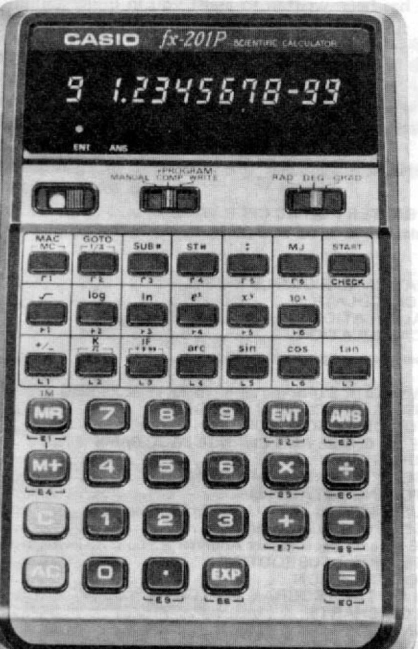
ELECTRONIC CALCULATOR

CASIO *fx-201P*

INSTRUCTION MANUAL



BM MARK FOR HIGH QUALITY & RELIABILITY

Recognized by JAPAN BUSINESS MACHINE MAKERS ASSOCIATION

INTRODUCTION

Dear Customer,

Congratulations on your purchase of the most advanced electronic calculator. This is a scientific calculator equipped with a 127-step program function that makes repeat and complex calculations easy and trouble-free. This program function is in addition to such big features as one-touch function keys that allow you to easily perform mixed calculations in the four arithmetic operations, an independent memory, 10 constant memories and 100-digit ($10^{\pm 99}$) calculation capacity.

To utilize the full features of this calculator no special training is required but we suggest you study this instruction manual to become familiar with its many abilities. It has been written to assist you in understanding the various control keys and functions of the calculator through simple examples and their applications.

INDEX

DISPOSABLE DRY BATTERY OR AC OPERATION

This calculator operates on either dry batteries or AC with the use of the AC ADAPTOR.

DRY BATTERY OPERATION

With four AA size manganese dry batteries (SUM-3) it operates for approximately 8 hours continuously.

Even when battery power decreases, the display will merely darken but cause no miscalculation. When you have finished your calculation, be sure to switch off the power to save the battery.

To change batteries, put the power switch off first. Slide open the battery cover and replace batteries.

AC OPERATION

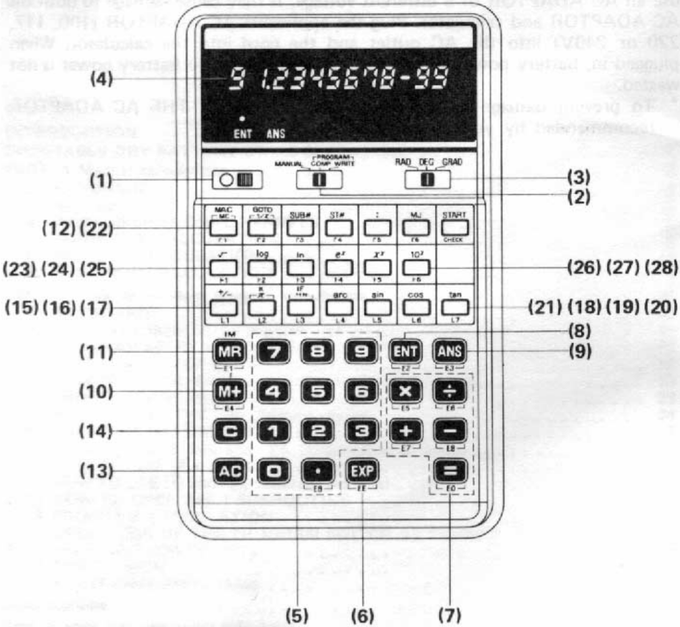
If you are in a 117V area, for instance, use a 117V AC ADAPTOR. When you use an AC ADAPTOR of a different voltage, it may cause damage to both the AC ADAPTOR and calculator. Plug the applicable AC ADAPTOR (100, 117, 220 or 240V) into the AC outlet and the cord into the calculator. When plugged in, battery power supply stops automatically, so battery power is not wasted.

* To prevent damage to the calculator, USE ONLY THE AC ADAPTOR recommended by your dealer.

PART-1
Manual calculations

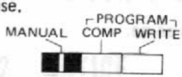
In Part 1 we will explain the functions of the calculator, excluding the program function, and the method of operation. That is, in this part, we will explain only what is necessary for manual calculating.

1-1. KEYBOARD



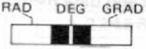
(1) POWER SWITCH:

Move the switch to the right and "O." appears to show the calculator is ready for use.



(2) PROGRAM SWITCH:

When performing manual calculations, set the program switch to the "MANUAL" position.



(3) ANGULAR MODE SELECTOR:

By setting the selector either at "RAD" (Radian), "DEG" (Degree) or "GRAD" (Grade) position, trigonometric (or inverse trigonometric) functions can be performed based on the angular measurement indicated by the selector.

(4) 0. READ–OUT:

Shows each entry and result, whether in the regular 10 digit display or in scientific notation, through a green digitron tube, suppressing unnecessary O's (zeroes).

In some calculations, the "–" sign appears momentarily while complicated formulas are being processed. So do not enter numerals or press the function keys until the previous answers are displayed.

The data memory numbers are displayed in the upper digit along with the lighting of the ENT lamp (When values are stored in the memory) and the ANS lamp (When values are read out). The ENT lamp is indicated by (ENT) and the ANS lamp by (ANS) in this manual.

(5) 0 ~ 9 · NUMERAL & DECIMAL POINT KEYS:

Enters numerals. For decimal places, use the · key in its logical sequence.

(6) EXP ENTER EXPONENT KEY:

Enter the exponent of ten up to $10^{\pm 99}$. To enter 2.56 x 10^{45}, operate 2 · 5 6 EXP 4 5.

(7) + , − , × , ÷ , = FUNCTION COMMAND AND RESULT KEYS:

Press the numeral and function command keys in the same logical sequence and the = key obtains the answer.

(8) ENT DATA INPUT KEY:

Press to store displayed number in a data memory. To store 12.3 in data memory number 3, for example, press 1 2 · 3 ENT 3.

(9) ANS DATA OUTPUT KEY:

Press to display the number stored in a data memory. To display the contents of data memory number 9, for example, press ANS 9.

* There are 10 data memories : 1~9 and 0.

The contents do not change until a new entry is made.

(10) M+ MEMORY PLUS KEY:

Transfers the number displayed into the memory positively. Obtains answers in four functions and N-th power calculation, and automatically accumulates them into the memory positively.

(11) MR MEMORY RECALL KEY:

Recalls contents of the memory without clearing the same.

(12) MC MEMORY CLEAR KEY:

Clears contents of the memory.

(13) AC ALL CLEAR KEY:

Clears the entire machine except the independent memory and data memories, and releases an overflow check.

(14) C CLEAR KEY:

Clears keyboard entry for correction (including entries in scientific notation), and also clears answers of functions while performing mixed calculations. Function command keys (+, −, x, ÷, x^y) can be interchanged and the last function key depressed is effective.

(15) +/- SIGN CHANGE KEY:

Changes the sign of the number displayed from plus to minus and vice versa.

(16) π Pi KEY:

Enters the circular constant in 10 digits (3.141592653).

(17) °′″ SEXAGESIMAL → DECIMAL CONVERSION KEY:

Converts the sexagesimal figure to the decimal scale.

(18) sin SINE KEY:

Obtains the sign for the angle on display.

(19) cos COSINE KEY:

Obtains the cosine for the angle on display.

(20) tan TANGENT KEY:

Obtains the tangent for the angle on display.

(21) arc ARC KEY:

Performs inverse trigonometric functions in combination with the sin, cos or tan key.

(22) 1/x RECIPROCAL KEY:

Obtains the reciprocal of the number displayed.

(23) √ SQUARE ROOT KEY:

Obtains the square root of the number displayed.

(24) log COMMON LOGARITHM KEY:

Obtains the common logarithm of the number displayed.

(25) ln NATURAL LOGARITHM KEY:

Obtains the natural logarithm of the number displayed.

(26) e^x EXPONENTIAL KEY:

Raises the constant e (2.7182818.....) to x powers.

(27) x^y POWER RAISING KEY:

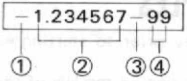
Raises the base x to y powers.

(28) 10^x X POWER OF 10 KEY (INVERSE LOG KEY):

Raises the constant 10 to x powers.

1-2. NOTICE

(1) SCIENTIFIC NOTATION



① The minus (–) sign for mantissa.
② The mantissa.
③ The minus (–) sign for exponent.
④ The exponent of ten.

When the answer is more than 1×10^{10} or less than 1×10^{-2}, it is automatically shown by the scientific notation, 8 digit mantissa (7 negative digits) and exponent of ten up to ±99.
Entry can also be made in the form of scientific notation by using the (Enter Exponent) key. Note that the EXP key does not work when the first entry (mantissa) is made exceeding 8 digits (7 digits, when the figure is negative).

EXAMPLE	OPERATION	READ-OUT
$(1.23 \times 10^{10}) + (4.56 \times 10^{7})$	1⊡23 EXP	1.23 00
$= 1.23456 \times 10^{10}$	10	1.23 10
	+	1.23 10
	4⊡56 EXP	4.56 00
	7	4.56 07
	=	1.23456 10

The answer is read: $1.23456 \times 10^{10} = 12345600000$

Remark: *Regular entries are possible up to 10 digits. No entry can be made above that even though the keys are pressed.
*Exponent entries are possible only when the mantissa part is from 1 to 8 digits (7 digits for negatives).
*When entries with 3 or more digits are made after depressing the EXP key, the final 2 digits becomes exponent.

(2) OVERFLOW CHECK

Overflow is indicated by an *"E"* sign and stops further calculations.
To release the locked registers caused by the overflow check, depress the AC key.
Overflow occurs:
1) When an answer or accumulated total in the memory becomes more than 1×10^{100}.

