the program cannot be restarted from a point in the middle once this key is pressed. For re-execution, the program must be restarted from the beginning.

[ ] Space bar

- Pressed to provide spaces between characters or commands.
  When this key is pressed once, a space corresponding to the length of one character is obtained.

1-4 VIDEO DISPLAY ADJUSTMENT

The Video Display (CRT) is adjusted by using the controlling screws located at the rear of the computer.



"BRIGHT" (the left controlling screw):
Adjusts the screen brightness. When this screw is turned clockwise, the screen becomes brighter. When it is turned counterclockwise, the screen becomes darker and characters become illegible. This screw should be adjusted until the screen can be read but is not too bright.

"CONTRAST" (the right controlling screw):
Adjusts character brightness. Characters become brighter when this screw is turned clockwise or darker when it is turned counterclockwise. Characters cannot be easily read when "CONTRAST" is too bright because they become very thick, or when "CONTRAST" is too dark because they become very narrow. Therefore, adjust this screw until characters can be easily read.

CHAPTER 2
BEFORE STARTING CALCULATION

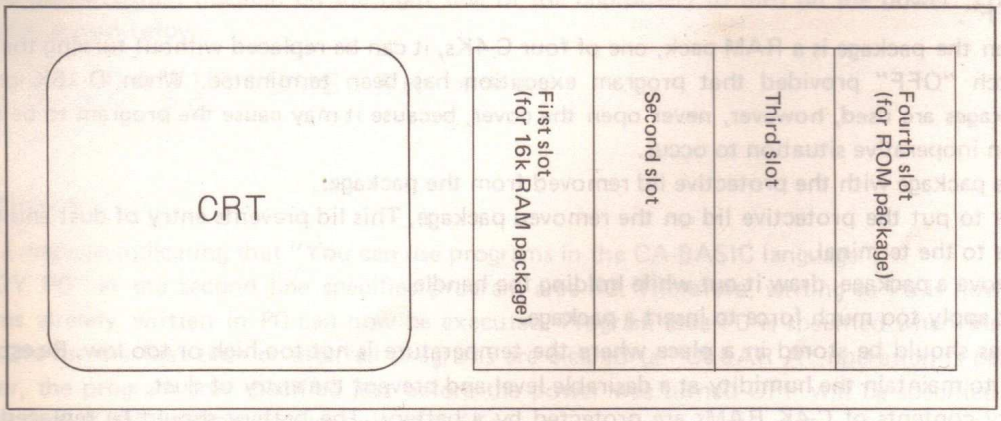
2-1 MEMORY PACKAGE AND OPTION BOX

■ **Insertable memory package**

ROM and RAM are available in insertable packages, which must be placed into the slots provided. The ROM package is inserted into the slot specially provided for it. This package is used as the extended BASIC interpreter or as a fixed program package.

The RAM package is provided to supply RAM as user's area. Two types of RAM packages are available: C-MOS 4k bytes (C-4K) and N-MOS 16k bytes (D-16K). C-4K is provided with a power supply backup, and, therefore, programs and data are protected even when the power switch of the computer is turned OFF or the package is pulled out of the slot. A total of four RAM packages can be inserted into four slots. However, they must be inserted starting from the left slot. If no RAM package is in the first slot (the leftmost slot), RAM packages will not operate. D-16K must be inserted into the first slot.

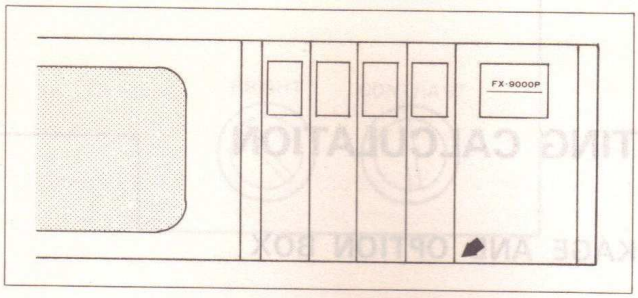
★ RAM packages must be inserted from left to right, and slots must not be skipped.



★ Although the fourth slot is provided for a ROM package (to use a ROM package, it must be inserted into this slot), a RAM package may be inserted into this slot to increase RAM area. The fourth slot can be used for either a RAM or a ROM package.

■ **RAM/ROM Package installation and removal**

- The power switch must be in the OFF position.
- Lightly push the memory pack cover (the position is indicated by an arrow in the figure below) to open the cover.



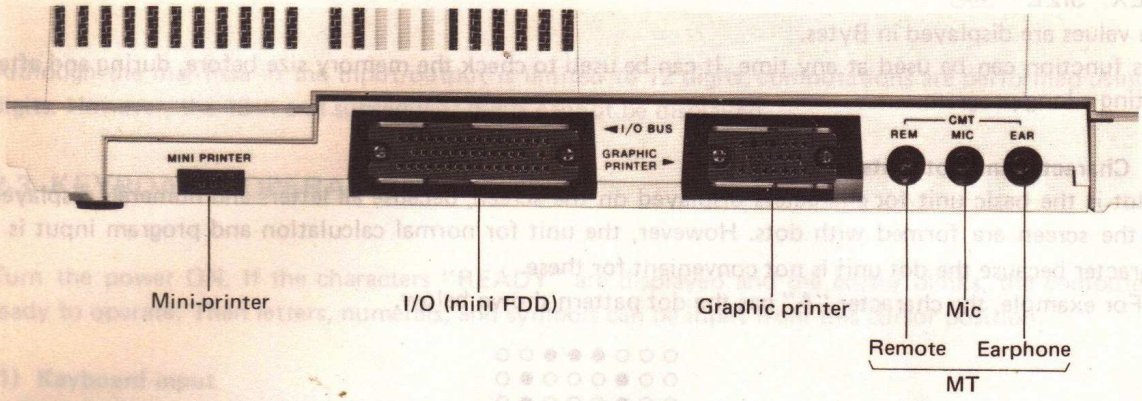
- Hold the handle of the package and draw the package out toward the front.
- Remove the protective lid from the new package. Hold the package handle, and insert the package into the the slot. Push it into the slot fully until a click is heard.
- Pull the cover down gently. Then close it completely by lightly pushing the spot indicated by the arrow.

■ **Caution**

- Never open the memory pack cover during program execution, or a buzzer will sound to indicate an "error".
- Packages can be installed or removed only after programs are terminated and the power switch is set to "OFF".
  - ★ When the package is a RAM pack, one of four C-4Ks, it can be replaced without turning the power switch "OFF" provided that program execution has been terminated. When D-16K or ROM packages are used, however, never open the cover, because it may cause the program to be cleared or an inoperative situation to occur.
- Insert a package with the protective lid removed from the package.
- Be sure to put the protective lid on the removed package. This lid prevents entry of dust and dirt, or damage to the terminal.
- To remove a package, draw it out while holding the handle.
- Do not apply too much force to insert a package.
- Packages should be stored in a place where the temperature is not too high or too low. Be especially careful to maintain the humidity at a desirable level and prevent the entry of dust.
- Memory contents of C-4K RAMs are protected by a battery. The battery should be replaced with a new one every three years, (the life span of the battery).
- If the battery is flat or needs replacement, the contents of memory will be erased. Therefore, important programs and data should be recorded on a suitable medium such as cassette magnetic tape (CMT).

■ **Option box (OP-1)**

Printers and MT cannot be connected directly to the computer. The OP-1 acts as an interface to control them when they are connected.



★ Refer to the OP-1 Instruction Manual for the OP-1 installation and other instructions.

★ For printer graphics, use the MX-82 made by EPSON Co., because it is the most suitable printer for the FX-9000P. For connecting the printer, cables made by EPSON should also be used.

2-2 BASIC OPERATIONS

(1) Turning ON the power

Press the power switch (located on the right side of the computer) to turn on the power. You will see the display shown below.

```
*** BASIC SYSTEM ***
READY P0
:_
```

Blinking

This is a message indicating that "You can use programs in the CA-BASIC language".

"READY P0" in the second line specifies program area P0. Therefore, writing to P0 is now possible. Programs already written in P0 can now be executed. Program area P0 is specified when the power is turned ON for the first time or after all programs are cleared (a "CLEAR A" operation is performed). However, the program area specified just before the power was turned OFF will be specified when the memory pack RAM packages consist only of C-4Ks.

The blinking cursor (—) indicates that the computer is waiting for input. At this time, manual calculation can be performed, and programs and commands can be written.

The screen size is 32 characters (horizontal length) x 16 characters (vertical length). The number of displayed characters, numerals, and symbols will be within this limit (512 characters).

Note: Even if the RAM packages consist only of C-4Ks, power must be turned OFF only AFTER program execution has been terminated. Stored programs may be erased if the power plug is pulled out of the socket or the power is turned OFF during program execution.

(2) Memory size

The "SIZE" function is used to determine the capacity of the currently usable RAM area (the memory size).

EX: SIZE [COMP]

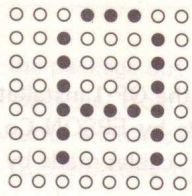
The values are displayed in Bytes.

This function can be used at any time. It can be used to check the memory size before, during and after writing a long program.

(3) Character and dot units on screen

A dot is the basic unit for characters displayed on the screen, because all letters and numerals displayed on the screen are formed with dots. However, the unit for normal calculation and program input is a character because the dot unit is not convenient for these.

For example, the character "A" has the dot pattern shown below.



A character is formed by 8x8 dots. When the dot unit on the screen is used, therefore, it is 256 (horizontal length) x 128 (vertical length) dots. The character unit differs from the dot unit as follows: Character units are used to write numerals, characters, and symbols from the keyboard. Dot units are used to specify graphs and patterns to be drawn using the graphic function.

Unit: One character = 8 x 8 dots

Screen: 32 x 16 characters = 256 x 128 dots (32,768 dots)

- Number of input/output digits and number of calculation digits for this computer

Number of input/output digits

Mantissa : 12

Exponent : 2

Range: $1.0^{-99} \sim +9.99999999999 \times 10^{+99}$

Input values exceeding these limits in data input or manual calculation are handled as follows:

Mantissa having 15 or more digits:

A maximum of 15 digits can be input, but succeeding digits are ignored.

Exponent more than 99:

An error results.

Example: $1.234567890123 \times 100 =$

```
1.234567890123*100
123.4567890
```

Exponential display is automatically used when a calculation result (answer) is 10^{12} or more or below 10^{-2} (0.01).

Example: $123456789012 \times 10 =$

```
123456789012*10
1.23456789012E+12
```

When exponential display is used, the result is displayed together with an exponent sign after the mantissa.