2) When the function calculations are performed on a number exceeding their input range.

Remarks: The content of the memory is protected against overflow and the total accumulated so far is recalled by the MR key after the overflow check is released by the AC key.

1-3. BASIC OPERATIONAL EXAMPLES

* There is no need to depress the AC or C key prior to starting each new calculation.

* A negative figure is displayed with a minus (−) sign on the left of the figure.

EXAMPLE	OPERATION	READ-OUT
$(12+3-45.6)\times89\div7$ $=-389.0571428.....$	12+3−45.6×89÷7=	−389.0571428
$963\times(56-23)=31779$	56−23×963=	31779.
$12369\times7532\times74103\fallingdotseq$ 6.9036806×10^{12} $(=6903680600000)$	12369×7532×74103=	6.9036806 12
$1.23\div56\div78.9\fallingdotseq$ 2.7838131×10^{-4} $(=0.00027838131)$	1.23÷56÷78.9=	2.7838131−04
$(7.9\times10^{56})\div(4.6\times10^{74})\times$ $(1.3\times10^{23})=$ $223260.8695....$	7.9EXP56÷4.6EXP74 ×1.3EXP23=	223260.8695

* The ± key changes the sign of displayed number from plus to minus (or vice versa). To enter the negative exponent, use the ± key before or after entering the exponent.

$(-9)\times2.6-(-12.3)=-11.1$	9±×2.6−12.3±=	−11.1
$\{(4.5\times10^{6})+(-7.8\times10^{5})\}\times(1.2\times10^{-19})\times(-2.3\times10^{-26})$ $\fallingdotseq-1.02672\times10^{-38}$	4.5EXP6+7.8±EXP5× 1.2EXP19±×2.3±EXP26±=	3720000. −1.02672−38

1-4. CONSTANT CALCULATION

ENTRY ++ (−−, ×× or ÷÷)ENTRY = ... obtains answer.

└──────→ To be set as a constant.

* To renew the constant, follow the operation step as above.

EXAMPLE	OPERATION	READ-OUT
$3+2.3=5.3$	$2\boxed{\cdot}3\boxed{+}\boxed{+}3\boxed{=}$	5.3
$6+2.3=8.3$	$6\boxed{=}$	8.3
$7-5.6=1.4$	$5\boxed{\cdot}6\boxed{-}\boxed{-}7\boxed{=}$	1.4
$(-4.5)-5.6=-10.1$	$4\boxed{\cdot}5\boxed{+/-}\boxed{=}$	−10.1
$2.3\times 12=27.6$	$12\boxed{\times}\boxed{\times}2\boxed{\cdot}3\boxed{=}$	27.6
$(-9)\times 12=-108$	$9\boxed{+/-}\boxed{=}$	−108.
$74\div 2.5=29.6$	$2\boxed{\cdot}5\boxed{\div}\boxed{\div}74\boxed{=}$	29.6
$85.2\div 2.5=34.08$	$85\boxed{\cdot}2\boxed{=}$	34.08
$17+17+17+17=6.8$	$1\boxed{\cdot}7\boxed{+}\boxed{+}\boxed{=}$	3.4
	$\boxed{=}$	5.1
	$\boxed{=}$	6.8
$1.7^2=2.89$	$1\boxed{\cdot}7\boxed{\times}\boxed{\times}\boxed{=}$	2.89
$1.7^3=4.913$	$\boxed{=}$	4.913
$1.7^4=8.3521$		8.3521
$\frac{1}{4}=0.25$	$4\boxed{\div}\boxed{\div}\boxed{=}$	1.
	$\boxed{=}$	0.25
$\frac{1}{4^2}=0.0625$	$\boxed{=}$	0.0625

* Reciprocal calculations can be performed by the use of the constant capability.

$\frac{56}{4\times(2+3)}=2.8$	$2\boxed{+}3\boxed{\times}4\boxed{\div}\boxed{\div}$	20.
	$56\boxed{=}$	2.8

1-5. MEMORY CALCULATION

This calculator is equipped with one independent memory, using $\boxed{\text{M+}}$ and $\boxed{\text{MR}}$, as well as 10 data memories using $\boxed{\text{ENT}}$ and $\boxed{\text{ANS}}$.

(1) CALCULATION USING THE INDEPENDENT MEMORY

$\boxed{\text{MC}}$ ENTRY $\boxed{\times}$ ($\boxed{\div}$, $\boxed{+}$, or $\boxed{-}$) ENTRY $\boxed{\text{M+}}$ .... Obtains answer and automatically accumulates it into the memory positively.

$\boxed{\text{MR}}$ .... Recalls the accumulated total in the memory.

→ Clears contents of the memory.

- * To accumulate a number into the memory negatively, change the sign from plus to minus by the +/- key prior to depressing the M+ key.
- * Be sure to depress the MC key prior to starting a memory calculation.

EXAMPLE	OPERATION	READ-OUT
852×147=125244	MC852×147M+	125244.
−)789×654=516006	789×654+/-M+	−516006.
−390762	MR	−390762.
70+40+100=210	MC70+40+100M+	210.
+)80− 5+ 20= 95	80− 5+ 20M+	95.
305	MR	305.
4.5×12=54	MC12××4.5M+	54.
−)5.6×12=67.2	5.6+/-M+	−67.2
+)6.4×12=76.8	6.4M+	76.8
63.6	MR	63.6

* The M+ key also works to transfer a number displayed, whether entry or result, to the memory positively as many times as the M+ key is depressed.

7+7−7+(2×3)+(2×3)=19	MC7M+M++/-M+2×3M+M+MR	19

(2) CALCULATION USING THE DATA MEMORIES

- * There are 10 data memories : 1~9 and 0. Data and answers can be freely stored in any of these.
- * Normally, displayed number is stored in the memory.
  When a new number is entered into the memory, the previous number stored is cleared automatically and the new number is stored.
- * When a number is put into a data memory the memory number and the"ENT" lamp light: when a number is recalled from the memory, the memory number and the "ANS" lamp light.

EXAMPLE	OPERATION	READ-OUT	
193.2÷23=8.4	193.2ENT1	ENT 1	193.2
193.2÷28=6.9	÷23=		8.4
193.2÷42=4.6	ANS1	ANS 1	193.2
	÷28=		6.9
	ANS1÷42=		4.6

EXAMPLE	OPERATION	READ-OUT
(1×8+1)×12345679=111111111 (2×8+2)×12345679=222222222 (3×8+3)×12345679=333333333	1[ENT][1][×]8[ENT]2[+][ANS][1][×]12345679[ENT][3][=]	111111111.
	2[ENT][1][×][ANS][2][+][ANS][1][×][ANS][3][=]	222222222.
	3[ENT][1][×][ANS][2][+][ANS][1][×][ANS][3][=]	333333333.
$\frac{9\times6+3}{(7-2)\times8}=1.425$	9[×]6[+]3[=][ENT][1]	(ENT) 1 57.
	7[−]2[×]8[=][ENT][2]	(ENT) 2 40.
	[ANS][1][÷][ANS][2][=]	1.425
(2+3)×(9−5)−(8×6)+(7÷4)=−26.25	2[+]3[=][ENT][1]9[−]5[=][ENT][2]8[×]6[=][ENT][3]7[÷]4[MC][M+]	
	[ANS][1][×][ANS][2][−][ANS][3][+][MR][=]	−26.25

1-6. FUNCTION CALCULATION

This calculator computes various specific functions at one touch, independent of basic arithmetic calculations.

* The maximum effective number of digits for functions is 8. Output accuracy is ± 1 in the 8th digit (±1 in the 7th digit for x^y).
The effective number of digits for [√], [1/x] and [°′″] is 10.

(1) SEXAGESIMAL ⟶ DECIMAL CONVERSION

The [°′″] key converts the sexagesimal figure (Degree, Minute and Second) to decimal scale.

EXAMPLE	OPERATION	READ-OUT
63°52′41″=63.87805555....°	63[°′″]52[°′″]41[°′″]	63.87805555

(2) TRIGONOMETRIC FUNCTION

The [sin], [cos] and [tan] keys obtain each trigonometric value of the entry. In case the degree is given on the sexagesimal scale, it is necessary to convert the figure to the decimal scale before performing the trigonometric functions.

Input range: sin$x$/cos$x$/ tan$x$; $|x| \leq 1440°$ (8πrad, 1600 gra)

EXAMPLE	OPERATION	READ-OUT
sin63°52′41″ =0.89785901	"DEG" 63[°′″]52[°′″]41[°′″][sin]	0.89785901
$\cos(\frac{\pi}{3}\text{rad})=0.5$	"RAD" [π][÷]3[=][cos]	0.5

EXAMPLE	OPERATION	READ-OUT
tan(−35gra)=−0.61280079		
	"GRA" 35 [+/−] [tan]	−0.61280079
2sin45°×cos65°=0.597672473		
	"DEG" 2 [×] 45 [sin] [×] 65 [cos] [=]	0.597672473

(3) INVERSE TRIGONOMETRIC FUNCTION

The [arc] key performs each inverse trigonometric function in combination with the [sin] , [cos] or [tan] key.

Input range: $\sin^{-1} x$ / $\cos^{-1} x$; $|x| \leqq 1$
$\tan^{-1} x$; $|x| < 1 \times 10^{100}$

EXAMPLE	OPERATION	READ-OUT
$\sin^{-1}0.5=30°$	"DEG" [·] 5 [arc] [sin]	30.
$\cos^{-1}\frac{\sqrt{2}}{2}=0.78539816$ rad ($\fallingdotseq\frac{\pi}{4}$rad)		
	"RAD" 2 [√] [÷] 2 [=] [arc] [cos]	0.78539816
	[÷] [π] [=]	0.249999998
$\tan^{-1}0.6128=31.499967°$ (≒31°30′)		
	"DEG" [·] 6128 [arc] [tan]	31.499967
	[−] 31 [×] 60 [=]	29.99802
$\sin^{-1}0.8-\cos^{-1}0.7=7.557106°$		
	"DEG" [·] 8 [arc] [sin] [−] [·] 7 [arc] [cos] [=]	7.557106

(4) LOGARITHMS

The [log] key obtains the common logarithms of the number displayed. The [ln] key obtains the natural logarithms of the number displayed.