Example: 1.234÷1000=

```
1.234/1000
1.234E-03
```

Although the mantissa in the input/output is limited to 12 digits, computations are performed using 15 digits. However, the 13th and subsequent digits cannot be displayed.

2-3 KEYBOARD OPERATION

Turn the power ON. If the characters "READY" are displayed and the cursor blinks, the computer is ready to operate. Then letters, numerals, and symbols can be input from this cursor position.

(1) Keyboard input

- Alphabetic letter input

Example: Input ABC with keys.

Operation [A][B][C]

```
READY P0
:ABC_
```

Example: Input SIN (with keys.

Operation [S][I][N][(]
or [SHIFT] [SIN(]

```
READY P0
:SIN(_
```

★ One-key commands are provided to allow the user to enter a command simply by operating one command key. However, commands can also be entered by completely spelling out the command using the alphabet keys.

- Numeral input

Example: Input 123.

Operation: [1][2][3]

```
READY P0
:123_
```

Example: Input 96.3

Operation: [9][6][.][3]

```
READY P0
:96.3_
```

- Symbol input

Example: Input $ @ ? ".

Operation: [$] [SHIFT] [@ $] [SHIFT] [? %] ["]

```
READY P0
:$@?"_
```

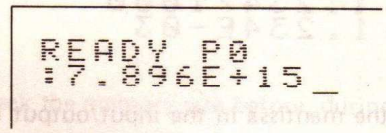
- Input of value with exponent

Example: Input 7.896×10^{15}.

Operation: [7] [.] [8] [9] [6] [E] [+] [1] [5]

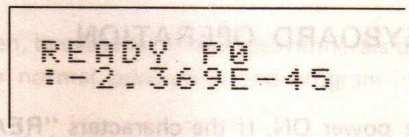
(Omissible)

(Exponent input)



Example: Input -2.369×10^{-45}.

Operation: [−] [2] [.] [3] [6] [9] [E] [−] [4] [5]

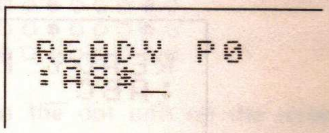


(2) Input change (correction, deletion, and insertion)

- Correction procedure

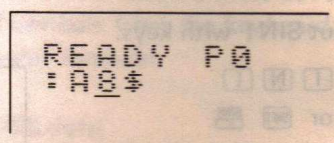
1) Move the cursor to the position at which a correction is to be made. (Use [⇦], [⇨], [⇧] and [⇩])
2) With the cursor at this position, press the key of the desired letter, numeral, or symbol.

Example: Correct "A8$" to "A9$"

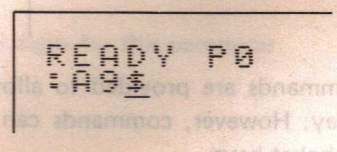


Operation: Move the cursor to the left by two characters.

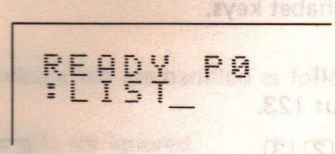
[⇦] [⇦]



Press the [9] key.

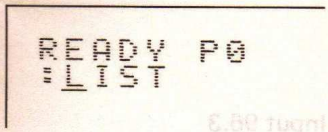


Example: Correct "LIST" to "EDIT".

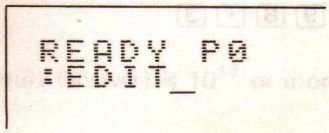


Operation: Move the cursor to the left by four characters.

[⇦] [⇦] [⇦] [⇦]



Press [E] [D] [I] [T] or [SHIFT] [EDIT] .



- Deletion procedure

1) Move the cursor to the position at which a deletion is to be made.
2) With the cursor at this position, press [SHIFT] [DEL] .

Each time this operation is performed, one character is deleted and the trailing characters are automatically moved up.

Example: Delete one "I" from "SIIN ("
(One character delection)

```
READY P0
:SIIN(_
```

Operation: Move the cursor to the left by three characters.
◁ ◁ ◁

```
READY P0
:SIIN(
```

Press SHIFT DEL .

```
READY P0
:SIN(
```

Example: Delete "X," from "INPUT X, Y".

```
READY P0
:INPUT X,Y_
```

Operation: Move the cursor to the left by three characters.
◁ ◁ ◁

```
READY P0
:INPUT X,Y
```

Press SHIFT DEL SHIFT DEL .
(The same operation is repeated.)
Two characters are deleted.

```
READY P0
:INPUT Y
```

Example: Delete ",B" from "PRINT A, B".

```
READY P0
:PRINT A,B_
```

Operation: Move the cursor to the left by two characters.
◁ ◁

```
READY P0
:PRINT A,B
```

Press the space key twice.

```
READY P0
:PRINT A  _
```

This deleting technique is effective when the deleted characters are followed by no other character. (Otherwise, spaces are placed before the following characters.)
The same result can be obtained by Pressing SHIFT DEL SHIFT DEL

- Insertion of characters

1) Move the cursor to the character which follows the position at which an insertion is to be made.
2) With the cursor at this position, press [SHIFT] [INS] .
   Each time this is done, a space corresponding to one character is created. With the cursor at the created space, press the key for the letter, numeral, or symbol to be inserted.

Example: Correct "T$" to "T1$"

```
READY P0
:T$_
```

Operation: Move the cursor to the left by one character.
[←]

```
READY P0
:T$
```

Press [SHIFT] [INS] .

```
READY P0
:T_$
```

Press the key for the character to be inserted ([1]).

```
READY P0
:T1$
```

Example: Correct "PRINT X" to "PRINT SIN(X)".

```
READY P0
:PRINT X_
```

Operation: Write a close parenthesis ([)]).

```
READY P0
:PRINT X)_
```

Move the cursor to the left by two characters.
[←] [←]

```
READY P0
:PRINT X)
```

Create a space corresponding to a four-character length to insert "SIN(".
[SHIFT] [INS] [SHIFT] [INS] [SHIFT] [INS] [SHIFT] [INS]

```
READY P0
:PRINT _    X)
```

Write the characters to be inserted by pressing [S][I][N][(] or [SHIFT] [SIN( ] .

```
READY P0
:PRINT SIN(X)
```

As described above, input data can be freely changed by using [SHIFT] [INS] or [SHIFT] [DEL] while watching the screen.

CHAPTER 3
MANUAL CALCULATION

3-1 INTRODUCTION

Manual calculation is performed from the command mode and does not allow the storage of the results in memory.

Examples of manual calculation:

- Substitute the right side of an equation for the left side.
- Perform a calculation operation "by hand".
- Call the value of a variable.

3-2 CALCULATION METHOD

Manual calculation can be performed in the same manner as on an ordinary small electronic calculator. Four fundamental arithmetic operations are performed by using the calculation keys and the numeric input keys (located on the right side of the keyboard). Functional operations are performed by using the one-key command keys and the alphabet keys (located on the left side of the keyboard).
This computer performs these operations as specified by the calculation expression. That is, the computer recognizes the calculation priority itself, and performs the operation accordingly — using true algebraic logic.

The calculation hierarchy is as follows:

① Function (sin, cos, tan, etc.)
② Power, factorial
③ x, ÷ (*, /)
④ +, –

When two or more items in a mathematic expression have the same priority, calculation is performed from the first one (from the left).
If parentheses are used, the expression inside them has the highest priority.

■ Parenthesis calculation and number of levels

A maximum of 17 pair of parentheses can be used for calculation. When parentheses are used, the maximum allowed number of calculation levels is 12.

The number of levels is counted as follows:

Example: 2 x (3 + 4 x (5 + 6) ÷ 7) =

This calculation is performed by pressing the following keys:

2 [×] [(] 3 [+] 4 [×] [(] 5 [+] 6 [)] [/] 7 [)] [COMP]

1 level, 1 level, 1 level, 1 level

[COMP] ↑ Functions as an equal (=)

Number of parenthesis pair: 2
Number of levels: 4

Although a maximum of 17 pair of parentheses can be used for each calculation, the allowed number of levels is 12 because they are combined with the number of levels to leave low-priority calculation for recognizing priority of function, multiplication, and division.

■ **Operational signs**

Signs for the four fundamental arithmetic operations used by BASIC differ slightly from those used conventionally.

Addition and subtraction:

"+" and "–" used by both BASIC and general calculation expressions.

Multiplication and division:

"x" and "÷" used by general calculation expressions.

"✱" and " / " used by BASIC.

Therefore, a calculation expression

$2+3-4\times5\div6$

is written

2 + 3 – 4 ✱ 5 / 6

when the BASIC operational signs are used.

The built-in functions used by this computer are:

Function name		Format	
Trigonometric function	$\sin x$	SIN(x)	[SHIFT SIN( ]
	$\cos x$	COS(x)	[SHIFT COS( ]
	$\tan x$	TAN(x)	[SHIFT TAN( ]
Inverse trigonometric function	$\sin^{-1} x$	ASN(x)	[SHIFT ARC SHIFT SIN( ]
	$\cos^{-1} x$	ACS(x)	[SHIFT ARC SHIFT COS( ]
	$\tan^{-1} x$	ATN(x)	[SHIFT ARC SHIFT TAN( ]
Hyperbolic function	$\sinh x$	HSN(x)	[SHIFT HYP SHIFT SIN( ]
	$\cosh x$	HCS(x)	[SHIFT HYP SHIFT COS( ]
	$\tanh x$	HTN(x)	[SHIFT HYP SHIFT TAN( ]
Inverse hyperbolic function	$\sinh^{-1} x$	AHS(x)	[SHIFT ARC SHIFT HYP SHIFT SIN( ]
	$\cosh^{-1} x$	AHC(x)	[SHIFT ARC SHIFT HYP SHIFT COS( ]
	$\tanh^{-1} x$	AHT(x)	[SHIFT ARC SHIFT HYP SHIFT TAN( ]
Square root	$\sqrt{x}$	SQR(x)	[SHIFT SQR( ]
Exponential function	e^x	EXP(x)	[SHIFT EXP( ]
Natural logarithm	$\ln x$	LN (x)	[SHIFT LN( ]
Common logarithm	$\log x$	LOG(x)	[SHIFT LOG( ]
Factorial	$x!$	x !	[SHIFT !]
Conversion into integer	INT x	INT(x)	[SHIFT INT( ]

Removal of the integer	FRAC x	FRAC(x)	[SHIFT FRAC]
Conversion into absolute value	$\|x\|$	ABS(x)	[SHIFT ABS]
Positive/negative sign exchange	Positive number → 1 0 → 0 Negative number → -1	SGN(x)	[SHIFT SGN]
Degree, minute, second		DEG(x)	[SHIFT DEG]
	(sexagesimal number → decimal number)		
Remainder calculation		MOD(x, y)	[SHIFT MOD]
Permutation	${}_xP_y$	PER(x, y)	[SHIFT PER]
Combination	${}_xC_y$	COM(x, y)	[SHIFT COM]
Rounding		ROUND(x, y)	[SHIFT ROUND]
	The y-th significant digit of x is counted as a unit when it is 5 or over, or disregarded when it is below 5.		
Memory size		SIZE	[SHIFT SIZE]
Random number		RND#	[SHIFT RND#]

Statistics calculation

Number of data items n		CNT	——
Standard deviation of x	${}_x\sigma_{n-1}$	SDX	[SHIFT SDX]
Standard deviation of y	${}_y\sigma_{n-1}$	SDY	——
Average of x	$\bar{x}$	MEANX	[SHIFT MEANX]
Average of y	$\bar{y}$	MEANY	——
Sum of x	Σx	SUMX	[SHIFT SUMX]
Sum of y	Σy	SUMY	——
Square sum of x	Σx^2	SUMX2	——
Square sum of y	Σy^2	SUMY2	——
Data product sum	Σxy	SUMXY	——
Constant term	A	LRA	[SHIFT LRA]
Regression coefficient	B	LRB	[SHIFT LRB]
Correlation coefficient	r	COR	[SHIFT COR]

These built-in functions can be used in both manual and program calculations.