Input range: $0 < x < 1 \times 10^{100}$

EXAMPLE	OPERATION	READ-OUT
$\log 123=\log_{10}123=2.0899051$	123 [log]	2.0899051
$\ln 90=\log_e 90=4.4998097$	90 [ln]	4.4998097
$\log 456 \div \ln 456=0.434294475$		
	[MC] 456 [M+] [log] [÷] [MR] [ln] [=]	0.434294475

(5) EXPONENTIATIONS

The [10^x] key raises the constant 10 to x powers.
The [e^x] key raises the constant e (base) to x powers.
In another words, this is to obtain antilog$_e$ x .
The [x^y] key raises x to y powers. The number displayed when the [x^y] key is

used, is an intermediate result.

Input range: 10^x ; $|x| < 100$
e^x ; $|x| \leqq 230$
x^y ; $0 < x < 1 \times 10^{100}$

EXAMPLE	OPERATION	READ-OUT
$10^{1.23} = 16.982437$	$1\boxed{\cdot}23\boxed{10^x}$	16.982437
$e^{4.5} = 90.017131$	$4\boxed{\cdot}5\boxed{e^x}$	90.017131
$2.3^{5.6} = 106.09035$	$2\boxed{\cdot}3\boxed{x^y}5\boxed{\cdot}6\boxed{=}$	106.09035
$(78-23)^{-12} = 1.3051118 \times 10^{-21}$	$78\boxed{-}23\boxed{x^y}12\boxed{+/-}\boxed{=}$	1.3051118−21
$4^{2.5} = 32$	$2\boxed{\cdot}5\boxed{x^y}\boxed{x^y}4\boxed{=}$	32.
$0.16^{2.5} = 0.01024$	$\boxed{\cdot}16\boxed{=}$	0.01024
$5.76^{2.5} = 79.62624$	$5\boxed{\cdot}76\boxed{=}$	79.62624
$3^{12} + e^{10} = 553467.466$	$3\boxed{x^y}12\boxed{+}10\boxed{e^x}\boxed{=}$	553467.466

(6) SQUARE ROOT & RECIPROCALS

The $\boxed{\sqrt{\ }}$ key extracts the square root of the number displayed.
The $\boxed{1/x}$ key obtains the reciprocal of the number displayed.

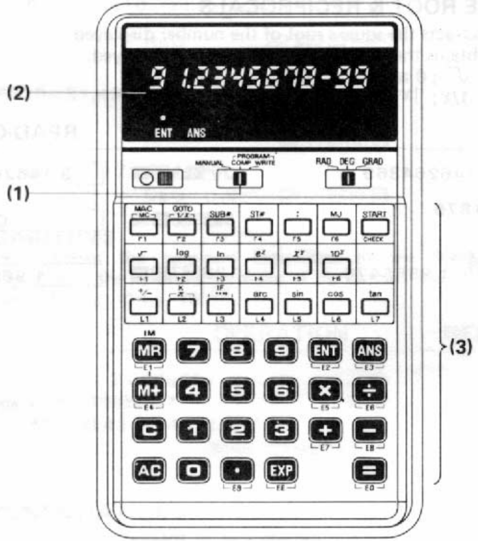
Input range: $\sqrt{\ }$; $0 \leqq x < 1 \times 10^{100}$
$1/x$; $|x| < 1 \times 10^{100}$ $x \neq 0$

EXAMPLE	OPERATION	READ-OUT
$\sqrt{2} + \sqrt{3} = 3.146264369$	$2\boxed{\sqrt{\ }}\boxed{+}3\boxed{\sqrt{\ }}\boxed{=}$	3.146264369
$\dfrac{1}{5+\dfrac{1}{3}} = 0.1875$	$5\boxed{+}3\boxed{1/x}\boxed{=}\boxed{1/x}$	0.1875
$\sqrt[7]{123} = 123^{\frac{1}{7}} = 1.9886478$	$123\boxed{x^y}7\boxed{1/x}\boxed{=}$	1.9886478

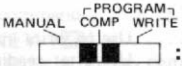
PART-2
How to perform program calculations

Program calculation is explained in Part 2. Whether or not a program is convenient is determined by the program content. Since programming is all based on theory, perfect understanding of the basic principles allows better programming and more efficient use of this calculator.

2-1. KEYBOARD



(1) PROGRAM SWITCH



:

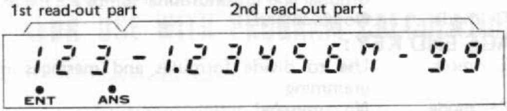
MANUAL (Manual mode) ...Set to this position for manual calculations where a program is not used. This position will be indicated by < MANUAL> in this instruction manual.

COMP (Compute mode) ... Set to this position to perform calculations using a program.
Indicated by <COMP>.

WRITE (Write mode) .... Set to this position when storing a program in the calculator, or to check a stored program.
Indicated by <WRITE>.

Note: When the C or MAC key is depressed immediately after the program switch position is changed, 0.0000 . . . . way appear in the read-out. Subsequent calculations, however, can be made correctly.

(2) READ-OUT :



- * 1st read-out part . . . . . the first 3 digits of the read-out display program step numbers or data memory number.
- * 2nd read-out part . . . . . displays entries, answers, or the program contents as code number or values. "–" also lights in the first column when the calculator is calculating.
- * ENT (entry) lamp . . . . . This lamp lights together with the display of the memory number when waiting the input of program data. Indicated by (ENT)
- * ANS (answer) lamp . . . . . This lamp lights together with the memory number when an answer to a program calculation is displayed. Indicated by (ANS).
- * The operation of the angular mode selector is the same as for manual calculations.

(3) OPERATION KEYS :

The numeral and decimal point keys (1 ~ 9 , 0 , .), command keys (+ , − , × , ÷ , = , +/− , EXP) and function calculation keys (√ , log , ln , e^x , x^y , 10^x , arc , sin , cos , tan ,) are used in the <WRITE> mode to write program into the calculator. In the <COMP> mode, they work to give the respective calculation commands.

ENT DATA ENTRY KEY :

<WRITE> mode . . . . . Use to write in data input messages.
<COMP> mode . . . . . Use to advance the program by the operation Data ENT .

ANS ANSWER KEY :

<WRITE> mode . . . . .Use to write in answer display messages.
<COMP> mode . . . . .Use after reading out an answer to advance the program.

MAC MEMORY ALL CLEAR KEY (MC) :

<WRITE> mode . . . . .Use to write in a clear command for the 10 data memories and I-memory.
<COMP> mode . . . . .Clears 10 data memories and I-memory.

It works as independent memory clear (MC) only in the <MANUAL> mode.

GOTO GOTO KEY (1/x) :

<WRITE> mode. . . . .Use to write in unconditional jump commands.
<COMP> mode. . . . .Works as 1/x (reciprocal key).

SUB# SUBROUTINE KEY :

<WRITE> mode . . . . .Use to write in subroutine programs.
<COMP> mode . . . . .No command.

ST# STATEMENT NUMBER KEY :

<WRITE> mode . . . . .Use to write in the address to which both conditional and unconditional jumps are made.
<COMP> mode . . . . .No command.

: MESSAGE END KEY :

<WRITE> mode . . . . .Use to divide formulas and messages in programming.
<COMP> mode . . . . .No command.

MJ MANUAL JUMP KEY :

<WRITE> mode . . . . .Use to write in MJ commands in programs.
<COMP> mode . . . . .Use during execution of a program to make a jump at the MJ position in the program.

CHECK CHECK KEY (START START KEY) :

<WRITE> mode . . . . .Use when advancing a written program ahead 1 step (called program check). Shown as CHK in this manual.
<COMP> mode . . . . .Use to start a program calculation (to read the program from the first step). Shown as STA in this manual.

K CONSTANT WRITE-IN KEY (π) :

<WRITE> mode . . . . .Use to write in constants in a program.
<COMP> mode . . . . .Works as a π (Pi) key.

IF IF KEY (°''') :

<WRITE> mode . . . . .Use to write in conditional jump commands.
<COMP> mode . . . . .Works as a °''' (Sexagesimal → decimal conversion key.)

I INDIRECT KEY (M+) :

<WRITE> mode . . . . .Use to write in the command to store a value in the I-memory

<COMP> mode . . . . .No command

IM INDIRECT MEMORY KEY (MR) :

<WRITE> mode . . . . .Use to write in the command to indirectly address the values stored in the I-memory during calculation.

<COMP> mode . . . . .No command.

It works as MR (Independent memory recall) only in <MANUAL> mode.

C CLEAR KEY :

<WRITE> mode . . . . .Use to back up a written program one step and clear.

<COMP> mode . . . . .Use to clear displayed data or answers.

AC ALL CLEAR KEY :

<WRITE> mode . . . . .Use to erase a written program.

<COMP> mode . . . . .Use when desired to stop a program calculation.

2–2. INTERNAL MEMORIES (FOR USE WITH PROGRAM CALCULATIONS)

Arithmetic operations register		
Function calculations register		
Independent memory	1	. . . . can not be used in program calculations
Data memory 1 ≀ Data memory 9 Date memory 0	10	. . . . for storing data and answers during calculations
I-memory	1	. . . . for indirect address indication, only 1 digit (1~9, 0) can be stored.
Program memory (127 steps)		

2–3. PROGRAM STEPS AND COMMAND CODES

When the program switch is set at <WRITE> position, step numbers and command codes are displayed to indicate what step up a program is being written in or during check to indicate what step belongs to what command, etc.

* Step number . . . . . Displayed in the 1st read-out part.
Steps are normally counted using one key for each step.