- Displaying the results of a previous calculation

Results of a manual calculation are stored in the computer until the next manual calculation is performed. The results can be displayed by pressing the [ANS] key.

Example: 741 + 852 = 1593

1593 − 963 = 630

Operation: [7][4][1][+][8][5][2][COMP]

[ANS][−][9][6][3][COMP]

```
READY P0
:741+852
 1593
:(_1593)-963
 630
:
```

■ **Manual calculation rules**

- An error message will be displayed if a mathematic expression or assignment statement violates the rules of CA-BASIC. (For errors, see page 49 and the table of error messages.)
- If an operation exceeds the range of ±9.99999999999E+99, an overflow will occur and an error message will be displayed. If an operation is below the limit of 1.0E−99, an underflow occurs and the operation result is 0.
- Specify angle units by using the SET command before performing trigonometric functions or similar calculations. The specified angle unit can be changed only by specifying another angle unit.
  When using the D-16K, the angle unit is specified in "DEG" (Degree) when the power switch is turned ON. However, this angle unit is not changed even if the power is turned OFF when RAM packages consist only of C-4Ks unless a new angle unit is specified.

3-3 MANUAL CALCULATION OPERATION

Manual calculations are performed in the same manner as on ordinary electronic calculators. However, FX-9000P assures easier viewing and simpler correction since the calculation process is displayed on the screen.

■ The four fundamental arithmetic operations are used to perform mathematic expressions.
The keys used are [+] · [−] · [*] (×) · [/] (÷) and [COMP] (=). The [COMP] key, which functions as an equal sign "=", is used to obtain the calculation result.

Example 12+36−9×5÷4=36.75

Operation [1][2][+][3][6][−][9][*][5][/][4][COMP]

```
:12+36-9*5/4
 36.75
```

■ Function calculation can contain four fundamental arithmetic operations. Input a function command followed by data in parenthesis "( )".

Example log 1.23=0.0899051114394

Operation LOG[(]1·23[)][COMP]

```
:LOG(1.23)
 0.0899051114394
```

★ In manual calculation, the last ") " can be omitted. That is,

LOG[(]1·23[)][COMP] is the same as LOG[(]1·23[COMP]

★ In this manual, the alphabet and numerical keys are symbolized as follows:

Example: [L][O][G][(][1][.][2][3][)][COMP] → LOG(1·23) COMP

■ Variables are used to store values and calculation results. Alphabetic letters (A, B, ..., Z) or combinations of alphabetic letters and numerals (A1, A2, ..., A9, ...) are used for variables. To place values or calculation results in a variable, an assignment equation is entered manually.

Example: Store 1234 in variable A.

Operation: A [=] 1234 [ENTER]

```
:A=1234
:
```

Example: Add the answer of 23 x 56 to variable K1.

Operation: K 1 [=] K1 [+] 23 [*] 56 [ENTER]

```
:K1=K1+23*56
:
```

This command (a manual command using the [ENTER] key in place of the [COMP] key) allows the user to manually perform the same operation as an assignment statement in a CA-BASIC program.

- Correction before pressing the [COMP] or [ENTER] key:
  move the cursor to the position at which correction is to be made, and press the correct key. (See 2-2.)
- Press the [CLS] key to erase all displayed items from the screen.

3-4 MANUAL CALCULATION EXAMPLES

(1) Basic calculation

■ Four fundamental arithmetic operations

Example $23+4.5-53=-25.5$

Operation 23 [+] 4·5 [−] 53 [COMP]

```
:23+4.5-53
-25.5
:
```

Example $56\times(-12)\div(-2.5)=268.8$

Operation 56 [*] [(] [−] 12 [)] [/] [(] [−] 2·5 [)] [COMP]

```
:56*(-12)/(-2.5)
 268.8
:
```

Example $12369\times7532\times74103=6.90368061272\times10^{12}$

$(=6903680612720)$

Operation 12369 [*] 7532 [*] 74103 [COMP]

```
:12369*7532*74103
6.903680612 72E+12
```

Example $1.23\div90\div45.6=2.99707602339\times10^{-4}$

$(=0.000299707602339)$

Operation 1·23 [/] 90 [/] 45·6 [COMP]

```
:1.23/90/45.6
2.997076023 39E-04
```

※ Exponential display is used when the calculation result is equal to 10^{12} or more or is below 10^{-2} (0.01).

Example $7\times8+4\times5=76$

Operation 7 ✱ 8 ＋ 4 ✱ 5 COMP

```
:7*8+4*5
 76
```

Example $12+(2.4\times10^{5})\div42.6-78\times36.9=2767.6028169$

Operation 12 ＋ 2·4 E 5 ／ 42·6 － 78 ✱ 36·9 COMP

```
:12+2.4E5/42.6-78*36.9
 2767.6028169
```

■ Memory calculation

Example $12\times45=540$

$12\times31=372$

$75\div12=6.25$

Operation A = 12 ENTER

A ✱ 45 COMP

A ✱ 31 COMP

75 ／ A COMP

```
:A=12

:A*45
 540
:A*31
 372
:75/A
 6.25
```

Example $23+9=32$

$53-6=47$

$-)\ 45\times2=90$

$99\div3=33$

Total 22

Operation M = 23 ＋ 9 ENTER

M = M ＋ 53 － 6 ENTER

M = M － 45 ✱ 2 ENTER

M = M ＋ 99 ／ 3 ENTER

M COMP

```
:M=23+9

:M=M+53-6

:M=M-45*2

:M=M+99/3

:M
 22
```

※ This latter operation does not allow you to view the individual calculation results. Use the following method to view these results.

23 ＋ 9 COMP

M = ANS ENTER

53 － 6 COMP

M = M ＋ ANS ENTER

45 ✱ 2 COMP

M = M － ANS ENTER

99 ／ 3 COMP

M = M ＋ ANS ENTER

M COMP

```
:23+9
 32
:M=(ANS)
 32
:53-6
 47
:M=M+(ANS)
 47
:45*2
 90
:M=M-(ANS)
 90
:99/3
 33
:M=M+(ANS)
 33
:M
 22
```

(2) Function calculation

■ Trigonometric function (sin, cos, tan) and inverse trigonometric function ($\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$)

● An angle unit must be specified by using the SET command when trigonometric or inverse trigonometric functions are used. This angle unit specification only has to be set once.

Example 14°25'36" = 14.4266666666°

Operation DEG(14·2536) COMP

```
:DEG(14.2536)
 14.4266666666
```

※ "DEG(" can be entered by using the one-key command or by spelling D E G (with the alphabet keys. (This applies for all the functions described below.)

Example sin 12.3456° = 0.213807920122

Operation SET D ENTER

SIN(12·3456) COMP

(Or, SHIFT SIN(12·3456) .)

```
:SET D
:SIN(12.3456)
 0.213807920122
```

Example $2 \cdot \sin 45° \times \cos 65°6' = 0.59543455751$

Operation 2 ✱ SIN(45) ✱ COS(DEG(65·06)) COMP

```
:2*SIN(45)*COS(DEG(65.06))
 0.59543455751
```

Example $\sin^{-1} 0.5 = 30°$ (Obtain x when $\sin x° = 0.5$)

Operation ASN(0·5) COMP

(Or, SHIFT ARC SHIFT SIN(.)

```
:ASN(0.5)
 30
```

Example $2.5 \times (\sin^{-1} 0.8 - \cos^{-1} 0.9) = 68.2204239775$

Operation 2·5 ✱ (ASN(0·8) − ACS(0·9)) COMP

```
:2.5*(ASN(0.8)-ACS(0.9))
 68.2204239775
```

Example $\cos(\frac{\pi}{3}\text{ rad}) = 0.5$

Operation SET R ENTER

COS(SHIFT π ÷ 3) COMP

```
:SET R
:COS(π/3)
 0.5
```

Example $\cos^{-1}\frac{\sqrt{2}}{2} = 0.7853981634$

Operation ACS(SQR(2) ÷ 2) COMP

```
:ACS(SQR(2)/2)
 0.7853981634
```

Example $\tan(-35\,\text{gra}) = -0.61280078814$

Operation SET G [ENTER]

TAN [(] [−] 35 [)] [COMP]

```
:SET G
:TAN(-35)
-0.61280078814
```

■ Logarithmic function (log, ln) and exponential function (e^x, x^y)

Example $\log 1.23 (= \log_{10} 1.23) = 0.0899051114394$

Operation LOG [(] 1·23 [)] [COMP]

```
:LOG(1.23)
 0.0899051114394
```

Example $\ln 90 (= \log_e 90) = 4.49980967033$

Operation LN [(] 90 [)] [COMP]

```
:LN(90)
 4.49980967033
```

Example $\log 456 \div \ln 456 = 0.434294481902$

Operation LOG [(] 456 [)] [÷] LN [(] 456 [)] [COMP]

```
:LOG(456)/LN(456)
 0.434294481902
```

Example $e^{4.5} = 90.0171313005$

(Find the antilogarithm of natural logarithm 4.5.)

Operation EXP [(] 4·5 [)] [COMP]

```
:EXP(4.5)
 90.0171313005
```

Example $10^{1.23} = 16.9824365246$

(Find the antilogarithm of common logarithm 1.23.)

Operation 10 [SHIFT] [log] 1·23 [COMP]

```
:10↑1.23
 16.9824365246
```

Example $5.6^{2.3} = 52.581438372$

Operation 5·6 [SHIFT] [log] 2·3 [COMP]

```
:5.6↑2.3
 52.581438372
```

Example $123^{\frac{1}{7}} (= \sqrt[7]{123}) = 1.98864779528$

Operation 123 [SHIFT] [log] [(] 1 [÷] 7 [)] [COMP]

```
:123↑(1/7)
 1.98864779528
```

Example $(78-23)^{-12} = 1.30511182934 \times 10^{-21}$

Operation [(] 78 [−] 23 [)] [SHIFT] [log] [(] [−] 12 [)] [COMP]

```
:(78-23)↑(-12)
 1.30511182934E-21
```

Example $2^2+3^3+4^4=287$

Operation 2 SHIFT $\sqrt{\ }$ 2 + 3 SHIFT $\sqrt{\ }$ 3 +
4 SHIFT $\sqrt{\ }$ 4 COMP

```
: 2↑2+3↑3+4↑4
  287
```

Example $\log \sin 40^\circ + \log \cos 35^\circ = -0.278567983822$

The antilogarithm is equal to 0.526540784519.
(Logarithmic calculation for sin 40° x cos 35°)

Operation SET D ENTER (DEG specified)

LOG(SIN(40)) + LOG(COS(35)) COMP

10 SHIFT $\sqrt{\ }$ ANS COMP

```
: SET D
: LOG(SIN(40))+LOG(COS(35))
-0.278567983822
: 10↑(-0.278567983822)
  0.526540784519
```

■ **Hyperbolic function (sinh, cosh, tanh) and inverse hyperbolic function ($\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$)**

Example $\sinh 3.6 = 18.2854553606$

Operation HSN(3·6) COMP

(Or, SHIFT HYP SHIFT SIN(.)

```
: HSN(3.6)
  18.2854553606
```

Example $\tanh 2.5 = 0.986614298151$

Operation HTN(2·5) COMP

```
: HTN(2.5)
  0.986614298151
```

Example $\cosh 1.5 - \sinh 1.5 = 0.22313016015$

Operation HCS(1·5) − HSN(1·5) COMP

```
: HCS(1.5)-HSN(1.5)
  0.22313016015
```

Example $\sinh^{-1} 30 = 4.09462222433$

Operation AHS(30) COMP

(Or, SHIFT ARC SHIFT HYP SHIFT SIN(.)

```
: AHS(30)
  4.09462222433
```

Example Obtain x for $\tanh 4x = 0.88$.

$$x = \frac{\tanh^{-1} 0.88}{4} = 0.34394191413$$

Operation AHT(0·88) ÷ 4 COMP

```
: AHT(0.88)/4
  0.34394191413
```

■ Permutation ($_nP_r$) and combination ($_nC_r$)

- Total number of permutations

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

- Total number of combinations

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

Example: Four items are taken out of ten and arranged. In how many ways can they be arranged? ……$_{10}P_4=5040$

Operation: PER[(] 10, 4 [)] [COMP]

```
:PER(10,4)
 5040
```

Example: How many four-digit even numbers can be created by using numerals 1 to 7 (Each four-digit number must not contain the same numerals)? [$\frac{3}{7}$ of the whole are even numbers] ……$_7P_4\times\frac{3}{7}=360$

Operation: PER[(] 7, 4 [)] [×] 3 [÷] 7 [COMP]

```
:PER(7,4)*3/7
 360
```

Example: Four items are taken out of ten. In how many ways can they be taken out?……$_{10}C_4=210$

Operation: COM[(] 10, 4 [)] [COMP]

```
:COM(10,4)
 210
```

Example: Five officials (at least one of them must be a woman) are selected from a class consisting of 15 men and 10 women. In how many ways can they be selected?

[at least one = (the whole) − (not contained)]

……$_{25}C_5-{}_{15}C_5=50127$

Operation: COM[(] 25, 5 [)] [−] COM[(] 15, 5 [)] [COMP]

```
:COM(25,5)-COM(15,5)
 50127
```

■ Other functions ($\sqrt{\ }$, $x!$, MOD, SGN, RND #, ROUND, ABS, INT, FRAC, SIZE)

Example: $\sqrt{2}+\sqrt{5}=3.65028153987$

Operation: SQR[(] 2 [)] [+] SQR[(] 5 [)] [COMP]

```
:SQR(2)+SQR(5)
 3.650281539
```

Example: $8!\ (=1\times2\times3\times\cdots\cdots\times7\times8)=40320$

Operation: 8 [SHIFT] [÷] [COMP]

```
:8!
 40320
```

Example: Obtain the remainder of 1356 ÷ 7 = 193 . . . 5

Operation: MOD[(] 1356, 7 [)] [COMP]

```
:MOD(1356,7)
 5
```

Example: Give "1" to a positive number, "-1" to a negative number, and "0" to a zero

Operation: SGN [(] 6 [)] [COMP]

SGN [(] 0 [)] [COMP]

SGN [(] [−] 2 [)] [COMP]

```
:SGN(6)
 1
:SGN(0)
 0
:SGN(-21)
-1
```

Example: Random number generation
(random number ($0 < RND\ \# < 1$))

Operation: RND [#] [COMP]

```
:RND#
 0.325017728704
```

Example: Put the answer of 12.3 x 4.56 into a round number consisting of three significant digits.
12.3 x 4.56 = 56.088

Operation: ROUND [(] 12·3 [*] 4·56, 4 [)] [COMP]

※ When three significant digits are to be obtained, specify as follows: Count the fourth digit as a unit if it is 5 or more. Otherwise, disregard it.

```
:ROUND(12.3*4.56,4)
 56.1
```

Example: What is the integer of $\frac{7800}{96}$? . . . 81

Operation: INT [(] 7800 [/] 96 [)] [COMP]

※ This command obtains the greatest integer that does not exceed the original value.

```
:INT(7800/96)
 81
```

Example: What is the decimal of $\frac{7800}{96}$? . . . 0.25

Operation: FRAC [(] 7800 [/] 96 [)] [COMP]

```
:FRAC(7800/96)
 0.25
```

Example: Obtain the size of the remaining memory (Bytes)

Operation: SIZE [COMP]

```
:SIZE
 1011
```

(3) Statistics calculation

- Clear the statistics memory by using the "SAC" command before performing statistics calculations.

■ **Standard deviation calculation**

- The [STAT] key is used. To input individual data items, enter,
  Data [STAT]

- To input several identical data items, press the [STAT] key consecutively as many times as the number of these data items, or enter,
  data; quantity [STAT]

- Standard deviation

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

Estimates the standard deviation of a population by using sample data from it.

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

Example: Data items are 55, 54, 51, 55, 53, 53, 54, 52.

Operation: SAC [ENTER]

55 [STAT] 54 [STAT] 51 [STAT] 55 [STAT]

53 [STAT] [STAT] 54 [STAT] 52 [STAT]

(Standard deviation σ_{n-1})	SDX	[COMP]
(Mean $\bar{x}$)	MEANX	[COMP]
(Number of data n)	CNT	[COMP]
(Sum Σx)	SUMX	[COMP]
(Square sum Σx^2)	SUMX2	[COMP]

```
: SAC
: 55
: 54
: 51
: 55
: 53
: 53
: 54
: 52
: SDX
1.407885953 17
: MEANX
53.375
: CNT
8
: SUMX
427
: SUMX2
22805
```

Example: Obtain $\bar{x}$ and σ_{n-1} from the following table.

Class No.	Value corresponding to the center of a class	Frequency
1	110	10
2	130	31
3	150	24
4	170	2
5	190	3

Operation: SAC ENTER

110; 10 STAT

130; 31 STAT

150; 24 STAT

170 STAT STAT

190 STAT STAT STAT

SHIFT ASTAT AMS

```
: SAC
:  110; 10
:  130; 31
:  150; 24
:  170
:  190
:  190
:
CNT          70
SUMX         9640
SUMY         0
SUMX2        1351000
SUMY2        0
SUMXY        0
MEAN X       137.714285714
MEAN Y       0
SDX          18.4289806878
SDY          0
RA           0
RB           0
COR          ***
:
```

★ Deletion and correction of incorrect input data (51 STAT is the correct operation):

① 50 STAT → immediately press SHIFT CSTAT STAT. Then input correct data.

② 49 STAT (several items before) → 49 SHIFT CSTAT STAT . Then input correct data.

Incorrect input data with frequency can be deleted and corrected in the same way.

49; 12 STAT (several items before) → 49; 12 SHIFT CSTAT STAT . Then input correct data.

```
: 50
:  50  ***

: 49  ***

: 49: 12  ***
```

■ **Regression calculation**

- To input data, press x data, y data STAT .
- When there are several identical data pair, press the STAT key consecutively as many times as the number of the data pair, or enter x data, y data; quantity STAT.

■ Linear regression calculation

- Regression equation : $y = A + Bx$
  Coefficients A and B are calculated by using the following formulas.

Regression coefficient of regression equation

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

Constant term of regression equation

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

- The correlation coefficient r of the input data pair is calculated by using the following formula.

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

Example: From these measurements, obtain the quantity, sum, square sum, product sum, mean, and standard deviation of x and y, the constant term, the coefficient, and the correlation coefficient.

- Steel bar

Temperature	Measured length
10℃	1003mm
15	1005
20	1010
25	1008
30	1014

Operation: SAC [ENTER]

10, 1003 [STAT]

15, 1005 [STAT]

20, 1010 [STAT]

25, 1008 [STAT]

30, 1014 [STAT]

[SHIFT] [ASTAT (ANS)]

```
:SAC

:10,1003
:15,1005
:20,1010
:25,1008
:30,1014
:
CNT          5
SUMX         100
SUMY         5040
SUMX^2       2250
SUMY^2       5080394
SUMXY        100925
MEANX        20
MEANY        1008
SDX          7.905694150422
SDY          4.301162633521
LRA          998
LRB          0.5
COR          0.9190182776617
:
```

★ Deletion and correction of incorrect input data (10,1003 [STAT] is the correct operation)

① 11,1003 [STAT] → immediately [SHIFT] [CSTAT] .
Then input correct data.

② 10,1030 [STAT] → immediately [SHIFT] [CSTAT] .
Then input correct data.

③ 11,1003 [STAT] (several items before)
→ 11,1003 [SHIFT] [CSTAT] . Then input correct data.
Incorrectly input data with frequency can be deleted and corrected in the same way.
11,1003; 10 [STAT] (several items before) → 11,1003; 10 [SHIFT] [CSTAT] . Then input correct data.

```
:11,1003
:  11,  1003  ***

:11,1003  ***

:11,1003;10  ***
```

■ **Logarithmic regression calculation**

- Regression equation : $y = A + B \cdot \ln x$
  Data x: Input the logarithm $(\ln)$ of x.
  Data y: Input in the same way as for linear regression.
- The regression coefficient can be obtained and corrected by performing the same operation as in linear regression.
  $\Sigma \ln x$, $\Sigma (\ln x)^2$, and $\Sigma \ln x \cdot y$ are obtained for Σx, Σx^2, and Σxy, respectively.