* Command code . . . Three steps are displayed simultaneously in the 2nd read-out part.

Command codes are written below the command keys and consist of symbols (┌, ├, L, E) and numerals (1~9, 0).

For example:

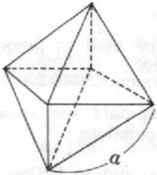
037	E2	6	┌5
Step No.	Second previous command	Previous command	Displayed command step

2-4. EXPLANATION OF BASIC PROGRAMS

* The steps for using a program to make a calculation are as follows.

(1) Investigate the problem and determine the formula to be used.
(2) Make a program for the formula (Programming).
(3) Store the program in the calculator (Writing in).
(4) Use the stored program to make the calculation (Program calculation).

EXAMPLE: To find the surface area and volume of a regular octahedron with 10 cm sides. With 7 cm sides? With 15 cm sides?



Length of side (a)	Surface area(s)	Volume(v)
10 cm	(346.4101614) cm²	(471.4045206) cm³
7	(169.740979)	(161.6917505)
15	(779.4228631)	(1590.990256)

(1) **Formula:** Surface area = S; Volume = V; Length of side = a.

Therefore, $S = 2\sqrt{3}a^2, V = \frac{\sqrt{2}}{3}a^3$

(2) **Programming:**

a. Each item of the formula will correspond to data memory number ①~⑨,⓪.

Surface area S in memory①
Volume V in memory②
Length of one side a in memory③; therefore
$S = 2\sqrt{3}a^2$ becomes①= 2 x $\sqrt{3}$ x③x③;
and, $V = \frac{\sqrt{2}}{3}a^3$ becomes②= $\sqrt{2}$ ÷ 3 x③x③x③.

These formulas can be used without change in the program, except that K must be placed before the constants 2 and 3 stored. Consequently, the program will be:

1 = K2 x K3√ x 3 x 3 :

(① (S) is the constant 2 multiplied by the square root of 3 and this multiplied twice by③(a).)

2 = K2√ ÷ K3 x 3 x 3 x 3 :

(②(V) is the square root of the constant 2 divided by the constant 3 and this multiplied by③(a) three times.)

b. The data to be input into the formula are:
the length of one side a, so write:
ENT 3 : (input is③(a).)

* "**ENT** n_1 : n_2 :..." are called data input message (ENT message).

c. Which are the answers to the calculations?
The answers are S (surface area) and V (volume) so write:

ANS 1 : 2 : (answers are①(S) and②(V).)

* "**ANS** n_1 : n_2 :.." are answer display messages (ANS messages).

The basic programming sequence is:

1. ENT message
2. Calculation formula
3. ANS message.

When we place the above programs in correct sequence for programming we get:

ENT 3 :
1 = K 2 x K 3√ x 3 x 3 :
2 = K 2√ ÷ K 3 x 3 x 3 x 3 :
ANS 1 : 2 :

(3) Writing in programs :

To write a program into the calculator:

1. Set the program switch at <WRITE>.
2. Key-in the program in correct sequence.

(To erase a previously stored program, press AC after setting at <WRITE>.)

OPERATION	READ-OUT	REMARK
Program switch at <WRITE>	0.	
AC	000	(Progarm clear, O displayed).
ENT	001 E2	(Step No.1, ENT)
3	002 E2 3	(Step No.2, value 3)
:	003 E2 3 Γ5	(Step No.3, :)

OPERATION	READ-OUT	REMARK
1	004 3 Γ5 1	(Step No.4, value 1)
−	005 Γ5 1 E0	(Step No.5, −)
K	006 1 E0 L2	(Step No.6, K)
2	007 E0 L2 2	(Step No.7, value 2)
×	008 L2 2 E5	(Step No.8, ×)
K	009 2 E5 L2	(Step No.9, K)
3	010 E5 L2 3	(Step No.10, value 3)
√	011 L2 3 ⊢1	(Step No.11, √)
×	012 3 ⊢1 E5	(Step No.12, ×)
3	013 ⊢1 E5 3	(Step No.13, value 3)
×	014 E5 3 E5	(Step No.14, ×)
3	015 3 E5 3	(Step No.15, value 3)
÷	016 E5 3 Γ5	(Step No.16, ÷)
2	017 3 Γ5 2	(Step No.17, value 2)
−	018 Γ5 2 E0	(Step No.18, −)
K	019 2 E0 L2	(Step No.19, K)
2	020 E0 L2 2	(Step No.20, value 2)
√	021 L2 2 ⊢1	(Step No.21, √)
+	022 2 ⊢1 E6	(Step No.22, +)
K	023 ⊢1 E6 L2	(Step No.23, K)
3	024 E6 L2 3	(Step No.24, value 3)
×	025 L2 3 E5	(Step No.25, ×)
3	026 3 E5 3	(Step No.26, value 3)
×	027 E5 3 E5	(Step No.27, ×)
3	028 3 E5 3	(Step No.28, value 3)
×	029 E5 3 E5	(Step No.29, ×)
3	030 3 E5 3	(Step No.30, value 3)
÷	031 E5 3 Γ5	(Step No.31, ÷)
ANS	032 3 Γ5 E3	(Step No.32, ANS)
1	033 Γ5 E3 1	(Step No.33, value 1)
÷	034 E3 1 Γ5	(Step No.34, ÷)
2	035 1 Γ5 2	(Step No.35, value 2)
÷	036 Γ5 2 Γ5	(Step No.36, ÷)

This completes the programming.

* In the <WRITE> mode, each time a key is pressed, that command is stored in the memory as a program. The number of step and the code number of the command written in, together with the code number of the previous

two commands, are displayed simultaneously in the read-out.

(4) Program calculation :

To perform a calculation using the program:

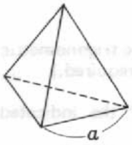
1. Set the program switch at <COMP> position.
   (When using a program including trigonometric or inverse trigonometric functions you must also set the angular mode selector as required.)
2. Press the STA key.
3. When the ENT lamp lights, input the required data for the indicated memory number and press ENT .
4. The answer is displayed by the memory number when the ANS lamp lights. (After reading out the answer, press ANS or ENT to advance the program.)
5. Press STA if desired to repeat the program calculation.
6. Press AC to stop the program calculation.

OPERATION	READ-OUT	REMARK
Program switch at <COMP> STA	ENT 3 0.	
	└→ Memory③(α) input	
10 ENT	ANS 1 346.4101614	
	└→ Memory①(S) answer	
(To advance the program) ANS	ANS 2 471.4045206	
	└→ Memory②(V) answer	
(To repeat the calculation) STA	ENT 3 0.	(Memory ③ input)
(If one side = 7cm) 7 ENT	ANS 1 169.740979	(Memory ① answer)
ANS	ANS 2 161.6917505	(Memory ② answer)
STA	ENT 3 0.	
(If one side = 15 cm) 15 ENT	ANS 1 779.4228631	
ANS	ANS 2 1590.990256	
(Program calculation completed) AC	0.	

* In program calculations, one advances in accordance with the lamps and the memory numbers displayed in the 1st read-out part to find the solution to the problem. The program can be advanced using either the ENT or the ANS keys.

2-5. HOW TO CHEK,ERASE AND CHANGE PROGRAMS

EXAMPLE: To find the surface area and volume of a regular tetrahedron with sides 10cm long. If sides are 7.5cm? If sides are 20cm?



Length of side (a)	Surface area(s)	Volume(v)
10 cm	(173.2050807) cm^2	(117.8511301) cm^3
7.5	(97.42785789)	(49.71844553)
20	(692.8203228)	(942.8090412)

(1) Formula: Surface area = S; Volume = V; Length of side = a; therefore, $S=\sqrt{3}a^2$, $V=\frac{\sqrt{2}}{12}a^3$

(2) Programming: S in memory①; V in memory②; a in memory③; therefore,

ENT 3 :
1 = K3√ x 3 x 3 :
2 = K2√ ÷ K12 x 3 x 3 x 3 :
ANS 1 : 2 :

This program may be written into the calculator from the beginning but it closely resembles the octahedron program on page 17 so that program can be revised if the methods of program checking, erasing, correction, etc., are understood.

(3) Program check and changes:

Program check is recalling the program written into the program memories to the display to confirm the contents. Each time the CHK key is pressed in the <WRITE> mode, the step numbers and contents are displayed one at a time, just as when the program was written.

The required steps are displayed in program check also to make use, erase or change previously stored programs.

If we compare the two programs:

A. Erase the 2nd item "K2 x" of the octahedron program.

B. Change the octahedron program from the 3rd item "K2√ ÷ K".

OPERATION	READ-OUT	REMARK
Program switch at <WRITE>	0.	
CHK	001 E2	(Step No.1, ENT)
CHK	002 E2 3	(Step No.2, value 3)
CHK	003 E2 3 Γ5	(Step No.3, :)
CHK	004 3 Γ5 1	(Step No.4, value 1)
CHK	005 Γ5 1 E0	(Step No.5, =)
CHK	006 1 E0 L2	(Step No.6, K)

→ Erase this.

OPERATION		READ-OUT	REMARK
Erase 1 step.	C	005 Γ5 1 E0	↳Back up the program with C . Γ5 (÷) displayed.
	CHK	006 1 E0 00	↳ Step No.6, no command
	CHK	007 E0 00 2	(Step No.7, value 2)
	CHK	008 00 2 E5	(Step No.8, ×)
Erase 2 steps.	C	007 E0 00 2	(Step back)
	C	006 1 E0 00	(Step back)
	CHK	007 E0 00 00	(Step No.7, no command)
	CHK	008 00 00 00	(Step No.8, no command)

K , 2 and × commands that were in step 6, 7 and 8 are now erased.

	CHK	009 00 00 L2	(Step No.9, K)
	CHK	010 00 L2 3	(Step No.10, value 3)

Next, advance the program to K 2 √ + by depressing CHECK .

	CHK	022 2 ⊢1 E6	(Step No.22, +)
	CHK	023 ⊢1 E6 L2	(Step No.23, K)

Next, write in the changed section.