Example:

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

Make logarithmic regression of this data, obtain the correlation coefficient, coefficient, and constant term of the regression equation, and calculate the determination coefficient (r^2).

Operation: SAC [ENTER]

LN(29), 1·6 [STAT]

LN(50), 23·5 [STAT]

LN(74), 38·0 [STAT]

LN(103), 46·4 [STAT]

LN(118), 48·9 [STAT]

(Constant term A of the regression equation) LRA [COMP]

(Regression coefficient B) LRB [COMP]

(Correlation coefficient r) COR [COMP]

(Determination coefficient r^2) COR [SHIFT] [↑] 2 [COMP]

```
:SAC
:LN(29),1.6
:LN(50),23.5
:LN(74),38.0
:LN(103),46.4
:LN(118),48.9
:LRA
 111.1283976 47
:LRB
 34.02014750 16
:COR
 0.994013946616
:COR↑2
 0.988063726067
```

■ Exponential regression calculation

- Regression equation : $y = A \cdot e^{B \cdot x} (\ln y = \ln A + B \cdot x)$
  Data y : Input the logarithm (ln) of y.
  Data x : Input in the same way as for linear regression.
- Correction can be made by performing the same operation as in linear regression. $\ln A$, $\Sigma \ln y$, $\Sigma (\ln y)^2$, and $\Sigma x \cdot \ln y$ are obtained for the constant term A, sum SUM Y, square sum SUM Y2, and product sum SUM XY, respectively.

Example:

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

Make exponential regression of this data, and obtain the regression equation and correlation coefficient.

Operation: SAC [ENTER]

6·9, LN(21·4) [STAT]

12·9, LN(15·7) [STAT]

19·8, LN(12·1) [STAT]

26·7, LN(8·5) [STAT]

35·1, LN(5·2) [STAT]

(Constant term A) EXP(LRA) [COMP]

(Coefficient B) LRB [COMP]

(Correlation coefficient r) COR [COMP]

```
:SAC
:6.9,LN(21.4)
:12.9,LN(15.7)
:19.8,LN(12.1)
:26.7,LN(8.5)
:35.1,LN(5.2)
:EXP(LRA)
 30.49758742 58
:LRB
-0.04920370830 75
:COR
-0.99724735 1987
```

■ Power regression calculation

- The regression equation is $y = A \cdot x^B (\ln y = \ln A + B \ln x)$, and input the logarithm (ln) for both data x and y.
- Correction can be made by performing the same operation as in linear regression. $\ln A$, $\Sigma \ln x$, $\Sigma (\ln x)^2$, $\Sigma \ln y$, $\Sigma (\ln y)^2$, and $\Sigma (\ln x \cdot \ln y)$ are obtained for the constant term A, Σx, Σx^2, Σy, Σy^2 and Σxy, respectively.

Example:

x_i	y_i
28	2410
30	3033
33	3895
35	4491
38	5717

Make power regression of this data, and obtain the regression equation and correlation coefficient.

Operation: SAC ENTER

LN (28), LN (2410) STAT

LN (30), LN (3033) STAT

LN (33), LN (3895) STAT

LN (35), LN (4491) STAT

LN (38), LN (5717) STAT

(Constant term A) EXP (LRA) COMP

(Coefficient B) LRB COMP

(Correlation coefficient r) COR COMP

```
:SAC
:LN(28),LN(2410)
:LN(30),LN(3033)
:LN(33),LN(3895)
:LN(35),LN(4491)
:LN(38),LN(5717)
:EXP(LRA)
 0.238801068547
:LRB
 2.771866157 61
:COR
 0.998906255118
```

3-5 GRAPHIC FUNCTION

The graphic function is a special feature of this computer. It can be used as a part of a program. It can also be used manually: graphs and patterns can be drawn just by giving simple coordinate specifications. The graphic function can be used manually in the same way it is used in programs. That is, the user is required only to specify coordinates directly from the keyboard. Commands used for the graphic function are outlined below.

- INIT statement: INIT$(x_1, y_1), x_2, y_2$
  This statement specifies the center point of coordinate axes and the step of the coordinates (the width of each division on the axis). The following items are specified at the same time.
  - ○ The point at which the x and y axes intersect. (Viewed from the lower left corner on the screen).
  - ○ The number of dots to be counted as a unit for x and y coordinates.

  (x_1, y_1) indicates the center of the coordinate axis, x_2 denotes the step unit for x coordinates, and y_2 represents the step unit for y coordinates.

- DRAW statement: DRAW(x, y)
  This statement writes dots on the screen. It draws a dot at the position of the x and y coordinates from the origin (center point) specified by the INIT statement. A line can be also be drawn by connecting two or more dots like "DRAW$(x_1, y_1)-(x_2, y_2)$".

- CDRAW statement: CDRAW(x, y)
  This statement is the reversed DRAW statement. That is, it erases dots and lines specified by x and y coordinates.

- QUAD statement: QUAD x_1, x_2, y_1, y_2
  The QUAD statement is provided to draw quadrangles. Quadrangles can easily be drawn only by specifying coordinates of the four angles. However, this statement can only draw quadrangles whose opposite sides are parallel to the x or y axis.

- CQUAD statement: CQUAD x_1, x_2, y_1, y_2
  This statement is the reverse of the QUAD statement. It erases the quadrangle whose four angles have been specified.

Example 1: With the center of coordinate axes located around the center of the screen, draw a dot. Then draw x and y axes passing through this point.

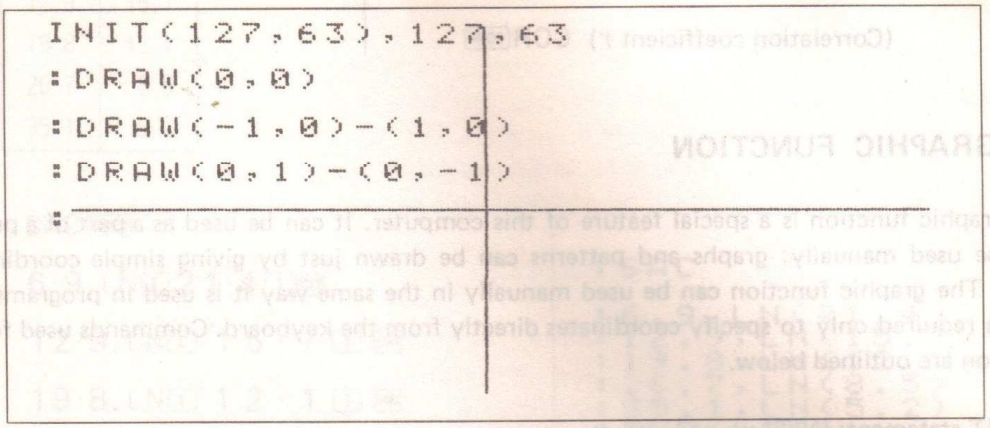
Operation: CLS

INIT (127, 63), 127, 63 ENTER

DRAW (0, 0) ENTER

DRAW (−1, 0)−(1, 0) ENTER

DRAW (0, 1)−(0, −1) ENTER

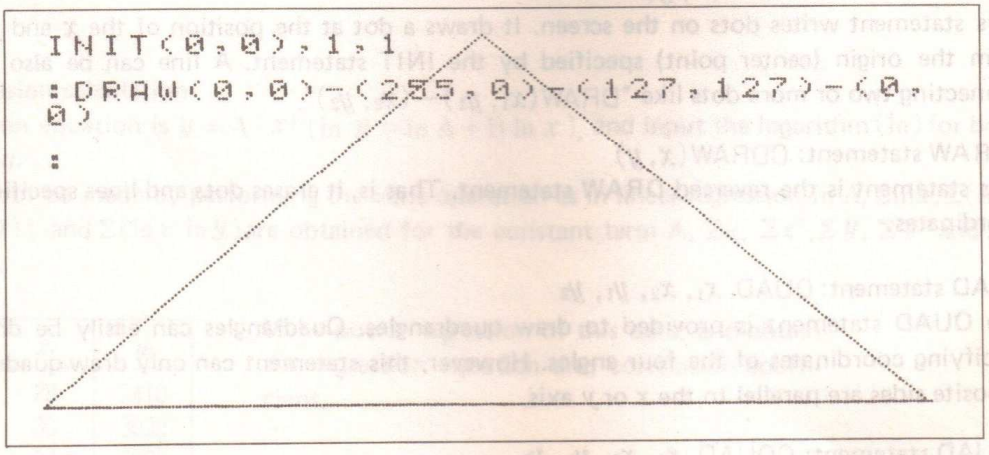


Example 2: Draw a isosceles triangle whose base is 256 dots and whose height is 128 dots.

Operation: CLS

INIT (0, 0), 1, 1 ENTER

DRAW (0, 0)−(255, 0)−(127, 127)−(0, 0) ENTER



Example 3: Erase the triangle drawn in Example 2 by using the CDRAW statement, and draw a quadrangle which occupies the full screen area.

Operation: (Continuously) CDRAW [(] 0, 0 [)] [−] [(] 255, 0 [)] [−] [(] 127, 127 [)] [−] [(] 0, 0 [)] [ENTER]

QUAD 0, 255, 0, 127 [ENTER]

```
INIT(0,0),1,1
:DRAW(0,0)-(255,0)-(127,127)-(0,
0)
:CDRAW(0,0)-(255,0)-(127,127)-(0
,0)
:QUAD 0,255,0,127
:
```

Example 4: Erase the quadrangle drawn in Example 3 by using the CQUAD statement.

Operation: (Continuously) CQUAD 0 , 255 , 0 , 127 [ENTER]

```
INIT(0,0),1,1
:DRAW(0,0)-(255,0)-(127,127)-(0,
0)
:CDRAW(0,0)-(255,0)-(127,127)-(0
,0)
:QUAD 0,255,0,127
:CQUAD 0,255,0,127
:
```

3-6 DIRECT MODE

Commands can be used in two modes: the program mode and the direct mode. In the program mode, commands written in programs are executed. In the direct mode, commands are executed manually from the keyboard. The preceding section (manual use of the graphic function) shows commands used in the direct mode.
Commands are executed in the same way in both modes. However, commands cannot be used repetitively in the direct mode, and must be input every time they are to be executed.
Commands must conform to the syntax of the CA-BASIC language. For details of the command types and functions, refer to the **"CA-BASIC Reference Manual"**.

Example: Assign $1200 to the character variable A$ and $450 to B$, and display both.

Operation:

- First, assign $1200 to the character variable A$ and $450 to B$.

  A$ [=] ["] [$] 1200 ["] [ENTER]

  B$ [=] ["] [$] 450 ["] [ENTER]

  This assignment operation can also be performed in the direct mode, i.e. an assignment statement can be executed in the program mode or the direct mode.

```
A$="$1200"
:B$="$450"
:_
```

- Then display contents of A$ and B$. Although the content of a numerical variable can be found by using the [COMP] key, the content of a character variable can be obtained only by entering the PRINT statement.

  PRINT A [$] [ENTER]

  PRINT B [$] [ENTER]

This operation can also be performed in the direct mode.

```
A$="$1200"
:B$="$450"
:PRINT A$
$1200
:PRINT B$
$450
:_
```

As described above, some of the commands used in the program mode can be used in the direct mode. The user, by entering such a command manually, can use program statements.
For commands (including commands which can be used in the direct mode), see the listing of commands given at the end of the CA-BASIC Reference Manual.

CHAPTER 4
PROGRAM CALCULATION

This computer can be programmed using the following procedure.

1) Code a program to carry out the desired calculation.
2) Store the program in the computer.
3) Enter data to be used by the program.

This manual only explains steps 2) and 3). For programming, refer to the CA-BASIC Reference Manual.

4-1 WRITING PROGRAMS TO MEMORY

To write a program to the computer, operate the keys on the keyboard using the following procedure.
write a program to the computer, operate the keys on the keyboard using the following procedure.

1. Specify a section in the program area.
2. Input the program line by line (writing to the computer).

The program area can be divided into 10 sections (P0, P1, P2, ..., P9). The program is entered in one of them.

1. Program area specification

Program areas can be specified in two ways.

(1) Execution type specification

When a numeral key from 0 to 9 is pressed in the shift-lock mode, the corresponding section of the program area is specified. Similarly, any program stored there will be executed.

SHIFT P0 → P0	SHIFT P5 → P5
SHIFT P1 → P1	SHIFT P6 → P6
SHIFT P2 → P2	SHIFT P7 → P7
SHIFT P3 → P3	SHIFT P8 → P8
SHIFT P4 → P4	SHIFT P9 → P9

(2) Non-execution type specification

Specify the program area as follows when the program does not need to be executed. Program execution may not be desired when the area contains an instruction which should not be executed, such as "CLEAR DATA".

PRG n ENTER [n is a number from 0 to 9]

Example: Specify P5.

PRG 5 ENTER

2. Programming the computer

A program is written to the computer line by line. It begins with ":" and must not exceed 94 characters including line number. Finally, the [ENTER] key is pressed to store it in memory.

★ Function of [ENTER] key

The [ENTER] key is used to write programs to the computer, execute commands, and execute manual calculations in the direct mode.

Programs are entered line by line. However, a line cannot be stored in the computer without pressing the [ENTER] key.

For all the following purposes, therefore, the [ENTER] key must be pressed as the final step.

- Writing programs.
- Modifying or deleting part of a program.
- Adding to a program.

Even if the data on the screen has been changed, the corresponding data in memory will not be changed until the [ENTER] key is pressed.

Example: Write the following program (written on a coding sheet) to P4.

PROGRAM NO._____ NAME _____

LINE NO.	10	20	30
10	INPUT A, B		
20	V1=A+B		
30	V2=A-B		
40	PRINT V1, V2		
50	END		

Operation:

① First, specify P4 for the program area.

PRG 4 [ENTER]

(The same effect is obtained by [SHIFT] [P4])

```
:PRG 4
READY P4
:_
```

② If previously written programs remain in the program area, clear them. This step may be omitted when no programs are stored in P4.

CLEAR [ENTER]

```
:PRG 4
READY P4
:CLEAR
READY P4
:_
```

③ Write line 10 to the computer.

10␣INPUT␣A, B [ENTER]

Symbol to give a space corresponding to the one-character length (Omissible)

Press at the end of a line.

```
:PRG 4

READY P4
:CLEAR

READY P4
:10 INPUT A,B
:_
```

④ Write line 20 to the computer.

20␣V1 [=] A [+] B [ENTER]

```
:PRG 4

READY P4
:CLEAR

READY P4
:10 INPUT A,B
:20 V1=A+B
:_
```

⑤ Write line 30 to the computer.

30␣V2 [=] A [−] B [ENTER]

```
:PRG 4

READY P4
:CLEAR

READY P4
:10 INPUT A,B
:20 V1=A+B
:30 V2=A-B
:_
```

⑥ Write line 40 to the computer.

40␣PRINT␣V1, V2 [ENTER]

```
:PRG 4

READY P4
:CLEAR

READY P4
:10 INPUT A,B
:20 V1=A+B
:30 V2=A-B
:40 PRINT V1,V2
:_
```

⑦ Write line 50 to the computer.

50␣END [ENTER]

```
:PRG 4

READY P4
:CLEAR

READY P4
:10 INPUT A,B
:20 V1=A+B
:30 V2=A-B
:40 PRINT V1,V2
:50 END
:_
```

- The one-character space between the line number and the command or between the command and the operand is used just for convenience. This space can be omitted because it is ignored by CA-BASIC.
- In this example, the line number is given in steps of ten. Although line numbers from 1 to 9999 can be used, those in steps of ten are useful because more lines may be added or inserted later. Since programs are executed by ascending line number, program lines must be numbered according to the order in which they are to be executed.

4-2 PROGRAM EXECUTION

Programs can be executed in two ways.

(1) Program execution methods

① Program area specification method

Execution of the desired program is started when the corresponding section of the program area is specified.

SHIFT { P0) P9 } [While pressing SHIFT , press P0 to P9 .]

Example: Start the program used in the previous example.

Operation: SHIFT P4

```
?_
```

* This "?" appears because there is an INPUT statement at the beginning of the program

To execute the program from a specific line number, name the program area after input of the line number.

Example: Start the program at line 20.

Operation: 20 SHIFT P4

```
20
-1                -1
READY  P4
:
```

② Run command method

RUN ENTER ["RUN" can be given by entering R U N or SHIFT RUN .]

```
RUN
?_
```

To execute the program from a specific line number, input RUN and the line number, and press the ENTER key.

Example: Start the program from line 20.

Operation: RUN 20 ENTER

```
RUN 20
-1                -1
READY  P4
:
```

Execution from a specific line number can also be performed using the GOTO statement.

Operation: GoTo 20 ENTER

```
GOTO 20
-1          -1
READY P4
:
```

★ When method ① (above) is used, the program area does not have to be set, but, when method ② is used, the program area must be specified.

(2) Key input during program execution

Both the INPUT and the KEYIN statements allow keyboard input operations to be performed during program execution.

- Key input using KEYIN statement
  Only one character can be entered. When the required key is pressed, the program proceeds to the next execution step.

- Key input using INPUT statement
  A question mark "?" is displayed, and execution stops until data is input. When the ENTER key is pressed, execution resumes.

Example: Execute the program written in P4 (the program given in the earlier example).

Operation:

- Execute the program:

  SHIFT P4

```
?_
```

- Two variables are input for this program. First, enter variable A.

  47 ENTER

```
?47
?_
```

- Next, enter variable B.

  69 ENTER

```
?47
?69
 116        -22
READY P4
:_
```

As shown above, data is input by performing the following operation.

Data ENTER

4-3 PROGRAM EDITING

The following operations are collectively called program editing. They are performed to assure that programs are logically correct and executable.

- Modify, add, or delete a program line.
- Re-number program lines.

Program editing is carried out from the keyboard in the direct mode.

- Program editing commands
  (1) Display the program list. ..... LIST
  (2) Modify, add, or delete a program ..... EDIT

(1) LIST (display of the program contents):

The LIST command format is as follows.

LIST [line No.] ENTER

["line No." can be omitted]

The LIST command is frequently used to edit or debug (see page 49) programs. When LIST is entered, all the subsequent program lines are displayed on the video screen sequentially (in ascending order) from the specified line number. When the line number is omitted from the LIST command, program lines are displayed sequentially from the first line.

Example: List the program written in P4 in the earlier example.

① List all the program lines from the first line.

Operation: LIST ENTER

```
LIST

  10  INPUT A,B
  20  V1=A+B
  30  V2=A-B
  40  PRINT V1,V2
  50  END

READY P4
:_
```

② List line 30 and subsequent lines of the program.

Operation: LIST 30 ENTER

```
LIST30

  30  V2=A-B
  40  PRINT V1,V2
  50  END

READY P4
:_
```

※ When a LIST command is executed, the program lines are displayed sequentially until the program ends. When all of the contents of a long program are being listed, the C/S key can be used to pause scrolling screen display. The halted LIST command can be resumed by pressing the C/S key again.

(2) Modification, addition, and deletion of a program

① Modification

Programs can be modified by using the EDIT command on a line-by-line basis.

EDIT [line No.] ENTER

["line No." can be omitted]

When an EDIT command is entered, program lines are sequentially displayed one by one starting from the specified line number. When the line number is omitted from the EDIT command, program lines are displayed from the first line.

a. Partial modification

Example: Change "+" to "*" in line 20 in the earlier example.

Operation:

- When P4 is not specified for the program area, specify P4.

  PRG 4 ENTER

```
PRG 4

READY P4
:_
```

- Call line 20 by using an EDIT command.

  EDIT 20 ENTER

  (Instead of "EDIT", SHIFT EDIT can be used).

```
PRG 4

READY P4
:EDIT 20

  20 V1=A+B_
```

- Move the cursor to the position at which the modification is to be made (i.e. "+").

  ⇦ ⇦

```
PRG 4

READY P4
:EDIT20

  20 V1=A+B
```

- Change the character over the cursor as desired.

  * ENTER

※ Be sure to press the ENTER key.

```
PRG 4

READY P4
:EDIT 20

  20 V1=A*B
  30 V2=A-B_
```

- After line 20 is changed, line 30 appears for modification. If there are no more lines to be modified, enter "break" to terminate the EDIT command.

  SHIFT BRK

```
PRG 4

READY P4
:EDIT 20

  20 V1=A*B
  30 V2=A-B
READY P4
:_
```

※ If any key other than ENTER or SHIFT BRK is pressed when a line, which does not need to be modified, is displayed, that line will be modified. Be careful not to press keys other than ENTER and SHIFT BRK.

- Display the program list to confirm that the modification has been performed properly.

LIST [ENTER]

```
READY  P4
:LIST

  10  INPUT  A, B
  20  V1=A*B
  30  V2=A-B
  40  PRINT  V1, V2
  50  END

READY  P4
:_
```

b. Full line change
Enter the line number of the line to be changed.
(As a result, this line number is cleared.)