	1	024 E6 L2 1	(Step No.24, value 1)
	2	025 L2 1 2	(Step No.25, value 2)
	×	026 1 2 E5	(Step No.26, ×)
	3	027 2 E5 3	(Step No.27, value 3)
	×	028 E5 3 E5	(Step No.28, ×)
	3	029 3 E5 3	(Step No.29, value 3)
	×	030 E5 3 E5	(Step No.30, ×)
	3	031 3 E5 3	(Step No.31, value 3)
	÷	032 E5 3 Γ5	(Step No.32, ÷)
	ANS	033 3 Γ5 E3	(Step No.33, ANS)
	1	034 Γ5 E3 1	(Step No.34, value 1)
	÷	035 E3 1 Γ5	(Step No.35, ÷)
	2	036 1 Γ5 2	(Step No.36, value 2)
	÷	037 Γ5 2 Γ5	(Step No.37, ÷)
(Look at the start of the program)	CHK	038 2 Γ5 00	(Step No.38, no command)
(Look at the start of the program)	CHK	039 Γ5 00 00	(Step No.39, no command) No parts of the program are displayed as no command (00).

This completes the changing of the program.

Making a program check :

1. Set the program switch at <WRITE>.
2. Each time CHK is depressed, step numbers and command codes are displayed for confirmation. (Steps containing no command are displayed as "00" or blank.)

* In order to make an addition to a program already written in, erase the command in the step where the addition is to be made and write in the new command by pressing the proper keys. Previous programs can be used when steps are erased or the number of steps is reduced but when the number of steps is increased the point from where the addition starts to the end must be written in again.

How to erase and change programs :

1. Advance to the required step using program check.
2. Display the command to be erased and press C . (Using C , at the time the program is backed up the command that leaves the display is erased and becomes no command.)
3. Display the step just before the one to be changed and then write in the altered program.
4. For corrections, display the step just before the one to be corrected and write in the new command.

* Program calculations except the message (IF M = m : A : B : C :) are not affected even if erased steps are in the middle of a program.

(4) Program calculation

OPERATION	READ-OUT	REMARK
Program switch at <COMP> STA	ENT 3 0.	(Memory③(α) input)
10 ENT	ANS 1 173.2050807	(Memory①(s) answer)
ANS	ANS 2 117.8511301	(Memory②(V) answer)
(To repeat the calculation) STA	ENT 3 0.	(Memory③ (α) input)
	The following is omited.	

2-6. PROGRAMMING RULES

* There are basic programming rules. Calculations cannot be performed if these rules are not observed.

1) Data input message (ENT message)

- The format is ENT M1 : M2 : M3 : ......
- M_1 , M_2 , M_3 , etc., are memory numbers 1 ~ 9 , 0 and I, IM.
- The sequence of memory numbers and I, IM are not determined.
- When inputting data into the memory, the previously stored values are erased and the new values entered.

2) Calculation message

- The format is [M1] [=] [M2] [+] …… M_1 must be followed by [=] always (the answer memory number). [=] cannot be used more than once (1 = 2 = 3 x 4 : etc.).
- In function calculations, the function command must be written in after the function data memory number, as in [M1] [=] [M2] [sin] [:].
- M_1, M_2, etc., are the same as ENT messages. Constants can also be used after [=] .
- Calculations are made from the left of [=] to the right, regardless of the importance of [+] [−] [×] [÷] in the calculation message and the answer is put into the memory before [=] .

  Example: When the memory contents are ②= 10.1, ③= 0.81, and ④= 8, (10.1 + 0.81) x 8 − 10.1 is calculated on the formula 1 = 2 + 3 x 4 − 2: and the answer 77.18 is put into memory①.
- [±] and function commands are calculated immediately and used in arithmetic operations. ([x^y] is the same as an arithmetic operation.)

  Example: When memory contents are ①= 1000, ②= 30, and ④= 2 (with "DEG"), $(\sin 30^\circ \times \log 1000)^{-2}$ is calculated on the formula 7 = 2 sin x 1 log x^y 4+/−: and the answer 0.444... is put into memory ⑦ .
- The contents of the memories used in the calculation are not changed, excluding the memory to the left of [=] .
- There is no limit to the length (number of steps) of a calculation message.
- Complex formulas can be broken down into several simpler formulas.

  Example: $x = \sqrt{\dfrac{(A{\times}B)+(C{\times}D)}{B+D}}$

  A in memory① D in memory④
  B in memory② x in memory⓪
  C in memory③

  If we take:
  (AxB) in memory⑤
  (CxD) in memory⑥
  (B+D) in memory⑦
  root in memory⑧
  Write

  5 = 1 x 2 :
  6 = 3 x 4 :
  7 = 2 + 4 :
  8 = 5 + 6 ÷ 7 :
  0 = 8√‾ :
- Any number of constants can be used in one calculation message but one constant is limited to a 10-digit value.
  (When [EXP] is used, the mantissa is 8 digits and the exponent 2 digits.)
- Constant calculations such as [+] [+] , [×] [×] can not be assembled in programs.
  When multiple calculation commands and function commands are assembled continuously, calculations are made in the same way as in manual calculations.

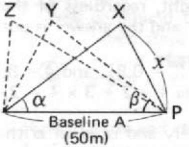
3) Answer display message

- The format is [ANS] [M1] [:] [M2] [:] [M3] [:] … .
- [M1] , [M2] , [M3] , etc., are the data memory numbers (①~⑨ , ⓪) and IM. (ANS I : is impossible.)
- The sequence for getting the answer is not determined.

* Next, we will make actual use of the above rules in making a program.

2-7. HOW TO USE THE MJ (MANUAL JUMP)

Example: In triangle mensuration, with a baseline of 50m, find the angles to points X, Y and Z. What are the distances in meters of PX, PY and PZ shown in the table below?



Point	Baseline A	Angle α	Angle β	Distance x
X	50 m	41°	76°	(36.815...)^{m}
Y	50	61°32′	49°25′	(47.066...)
Z	50	50°06′03″	37°53′	(38.382...)

(1) Formula:

$$x = \frac{Ax \sin \alpha}{\text{Sin}\ (180° - \alpha - \beta)}$$

(2) Memory: Baseline A in memory①
Angle α in memory②
Angle β in memory③
Angle $(180° - \alpha - \beta)$ in memory④
Distance x in memory⑤

(3) Programming:

```
ENT 1 : 2 : 3 :
4 = K180−2−3 :
5 = 1x2 sin ÷ 4 sin :
ANS 5 :
```

* In this program it is necessary to input the baseline length for each calcualtion.

```
   ENT 1 :
MJ
   ENT 2 : 3 :
   4 = K180−2−3:
   5 = 1x2 sin ÷ 4 sin :
   ANS 5 :
```

Make in this way.

(4) Program calculation :

OPERATION	RAED-OUT		REMARK
Program switch at <COMP>			
"DEG" STA	ENT 1	0.	(A input)
50 ENT	ENT 2	0.	(α input)
41 ENT	ENT 3	0.	(β input)
76 ENT	ANS 5	36.81561331	(Distance PX)
MJ	ENT 2	0.	(α input)

↳Jump to the MJ of the program with this key.

```
                  61[°'"]32[°'"][ENT]   | (ENT) 3                0. |
                  49[°'"]25[°'"][ENT]   | (ANS) 5      47.06613813 |  (Distance PY)
                             [STA]   | (ENT) 1                0. |
                               └→ [STA] is touched by error.
                              [MJ]   | (ENT) 2                0. |
                               └→ Jump to MJ with memory①as it is.
               50[°'"]6[°'"]3[°'"][ENT]   | (ENT) 3                0. |
                  37[°'"]53[°'"][ENT]   | (ANS) 5      38.38247737 |  (Distance PZ)
(Program calculation completed) [AC]   |                        0. |
```

* If the baseline changes input from the start with [STA], and if the baseline is the same input from the angle using [MJ].

> [MJ]
>
> 1. If [MJ] is pressed when a program is stopped (with (ENT), (ANS)) while performing a calculation, a jump is made up to the written in [MJ] (both backward and forward).
> 2. When more than one [MJ] are written in, the jump is made to the final one and the others are ignored.
> 3. If [MJ] is pressed when [MJ] is not written into the program, the program will not operate correctly.

2-8. PROGRAM TO FIND TOTALS (Σ)

Example: There are several triangles, the lengths of the sides of which we know. Find the area of each triangle and the total area.

Triangle	Side a	Side b	Side c	Area S
A	12^{m}	15^{m}	19^{m}	(89.977) m^2
B	19	18.5	25	(174.657)
C	25	14	21.3	(148.972)
			Total	(413.607)

(1) Formula:

$$S = \sqrt{S(S-a)\,(S-b)\,(S-c)}$$

$$S = \frac{a+b+c}{2}$$

(2) Memory:

a in the memory①　S in the memory④
b in the memory②　(S−a) in the memory⑨
c in the memory③　(S−b) in the memory⑧
S in the memory⓪　(S−c) in the memory⑦
values within the radical → ⑥

* In programs for totaling we indicate the memory used for totaling by M and the totaled memory as m, so we get M = M + m:.
This means that the M to which a new answer with m is added becomes a new M. 0 must always be placed before M. To make M into 0, assemble as M = K 0: or else use MAC . Thus, if we put the total area of the above calculation in memory⑤:

(3) Programming :

```
MAC
ENT 1 : 2 : 3 :
0 = 1 + 2 + 3 ÷ K2 :
9 = 0 − 1 :
8 = 0 − 2 :
7 = 0 − 3 :
6 = 0 x 9 x 8 x 7 :
4 = 6√ :
ANS 4 :
5 = 5 + 4 :
ANS 5 :
```

```
The program calculation with this program is :
<COMP>              STA
          side (a)  ENT
          side (b)  ENT
          side (c)  ENT  →  Area display
(To get the total-  ANS  →  Total area?
    ing program)            (ANS 5 : to be read)
(To calcuate the    STA
following triangle)  └→ The first MAC is read and
                        the total becomes 0.
                        (wrong program).
```

Here we vary the program sequence and use MJ .

```
  MAC
MJ
  ENT 1 : 2 : 3 :
   0 = 1 + 2 + 3 ÷ K2 :
   9 = 0 − 1 :
   8 = 0 − 2 ;
   7 = 0 − 3 :
   6 = 0 x 9 x 8 x 7 :
   4 = 6√ :
   5 = 5 + 4 :
  ANS 4 : 5 :
```

```
The calculation using this program is :
<COMP>              STA
          side (a) ENT
          side (b) ENT
          side (c) ENT →  Area display
                    ┌→When there is          (Repeat
One or the other    │ following        MJ ... from
                    │ triangle               side(a) ENT)
                    └→When giving              Total
                      total            ANS →   area
* When calculating another group of triangle,
                    STA
                    └──→ read  MAC ,
        and the total becomes 0.
```

How to perform totaling calculations

1). Prepare memory M for use in totaling. M = M+m: or M = m+M: is used. (m is the data or answer memory.)