Example: Change line 30 from "V2= A – B" to "V2 = V1/2".

Operation:

- Write new line 30
  30 ␣ V2=V1/2 [ENTER]

```
READY  P4
:30  V2=V1/2
:_
```

- Check the program list.
  LIST [ENTER]

```
READY  P4
:30  V2=V1/2
:LIST

  10  INPUT  A, B
  20  V1=A*B
  30  V2=V1/2
  40  PRINT  V1, V2
  50  END

READY  P4
:_
```

② Addition of program lines
When a line is to be added, the line number must be selected and entered as described below.

Example: Add "V3 = V1 * 2" to the program in the previous example (this line is to be inserted between lines 30 and 40), and change line 40 to "PRINT V1, V2, V3".

Operation:

- First, display the program list and check it.
  LIST [ENTER]

```
READY  P4
:LIST

  10  INPUT  A, B
  20  V1=A*B
  30  V2=V1/2
  40  PRINT  V1, V2
  50  END

READY  P4
:_
```

- Assign line number 35 to the new line so that it will be inserted between lines 30 and 40. Then enter this line.

  35 ␣ V3 = V1 * 2 ENTER

※ For the line number of the additional line, any line number existing between the lines where the additional line is inserted can be used. In this example, therefore, any line number between 30 and 40 (i.e. 31 to 39) can be used.

```
READY  P4
:LIST

   10  INPUT  A,B
   20  V1=A*B
   30  V2=V1/2
   40  PRINT  V1,V2
   50  END

READY  P4
:35  V3=V1*2
:_
```

- To modify line 40, call it by using an EDIT command. Then add ",V3" to this line.

  EDIT 40 ENTER

  ▷ V 3 ENTER

  SHIFT BRK

```
:LIST

   10  INPUT  A,B
   20  V1=A*B
   30  V2=V1/2
   40  PRINT  V1,V2
   50  END

READY  P4
:35  V3=V1*2
:EDIT40

   40  PRINT  V1,V2,V3
   50  END
READY  P4
:_
```

- Display the program list to verify that the new line has been added and line 40 has been correctly modified.

  LIST ENTER

```
READY  P4
:LIST

   10  INPUT  A,B
   20  V1=A*B
   30  V2=V1/2
   35  V3=V1*2
   40  PRINT  V1,V2,V3
   50  END

READY  P4
:_
```

③ Deletion

a. Partial deletion

Example: Delete "V1," from line 40 in the previous example.

Operation:

- As in the partial modification procedure, call line 40 by using the EDIT command.

EDIT 40 [ENTER]

```
EDIT 40
  40 PRINT V1,V2,V3_
```

- Move the cursor to the position where the deletion is to be made, i.e., the first "V".

[←][←][←][←][←][←][←][←]

```
EDIT 40
  40 PRINT V1,V2,V3
```

- Delete "V1," by using the [SHIFT] [DEL] key.

[SHIFT] [DEL] [SHIFT] [DEL] [SHIFT] [DEL] [ENTER] [SHIFT] [BRK]

※ If the [ENTER] key is not pressed, the program contents remain unchanged.

※ After "V1," is deleted from line 40, the EDIT command for modification is effective for line 50. Therefore, "break" must be entered if line 50 does not need to be modified.

```
READY P4
:EDIT 40

  40 PRINT V2,V3
  50 END
READY P4
:_
```

- Display the program list to verify that the deletion has been properly performed.

LIST [ENTER]

```
:EDIT 40

  40 PRINT V2,V3
  50 END
READY P4
:LIST

  10 INPUT A,B
  20 V1=A*B
  30 V2=V1/2
  35 V3=V1*2
  40 PRINT V2,V3
  50 END

READY P4
:_
```

b. Full line deletion

A program line can be completely cleared only by entering its line number.

Example: Delete line 30.

Operation:

- Enter the line number, "30", to be deleted.

  30 ENTER

```
READY  P4
:30
:_
```

- Check whether the line has been deleted.

  LIST ENTER

```
READY  P4
:LIST

 10  INPUT  A,B
 20  V1=A*B
 35  V3=V1*2
 40  PRINT  V2,V3
 50  END

READY  P4
:_
```

(3) Renumbering program lines

Program lines can be renumbered by simply using the EDIT command. Program lines can be moved within a program.

a. To move a program line to another number position with its contents unchanged.

Example: The following program is written in P2.

```
10  INPUT  N
20  I=0:T=0
30  INPUT  A
40  IF  I<N  THEN  30
50  I=I+1
60  T=T+A
70  PRINT  T
80  END
```

Move line 40 (IF statement) to a position between lines 60 and 70.

Operation:

- Call line 40 by using the EDIT command.

  EDIT 40 ENTER

```
READY  P2
:EDIT40

  40  IF  I<N  THEN  30_
```

- Move the cursor to the position below the "4" of the line number "40".

  ⇦⇦⇦ ····· ⇦

```
READY  P2
:EDIT40

  40  IF  I<N  THEN  30
```

- Change 40 to 65 and enter it.

  65 ENTER

```
READY  P2
:EDIT40

  65  IF  I<N  THEN  30
  50  I=I+1_
```

- To terminate line number change, cancel the EDIT command by entering "break".

SHIFT BRK (CLS)

```
READY  P2
:EDIT40

*  40  IF  I<N  THEN  30
   50  I=I+1
READY  P2
:_
```

- Display the program list to check how the program has been changed.

LIST ENTER

```
READY  P2
:LIST

   10  INPUT  N
   20  I=0:T=0
   30  INPUT  A
   40  IF  I<N  THEN  30
   50  I=I+1
   60  T=T+A
   65  IF  I<N  THEN  30
   70  PRINT  T
   80  END

READY  P2
:_
```

- The displayed program list shows that, although the IF statement of line 40 has been moved to line 65, the line 40 remains. Delete this unnecessary line.

40 ENTER

- Line number reassignment is now finished. Display the program list to verify the operation.

LIST ENTER

```
READY  P2
:LIST

   10  INPUT  N
   20  I=0:T=0
   30  INPUT  A
   50  I=I+1
   60  T=T+A
   65  IF  I<N  THEN  30
   70  PRINT  T
   80  END
```

b. To move a program line to another number position with its contents changed.

Example: The following program is contained in P6.

```
10  INPUT  X
20  I=0
30  Y=SQR(X)
40  IF  Y<=1  THEN  70
50  I=I+1
60  X=Y
70  GOTO  30
80  PRINT  I
90  END
```

Change "70" to "80" in the IF statement in line 40, and move this line to a position between lines 50 and 60.

Operation:

- Call line 40 by using the EDIT command.

EDIT 40 ENTER

```
READY P6
:EDIT40
  40 IF Y<=1 THEN 70_
```

- Moving the cursor, change "70" to "80", and "40" to "55".

← ← 8
← ← ····· ← 55 ENTER

```
READY P6
:EDIT40
  55 IF Y<=1 THEN 80
  50 I=I+1_
```

- Display the program list to check the contents.

SHIFT BRK CLS
LIST ENTER

```
READY P6
:LIST
  10 INPUT X
  20 I=0
  30 Y=SQR(X)
  40 IF Y<=1 THEN 70
  50 I=I+1
  55 IF Y<=1 THEN 80
  60 X=Y
  70 GOTO 30
  80 PRINT I
  90 END

READY P6
:_
```

- The program list shows that the IF statement in line 40 remains. Therefore, delete line 40.

40 ENTER

```
READY P6
:40
:_
```

- Display the program list to verify the operation.

LIST ENTER

```
READY P6
:LIST
  10 INPUT X
  20 I=0
  30 Y=SQR(X)
  50 I=I+1
  55 IF Y<=1 THEN 80
  60 X=Y
  70 GOTO 30
  80 PRINT I
  90 END

READY P6
:_
```

4-4 PROGRAM DEBUG

(1) Program debugging system

Debugging operations are divided into Desk Debug and Conversational Debug.

- Debug system
  - I. Desk debug
    - a. Overall debug — Program logic check
    - b. Partial debug — Line-by-line program check
  - II. Conversational Debug — The user checks, on the video display, whether his program contains any error in the execution flow and CA-BASIC syntax, by using the automatic checking function of the computer.

Since the Desk Debug is carried out during program coding, this section explains the Conversational Debug only.

(2) Conversational debug

An error message is displayed on the video display if an error occurs when a program is being executed. Errors are displayed on a line-by-line basis. The computer will display an error code to indicate the type of error. The user then performs a manual debugging operation according to the error message displayed.

The program editing function described in the previous section is used to perform this debugging operation.

See the table of error messages on page 54.

- Error message interpretation

Error messages displayed during program execution use the following format.

```
*** ERR-   IN        ***
```

Error code — Line No.

The error code is a number and represents the details of the error that occurred. The line No. represents the line in which the error was detected.

For example, if " *** ERR-2 IN 10 *** " is displayed, line 10 contains a syntax error (the line is incorrect in format).

If " *** ERR-8 IN 35 *** " is displayed, line 35 contains a READ-DATA error (the DATA statement is insufficient for the READ statement).

Example:

```
10 INPUT X
20 Y=X↑2+3*X+15
30 PRINT Y
40 END
```

Line 20 of this program has been entered incorrectly.

```
20 Y=X↑2+3X+15
```

Operation:

- Execute this program. A "?" is displayed by the INPUT statement in line 10.

  RUN [ENTER]

```
RUN
?_
```

- At this time, for example, "45" is input.

  45 [ENTER]

```
RUN
?45
 *** ERR-2 IN 20 ***
READY P0
:_
```

- Since this error message indicates that a syntax error has occurred in line 20, check the program using the LIST command.

  LIST [ENTER]

```
RUN
?45
 *** ERR-2 IN 20 ***
READY P0
:LIST

  10 INPUT X
  20 Y=X↑2+3X+15
  30 PRINT Y
  40 END

READY P0
:_
```

- Look at the 20th line. An "✱" is dropped between "3" and "X". Correct it using the program editing procedure.

  EDIT 20 [ENTER]

  [←][←][←][←]

  [SHIFT] [INS]

  [✱] [ENTER]

  [SHIFT] [BRK]

  LIST [ENTER]

```
READY P0
:EDIT20

  20 Y=X↑2+3*X+15
  30 PRINT Y
READY P0
:LIST

  10 INPUT X
  20 Y=X↑2+3*X+15
  30 PRINT Y
  40 END

READY P0
:_
```

(3) Debug in conjunction with program execution

The conversational debug function allows the user to correct his program using information given by error messages from the computer. However, the user may not be able to obtain satisfactory calculation results even though no error message is displayed. In such a case, he can debug his program by executing it repetitively and checking the calculation results up to a point in the middle.