2). This totaling use memory must be made 0 before starting the totaling.

3). How to make the totaling memory 0.

   a. Do not include a clear command in the program but press the MAC key before entering the first data of the calculation.

   b. A clear command such as MAC, etc., is included in the first part of the program and below the 2nd line MAC is not red. There are also methods using MJ or GOTO (GOTO is explained below.).

2-9. HOW TO USE GOTO and ST# (UNCONDITIONAL JUMP)

The program given above for finding the total surface area of triangles involves pressing MJ once for each triangle to make a jump to after MAC but, instead of pressing the key, if the GOTO command and ST# are programmed in front of the jump this can be done automatically without pressing MJ each time. This is an unconditional jump. By adding memory numbers after GOTO and ST# up to 10 jumps can be programmed.

- Altering the previous program.

```
┌─→ST#2: MAC
│┌→ST#1: ENT 1:2:3:
││       0=1+2+3÷K2:
││       9=0−1:
││       8=0−2:
││       7=0−3:
││       6=0x9x8x7:
││       4=6√ :
││       5=5+4:
││       ANS 4:
│└───────GOTO 1:
│To get
│total
│  ──→MJ ANS 5:
└────────GOTO 2:
```

* The calculation using this program is:

<COMP> STA

side(a) ENT side(b) ENT side(c) ENT → Area display

(To advance the program) ANS .... (Repeat from the input of side (a))

(When finding the total) MJ → Total area

(To advance the program) ANS .... (Repeat from the input of side (a))

* In this program calculation, press the keys in accordance with the indications of the displayed (ENT), (ANS) lamps. If the MJ key is considered the total key, anyone can understand the procedure and perform the calculations.

GOTO ST#

1). By putting in GOTO N: a jump can be made to the program ST#N:.

2). N is a natural number from 1 to 9, 0.

3). GOTON:, and ST#N: can be added at any position in the program. A maximum of 10 jumps can be used in accordance with the N number.

4). GOTO N1: is effective no matter how many times used but ST# N1: can only be used once. (When ST# N1: is used more than once, only the last is effective.)

5). When there is no ST# N1: to correspond to GOTO N1: the program calculation is stopped.

2-10. HOW TO USE IF (CONDITIONAL JUMP)

Example: To calculate the square root of the quadratic equation $ax^2 + bx + c = 0$. The way the answer is shown will differ depending on the formula used.

Problem	Coefficient			Answer
	a	b	c	
$8x^2 + 6x + 1 = 0$	8	6	1	(–0.25, –0.5)
$2x^2 - 28x + 98 = 0$	2	–28	98	(7)
$2x^2 + 26x + 89 = 0$	2	26	89	(–6.5 ± 1.5i)

1) Formula:

$$x = \frac{-b \pm \sqrt{D}}{2a}$$

$$D = b^2 - 4ac$$

2) Memory:

Coefficient a) in memory ①

b) in memory ②

c) in memory ③

Separate formula D in memory ④

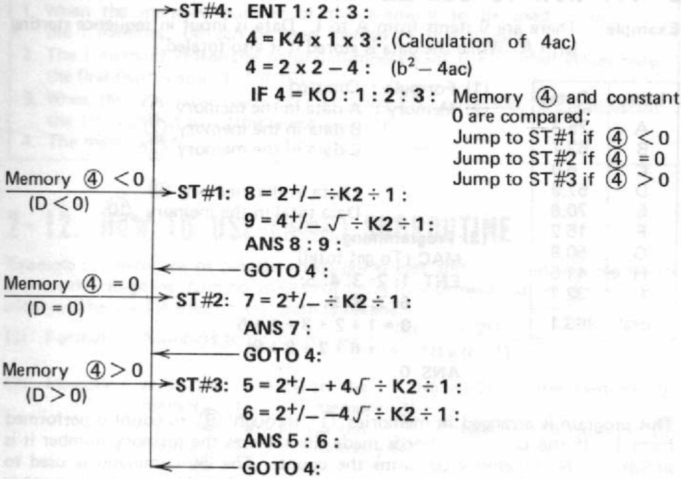
Answer (actual root) in memory ⑤, ⑥

Answer (compound root) in memory ⑦

Answer (imaginary root) { real number part in memory ⑧
imaginary number part in memory ⑨

* The conditional jump is jumping to a different place in accordance with the conditions after comparing the size of the values. IF and ST# are used. Conditional jump is IF M = m : A : B : C :. This compares M and m; when M is less than m the jump is to ST#A; when M = m the jump is to ST#B; and when M is larger than m the jump is to ST#C.

(3) Programming :



```
                 →ST#4: ENT 1 : 2 : 3 :
                        4 = K4 x 1 x 3 : (Calculation  of  4ac)
                        4 = 2 x 2 – 4 :   (b² – 4ac)
                        IF 4 = KO : 1 : 2 : 3 : Memory ④ and constant
                                                0 are compared;
                                                Jump to ST#1 if ④ <0
                                                Jump to ST#2 if ④ =0
Memory ④ <0                                     Jump to ST#3 if ④ >0
  (D<0)          →ST#1: 8 = 2+/– ÷K2 ÷ 1 :
                        9 = 4+/– √‾ ÷ K2 ÷ 1:
                        ANS 8 : 9 :
                 ←      GOTO 4 :
Memory ④ = 0
  (D = 0)        →ST#2: 7 = 2+/– ÷ K2 ÷ 1 :
                        ANS 7 :
                 ←      GOTO 4:
Memory ④>0
  (D>0)          →ST#3: 5 = 2+/– + 4√‾ ÷ K2 ÷ 1 :
                        6 = 2+/– –4 √‾ ÷ K2 ÷ 1 :
                        ANS 5 : 6 :
                 ←      GOTO 4:
```

(4) Program calculation :

<COMP> STA a ENT b ENT c ENT → actual root if ANS 5, ANS 6 light.
→ compound root if ANS 7 light.
→ imaginary root if ANS 8, ANS 9 light.

* In any case, the answer can be displayed merely by advancing the program. If advance is continued, a return is made to input of the coefficient a) with ENT 1.

IF

1. By programming IF M = m : A : B : C : , the place to which a jump is to be made can be made by comparing M and m. If M<m the jump is to ST#A; if M = m the jump is to ST#B; if M>m the jump is to ST#C.
2. M and m are data memory numbers, I, IM and constant. A, B and C are natural numbers 1~9,0. If M becomes a negative number when m is a positive constant or I, program IF m = M : A : B : C :.
3. When M≦m, A and B are the same values; when M≦m, B and C are the same values. A, B and C are written in.
4. A, B and C can be the same as the ST# for GOTO.
5. When calculations are performed without a jump destination, the program calculation is stopped.

2-11. HOW TO USE $\boxed{\text{I}}$ AND $\boxed{\text{IM}}$ (INDIRECT ADDRESS)

Example: There are 9 items from A to I. Data is input in sequence starting with A. While the data is stored it is also totaled.

Item	Data
A	25.3
B	63.7
C	6.0
D	57.9
E	70.6
F	15.2
G	50.8
H	41.5
I	32.1
Total	363.1

(1) Formula : Omitted

(2) Memory : A data in the memory ①
B data in the memory ②
C data in the memory ③
⋮
I data in the memory ⑨
Data total in the memory ⑩

(3) Programming :

```
MAC (To get total)
ENT 1: 2: 3: 4: 5:
    6: 7: 8: 9:
    0 = 1 + 2 + 3 + 4 + 5
          + 6 + 7 + 8 + 9:
ANS 0:
```

This program is arranged in memories ① through ⑨ so count is performed from 1. If the count number is made the same as the memory number it is simpler. The I-memory performs the count. The $\boxed{\text{IM}}$ command is used to employ the count number as the memory number. In other words, the number of the memory to be used is put into the I-memory and, during the calculation, $\boxed{\text{IM}}$ is used instead of that number.

For example, **I = K5:**
IM = 2 x 3: is the same as **5 = 2 x 3:**

Or, with **I = I + K1:** the $\boxed{\text{I}}$ memory counts 1 for each time the program is read. Therefore, the previous program is written as:

```
              MAC
┌─→ST#1:      I = I + K1:
│             ENT IM:
│             0 = 0 + IM:
└←────────────GOTO 1:
Total
→        MJ   ANS 0:
```

- All memories, including $\boxed{\text{I}}$, cleared with $\boxed{\text{MAC}}$.
- I = I + K1 : and (0 + 1) is put into the I-memory.
- Since 1 is in the I-memory, ENT IM : is the same as ENT 1 : and 0 = 0 + IM: is the same as 0 + 1 :.
- The 2nd time 1 is in the I-memory so I = I + K1 : is put into the I-memory as (1 + 1)
- I is 2 so ENT IM : is the same as NET 2 :. Also, 0 = 0 + IM : is the same as 0 = 0 + 2 :.
- The following is the same until $\boxed{\text{MJ}}$ is pressed.

$\boxed{\text{I}}$ and $\boxed{\text{IM}}$

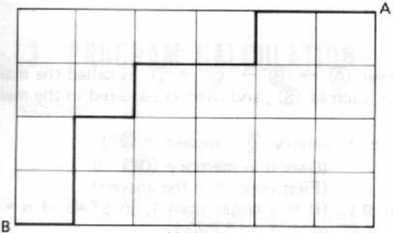
1. When the memory number of the memory to be used is put into the I memory, IM can be used instead of that number.
2. The I memory stores the natural numbers 1~9, 0. For other values only the first digit is stored (1 if 10).
3. When the calculator reads ENT I : , the ENT lamp and "E" light up on the 1st read-out part (this is not an error).
4. The message ANS I : can not be programmed.

2-12. HOW TO USE ($\boxed{\text{SUB\#}}$) SUBROUTINE

Example : There are m parallel roads going east and west and n going north and south. In going from northeast corner A to southwest corner B how many roads will be used if the shortest path is taken?

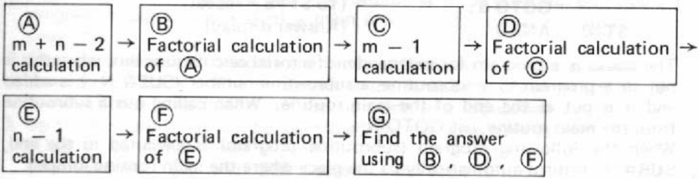
(1) Formula : Numbers to be assembled = $\dfrac{(m+n-2)!}{(m-1)!\times(n-1)!}$

(2) Memory : m in the memory ① (m−1)! in the memory ③ (m+n−2) in the memory ⑤
n in the memory ② (n−1)! in the memory ④ Answer in the memory ⑥

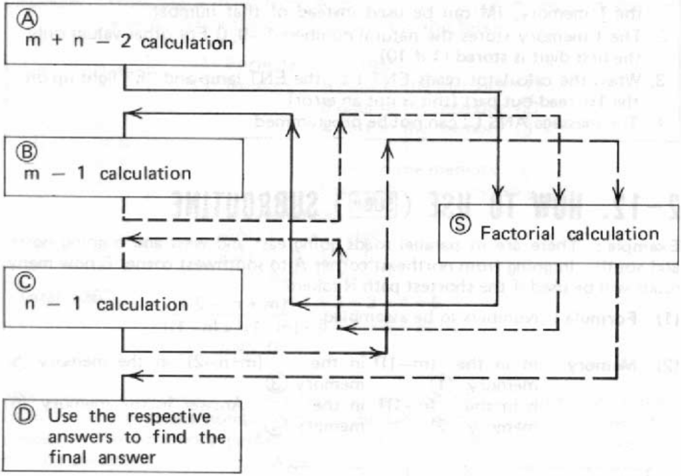


(210 reads in this diagram)

* This calculation is normally performed as follows.


Ⓐ m + n − 2 calculation
Ⓑ Factorial calculation of Ⓐ
Ⓒ m − 1 calculation
Ⓓ Factorial calculation of Ⓒ
Ⓔ n − 1 calculation
Ⓕ Factorial calculation of Ⓔ
Ⓖ Find the answer using Ⓑ, Ⓓ, Ⓕ

* The same calculation can be performed by making only the factorial calculations independent.



As shown in this diagram, the flow Ⓐ → Ⓑ → Ⓒ → Ⓓ is called the main routine. Parts made independent, such as Ⓢ , and used as required in the main routine are called subroutines.

Program to find the factorial n! (n → memory ⓪ , answer → ⑨)

```
          ENT 0  :              (Data n in memory ⓪ )
          9 = K1 :              (First enter 1 in the answer)
┌─►ST#8:  IF 0 = K1 : 0 : 0 : 9 :  (If n is larger than 1, to ST#9; if n = 1
│                               or n<1, to ST#0.)
│  ST#9:  9 = 9 x 0 :           (Multiply the answer by n)
│         0 = 0 − K1:           (n reduced by 1)
└─────────GOTO 8 :              (To ST#8 = repeat)
   ST#0:  ANS 9  :              (Answer display)
```

The above is a program for independent factorial calculations but, when this is put in a program as a subroutine, a subroutine number (SUB# N :) is added and it is put at the end of the main routine. When calling out a subroutine from the main routine, use GOTO N :.

When the following program (subroutine program) is executed to the end, SUB# N : returns automatically to the place where the main routine jumped.

(3) Programming :

```
┌→ST#2 :  ENT 1 : 2 :          (m and n input)
│         0 = 1 − K1 :         (m − 1 in ⓪)
│         GOTO 1 :             (To subroutine)
│         3 = 9 :              (Answer to m − 1 factorial in ③)
│         0 = 2 − K1 :         (n − 1 in ⓪)
│         GOTO 1 :             (To subroutine)
│         4 = 9 :              (Answer to n − 1 factorial in ④)
│         0 = 1 + 2 − K2 :     (m + n − 2 in ⓪)
│         GOTO 1 :             (To subroutine)
│         5 = 9 :              (m + n − 2 factorial in ④)
│         6 = 5 ÷ 3 ÷ 4 :      (Calculation of combined numbers)
│         ANS 6 :              (Displayofanswerofcombinednumber)
└─────────GOTO 2 :             (To start to repeat)
 SUB#1:  9 = K1 :
 ST#8 :  IF 0 = K1 : 0 : 0 : 9 :
 ST#9 :  9 = 9 x 0 :
         0 = 0 − K1 :
         GOTO 8 :
 ST#0 :
```

Subroutine program (Compared to the above, data is set by the main routine so ENT 0 : is taken. The answers to the respective factorials need not be seen so ANS 9 : is taken.)

2-13. PROGRAM CALCULATION

```
<COMP>  STA
     m  ENT
     n  ENT  → answer
        ANS
         └──→ (Repeat from m  ENT )
```

SUB#

1. When SUB# N : is placed at the beginning of a program and that program is put at the end of a main program, it is called a subroutine program.
2. When it is desired to execute a subroutine program in a main program, assemble GOTO N :. The N number is the same as the N of SUB# N :. A conditional jump (IF) can also be made to a subroutine.
3. Up to 10 subroutines can be assembled but the same number as the N in SUB# N : cannot be used.
(Consequently, ST# and SUB# together will total.10.)

4. When a subroutine program has been executed (go to the next SUB# N : or read to the 127th step), an automatic return is made to the step after the one from which the main program jumped. (This is not ST# N : .)
5. The destination of GOTO and IF jumps in a subroutine are in that subroutine. The destination of a GOTO of IF jump by a main routine cannot be in a subroutine.
6. A subroutine cannot be called by another subroutine.

* Special ways of using SUB# . . . . . . Using to make program additions.

* When a change is made in a program already stored in the calculator, the number of steps is increased (insert no command 00 when the number of steps is reduced), or an addition is made to the program, we can use SUB# to avoid re-entering all of the program after the changed part. This is especially convenient when changes are made near the start of a long program.

The method

1. Change to GOTO N : one line before or after the addition. (When this 1 line is 3 steps or more, enter 00 in the remainder.)
2. Write in the additional program at the end of the program, after SUB# N :
   Do not forget to enter the erased line at the beginning and end of the added part.

EXAMPLE:

```
ENT 1: 2: 3:          Add    ⎧ 6 = K2 x 1 x 2 x 3 cos:
0 = 1 + 2 + 3 ÷ K2:          ⎪ 7 = 2 x 2:
        ,                    ⎨ 8 = 1 x 1 + 7 – 6:
        ,                    ⎪ 3 = 8 √‾:
        ,                    ⎩ ANS 3:
   ANS 5:
   GOTO 2:
```

* Change or addition

1) Change **ENT 1: 2: 3:** to **GOTO 0:** and 4-step no command.
2) Write from **SUB# 0: ENT 1: 2: 3:** to added program ----6=to **ANS 3:** after the final **GOTO 2:** of the program.

Note: The above change is made when the angle included by two sides (angle in memory ③ is included in the area calculation. The other side is displayed after the calculation and then the area is displayed.

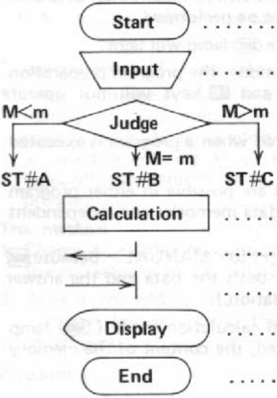
2-14. OPERATION OF THE PROGRAM SWITCH AT ⟨COMP⟩

1. When performing automatic calculations (program calculations) using a program stored in the calculator, the program switch must be set at⟨COMP⟩ STA is pressed; otherwise, calculations will not be performed.
2. For program calculations either the ENT or the ANS lamp will light.
3. When AC is pressed in program calculation mode, the program preparation mode will first be entered and the ENT and ANS keys will not operate. (The ENT and ANS lamps also go out.)
4. The program preparation mode is also entered when a program is executed breaking the programming rules.
5. Manual arithmetic and function calculations are possible in either program mode or program preparation mode but the data memories and independent memory cannot be used.

   (To see the contents of the data memories, set to <MANUAL>.Because MAC clears all the data memories in <COMP>, both the data and the answer are erased when it is pressed during a calculation.)
6. When an answer is displayed in the program calculation mode (ANS lamp lit), an entry is made and a program executed, the content of the memory number changes to the entered value.

2-15. WRITING FLOW CHARTS

Programs are easier to write when the sequence of the calculation is arranged clearly. This calculation sequence is called the flow chart and the symbols used are determined.



.... Program starts

.... Input the data

...... Write the ENT message.

.... Sizes of values compared . . . Write IF M = m : A : B : C :

.... Write the calculation message.

.... Where to jump unconditionally . . . Write GOTO N : .

.... Jump to here . . . Write ST# *n*: .