This type of debug requires the STOP command, which halts the program run at a specific point. This halt allows the user to check the variable values and characters and view the calculation up to that point. After this check, he can resume the program by pressing the C/S key.

Example: The following program is written in P8.

```
10  Y=0
20  INPUT  N,X
30  FOR  I=1  TO  N
40  Y=Y+X↑2
50  NEXT  I
60  PRINT  Y
70  END
```

Use the STOP statement to view the result of each operation of the FOR·NEXT loop to check the value of Y in the loop.

Operation:

- Since a STOP statement should follow a calculation formula, write it between lines 40 and 50.

```
45 STOP [ENTER]
```

```
:45 STOP
:_
```

- The STOP statement causes the program to be halted after the calculation in line 40 is finished. Then the user can check the results.

```
RUN [ENTER]
4 [ENTER]
87 [ENTER]
```

```
READY P0
:RUN

?4
?87

:_
```

- What is the value of Y when the program run is halted?

```
Y [COMP]
```

```
READY P0
:RUN

?4
?87

:Y
 7569
:_
```

- Resume the program. The run is halted by the next STOP statement. Then obtain the value of Y again.

C/S

Y COMP

```
READY P0
:RUN

?4
?87

:Y
 7569
:
:Y
 15138
:_
```

- The user can view the calculation process by repeating this operation.

```
:Y
 15138
:
:Y
 22707
:
:Y
 30276
:
 30276
READY P0
:_
```

Although this is a simple program, this debug function is useful for preparing complicated programs for which desk debug is not so effective in checking calculation processes. The STOP statement allows the user to find programming errors in complicated programs by checking variables and calculation results.

CHAPTER 5
PROGRAM/DATA CLEARING

There are three commands to clear (erase) stored programs and data: CLEAR A, CLEAR, and CLEAR DATA.

(1) CLEAR A [CLEAR A ENTER]

"CLEAR A" stands for "Clear All". When entered, this command clears all programs and data stored in the user's area. Regardless of the program division (P0 to P9), the CLEAR A command clears memory containing programs or data, presets the angle unit to Degree (SET D), and, for the INIT specifications, presets the origin to (0, 0) and the step to 1.

Character variables are also cleared.

Programs and data in files are not erased when RAM files are specified. Also, the CLEAR A command does not clear the memory for statistics calculation and previous answer calling (ANS) used for manual calculation.

(2) CLEAR [CLEAR ENTER]

"CLEAR" is a command to clear programs. It clears only those programs in preset program areas. When P0 is preset, for example, only programs written in P0 are cleared.

(3) CLEAR DATA [CLEAR DATA ENTER]

"CLEAR DATA" is a command to clear data. Regardless of the program area preset, it clears all memory (variables) containing data.

This command can be used in both modes; in the manual operation mode and as a command incorporated in a program.

ERROR MESSAGES

Error code	Meaning	Error cause	Remedy
1	Memory overflow or system stack overflow.	• The memory capacity is too small for programs or data. • The mathematic expression is too complicated.	• Expand the memory or reduce the data used for arrays, etc.
2	Syntax error	• The program contains a format error. • The left and right sides of an assignment statement are different in format.	• Correct the error in the entered program or data.
3	Mathematic error	• The calculation result of a mathematic expression is 10^{100} or more. • An argument of a numerical function is outside the input range. • The result is "indefinite" or "impossible".	• Modify the calculation formula or data. • Modify the program so that the data is verified as to whether or not it is within range.
4	Undefined line number error	• The GOTO, GOSUB statement lacks the specified line number.	• Correct the specified line number.
5	Array error	• One array has been defined twice. • An undefined array was used.	• Add a CLEAR DATA statement. • Issue an array declaration.
6	Argument error	• An argument outside the input range is used for commands or functions which requrie arguments. • An array argument is a number outside the range of 1 to 255. • A different dimension is specified for array use.	• Correct the incorrect argument.
7	Nesting error	• A RETURN statement appears despite the fact that no subroutine is being executed. • A NEXT statement appears despite the fact the FOR loop is not running. • The number of subroutine levels exceeds 10. • The number of FOR·NEXT loop levels exceeds 10.	• Delete unnecessary RETURN and NEXT statements. • Reduce the number of subroutines and FOR·NEXT loop levels to 10 or less.

Error code	Meaning	Error cause	Remedy
8	READ-DATA error	• Data is insufficient for the READ statement.	• Use as many data items as required by the READ statement.
9	RAM error	• An RPUT or RGET command was executed when the RAM file was not opened. • A file read or write operation was performed when the RAM file was not defined.	• Enter an OPEN command to open the RAM file. • Define the RAM file.
11	Password error	• A command which cannot be used (such as LIST and EDIT) was used with the password cataloged. • When passwords are cataloged, a password which differs from the catalog was entered.	• Input the correct password.
21	Mini-printer error	• An attempt to execute a command for printer output was made when the power to the mini-printer is not turned ON or the mini-printer is not connected to the power supply.	• Connect the printer.
22	MT parity error	• Data from MT was not read correctly. • Data to read was not found. • An attempt to read more data than recorded was made.	
23	Open/close error	• An attempt to load a program in a program area which already contains a program was made. • An attempt to execute a PUT, GET, or CLOSE command was made when the MT file is not opened.	
24	Verify error	• An attempt to compare and check an empty program area to MT was made. • The content of the program area differs from that of the program on the compared MT.	• Be sure to make a comparison check when the program is saved from the program area onto MT.

Error code	Meaning	Error cause	Remedy
30	Array argument error (E-4K ROM pack for matrix)	• When a matrix addition or subtraction is performed, the size of the operand does not agree with that of the number subjected to the operation. • In a matrix vector product, the size of the row of the number subjected to the operation does not agree with that of the column of the operand.	• Use the same array size.
31	Identical array name error (E-4K ROM pack for matrix)	• In a matrix vector product, the array name on the right side is also used on the left side.	• Change the array name on the left side.
32	Square matrix error (E-4K ROM pack for matrix)	• An attempt to generate a unit matrix which is not a square matrix was made. • An attempt to calculate the inverse matrix or the value of a non-square matrix was made.	• Convert the array to a square matrix.

SPECIFICATIONS

Type:	**FX-9000P**	
Basic Calculation Functions:	Four fundamental arithmetic operations including negative numbers, exponent numbers, and parenthesis calculation (with a maximum of 17 calculation levels) (with capability for automatic recognition of calculation priority regarding order of execution as to arithmetic operations for addition, subtraction, multiplication and division — true algebraic logic).	
Built-in Functions:	Trigonometric function, inverse trigonometric function (Angles in decimal degrees, radians and gradients), hyperbolic function, inverse hyperbolic function, logarithm/exponential function, factorials, square roots, powers, permutations, combinations, conversion into integer, removal of integer part, conversion of sexagesimal to decimal number, remainder calculation, absolute value, encoding, rounding, random numbers, Pi.	
Statistical Functions:	**Standard Deviations:**	Number of data items, sum, square sum, mean, standard deviation.
	Linear Regression:	Number of data items, sum of x, sum of y, square sum of x, square sum of y, data product sum, mean of x, mean of y, standard deviation of x, standard deviation of y, constant term, regression coefficient, correlation coefficient
Direct Commands:	PRG, LIST, EDIT, RUN, CLEAR, CLEAR A, PASS, RFILE, RLIST, RCLEAR, RSAVE, RLOAD.	
Basic Commands:	REM, LET, READ, DATA, RESTORE, INPUT, KEYIN, PRINT, IF-THEN, GoTo, GoSUB, RETURN, ON-GoTo, ON-GoSUB, FOR, NEXT, SET, STOP, END, DIM, CLEAR DATA, SAC, STAT, ROPEN, RPUT, RGET, RCLEAR.	
Graphic Commands:	CLEAR DISP, INIT, DRAW, CDRAW, QUAD, CQUAD.	
Numerical Functions:	SIN, COS, TAN, ASN, ACS, ATN, HSN, HCS, HTN, AHS, AHC, AHT, SQR, EXP, LN, LOG, INT, FRAC, ABS, SGN, DEG, MOD, PER, COM, ROUND, RND#, SIZE, CNT, SUMX, SUMY, SUMXY, SUMX2, SUMY2, MEANX, MEANY, SDX, SDY, LRA, LRB, COR, VAL, LEN, ASC, π.	
Character Functions:	MID$, CHR$, STR$.	
Output Control Functions:	TAB, CSR, REV, NORM.	
Graphic Functions:	DOT, CHGX, CHGY, GIN$, GOUT, POS.	
Operating Range:	Operating range is from $\pm 1 \times 10^{-99}$ to $\pm 9.99999999999 \times 10^{99}$ and 0. Calculations are performed internally with 15 calculation digits (Mantissa) in the computer.	

Functional Digit Capacity:

	Input Range	Computation Accuracy
$\sin x$, $\cos x$, $\tan x$	$\|x\| < 1440°$ (8π rad, 1600gra)	12th digit ± 1
$\sin^{-1} x$, $\cos^{-1} x$	$\|x\| \leqq 1$	〃
$\tan^{-1} x$		〃
$\sinh x$	$\|x\| \leqq 230$	〃
$\cosh x$	$-227 < x < 230$	〃
$\tanh x$		〃
$\sinh^{-1} x$		〃
$\cosh^{-1} x$	$x \geqq 1$	〃
$\tanh^{-1} x$	$x \geqq 0$	〃
$\log x$, $\ln x$	$x > 0$	〃
e^x	$-227 < x < 230$	〃
$\sqrt{x}$	$x \geqq 0$	〃
$x!$	$0 \leqq x \leqq 69$	〃
x^y ($x \uparrow y$)	$x < 0 \rightarrow y$: natural numbers	〃
Sexagesimal → decimal conversion		
${}_nP_r$, ${}_nC_r$	n, r : natural numbers ($n \geqq r$)	

Processor: Compatible with Z-80

Memory Capacity: ROM: 12k Bytes (Standard), 24k Bytes (Maximum)
RAM: 4k Bytes (Standard) ~ 32k Bytes (Maximum)
* RAM capacity is expanded by packages.

CRT Display: 5.5 inches, green colored data
Screen size of 32 characters X 16 lines
Graphic format of 256 X 128 dots
Character format of 8 X 8 dots

Power Source: AC 100/117/220 or 240V (±10V), 50/60 Hz ..... Fixed

Power Consumption: 32W

Ambient temperature and humidity range: 0°C ~ 40°C (32°F ~ 104°F); 20% ~ 85% humidity

Dimensions: 187H x 415W x 430mmD (7-3/8"H x 16-3/8"W x 17"D)

Weight: 7.2 kg (15.9 lb) (Computer unit only)

CASIO®

121CM (英) Printed in Japan.