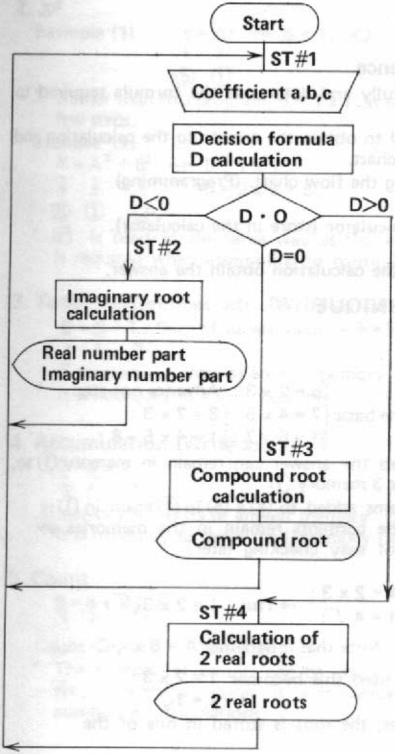
.... Display answer . . . . Write ANS message.

.... Program end

The following is the flow chart for the equation on page 28.

- Content of memories used

①	Coefficient a
②	Coefficient b
③	Coefficient c
④	Decision formula D
⑤ ⑥	Actual root
⑦	Compound root
⑧ ⑨	Imaginary root < real number part imaginary number part



```
ST#1 :
   ENT 1: 2: 3:

  4 = K4 x 1 x 3:
  4 = 2 x 2–4:

  IF 4= K0: 2: 3: 4:
ST# 2:

  8 = 2 +/–÷K2÷1:
  9 = 4 +/–√¯÷K2÷1:

  ANS 8: 9:

  GOTO I:

ST#3 :
  7 = 2 +/–÷K2÷1:

  ANS 7:

  GOTO I:

ST# 4:
  5 = 2 +/– +4 √¯÷ K2 ÷ 1:
  6 = 2 +/– –4 √¯÷ K2 ÷ 1:

  ANS 5 : 6:

  GOTO 1:
```

Note: Because the ST# go from above in sequence the contents are the same but this is not the samd as the program on page 29.

2-16. CONCLUSION

• Program calculation sequence

1. Investigate the problem carefully and determine the formula required to obtain the answer.
2. Clarify the procedure required to obtain the answer to the calculation and write this procedure in a flow chart.
3. Program the formula following the flow chart. (Programming)
4. Check the program.
5. Write the program into the calculator (store in the calculator).
6. Check the program for errors.
7. Use the program to perform the calculation obtain the answer.

• BASIC PROGRAM TECHNIQUE

1. Mixed calculation

Example (1).

$$\underset{\substack{\downarrow\\ \textcircled{1}}}{A} = (\underbrace{\underset{\substack{\downarrow\\ \textcircled{2}}}{B} \times \underset{\substack{\downarrow\\ \textcircled{3}}}{C}}_{\textcircled{6}}) + (\underbrace{\underset{\substack{\downarrow\\ \textcircled{4}}}{D} \times \underset{\substack{\downarrow\\ \textcircled{5}}}{E}}_{\textcircled{7}}) \rightarrow \text{The basic} \begin{cases} 6 = 2 \times 3: & \text{Advance one step} \\ 7 = 4 \times 5: & \begin{cases} 6 = 2 \times 3: \\ 1 = 4 \times 5 + 6: \end{cases} \\ 1 = 6 + 7: & \end{cases}$$

When we advance one step the answer can remain in memory ① so,

1 = 2 x 3: .... put 2 x 3 memory ①

1 = 4 x 5 + 1: ...... put items added to ④ x ⑤ in ① again in ①.

* In the basic form all the contents remain in the memories so there is the advantage of easy checking later.

Example (2).

$$\underset{\substack{\downarrow\\ \textcircled{1}}}{A} = \sqrt{\underbrace{\underset{\substack{\downarrow\\ \textcircled{2}}}{B} \times \underset{\substack{\downarrow\\ \textcircled{3}}}{C}}_{\textcircled{4}}} \rightarrow \text{The basic} \begin{cases} 4 = 2 \times 3: \\ 1 = 4\sqrt{\ }: \end{cases} \rightarrow \text{This} \quad 1 = 2 \times 3\sqrt{\ }:$$

Note that it becomes $A = B \times \sqrt{C}$.

Because memory ④ is not used this becomes **1 = 2 x 3 :**
1 = 1√ :.

* In extraction calculations, the root is stored in one of the memories.

* The same for finding functions of a calculation result ($\log\frac{123}{456}$, etc).

Example (3).

$$\underset{\substack{\downarrow\\ \textcircled{1}}}{A} = \frac{B \rightarrow \textcircled{2}}{\underbrace{\underset{\substack{\downarrow\\ \textcircled{3}}}{C} + \underset{\substack{\downarrow\\ \textcircled{4}}}{D}}_{\rightarrow \textcircled{5}}} \rightarrow \text{The basic} \begin{cases} 5 = 3 + 4: \\ 1 = 2 \div 5: \end{cases} \quad \text{Because memory } \textcircled{5} \text{ is not used.} \begin{cases} 1 = 3 + 4: \\ 1 = 2 \div 1: \end{cases}$$

* When the denominator is sum or difference this is put into the memory.

2. x^y

Example (1) $x = A^2$ ⇒ **5 = 1x^yK2:**

↓ ↓

⑤ ①

* Rather than this program, **5 = 1 x 1** : is easier to understand and there are few steps.

Example (2)

$x = A^2 + B^2$ ⇒ **5 = 1 x 1 :**

↓ ↓ ↓ **5 = 2 x 2 + 5 :**

⑤ ① ②

* $\boxed{x^y}$ is used in the same way as the arithmetic operations so caution is required when assembling in continued calculations.

3. Totals, differences, etc. (Write as Σ)

$\underline{9} = \underline{9} + \underline{1}$: Sum of square value → **9 = 1 x 1 + 9 :**

↖ ↗ ↖

Σ memory Data or answer memory

* It is necessary to make the Σ memory 0 in the beginning.

4. Accumulation (Write as Π)

$\underline{9} = \underline{9} \times \underline{1}$:

↖ ↗ ↖

Π memory Data or answer memory

* It is necessary to make the Π memory I in the beginning.

5. Count

$\underline{6} = \underline{6} + \underline{K1}$:

↖ ↗ ↖

Count memory Constant

* The number of counts can be checked with $\boxed{IF}$ in repeat calculation, etc., when the repeat is automatically stopped after the required number of calculations.

2-17. REFERENCE PROGRAMS

CASIO *fx-201P* PROGRAM SHEET

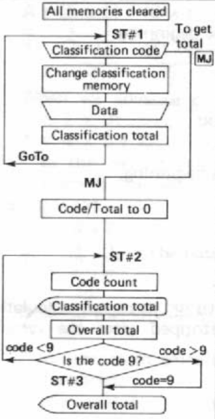
Program Name	Date	No.
Classification totals		1

Formula (examples using actual values are on the following page)

Input data in order and find total for a maximum of 9 classifications

Memory content	
① Code 1 total	⑥ Code 6 total
② Code 2 total	⑦ Code 7 total
③ Code 3 total	⑧ Code 8 total
④ Code 4 total	⑨ Code 9 total
⑤ Code 5 total	⓪ Memory exchange and overall total

Calculation flow chart



PROGRAM

ST#	MJ	Calculation message												Step
		MAC												
ST#1:														
		ENT	I	:										7
		0	=	IM	:									
		ENT	IM	:										14
		IM	=	IM	+	0	:							
		GoTo	1	:										23
	MJ													
		0	=	K	0	:								
		I	=	0	:									33
ST#2:														
		I	=	I	+	K	1	:						43
		ANS	IM	:										
		0	=	0	+	IM	:							52
		IF	I	=	K	9	:	2	:	3	:	3	:	
ST#3:														
		ANS	0	:										70

* Performing program calculations is explained on the next page.

CASIO *fx-201P* Program calculation operation manual

Calculation example. . . . actual calculation table, values, etc.

classification code	Data
3	1,850
1	3,100
2	2,000
9	3,600
4	6,120
2	1,450
8	3,880
5	2,230
3	5,360
5	4,870
6	3,190
7	2,310
1	2,500
7	1,960
8	3,300
5	1,250
4	1,890

➡

Code	Total
1	(5,600)
2	(3,450)
3	(7,210)
4	(8,010)
5	(8,350)
6	(3,190)
7	(4,270)
8	(7,180)
9	(3,600)
total	(50,860)

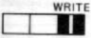
figures in parentheses are answers

Memory no.	When ENT lamp is lit (data)	When ANS lamp is lit (answer)
1	code 1 data	code 1 total
2	code 2 data	code 2 total
3	code 3 data	code 3 total
4	code 4 data	code 4 total
5	code 5 data	code 5 total
6	code 6 data	code 6 total
7	code 7 data	code 7 total
8	code 8 data	code 8 total
9	code 9 data	code 9 total
0		data total
E (I-memory)	classification code	

Preparation:

1. Write-in program

- Program switch



- Key-in in the sequence of the program

2. Program calculation

- Program switch

- The following in the sequence given at the right

Procedure	Lamp	Memory No.	Display meaning	Key operation	Remarks
1				START	
2	ENT	E	classification code input	3 ENT	code is 3 so ENT 3 light
3	ENT	3	data input	1850 ENT	
4	ENT	E	classification code input	1 ENT	code is 1 so ENT 1 light
5	ENT	1	data input	3100 ENT	
6	ENT	E	repeat the following from procedure 2		
7				When all data end, MJ	
8	ANS	1	code 1 total	ANS	
9	ANS	2	code 2 total	ANS	
10	ANS	3	code 3 total	ANS	
11	ANS	4	code 4 total	ANS	
12	ANS	5	code 5 total	ANS	
13	ANS	6	code 6 total	ANS	
14	ANS	7	code 7 total	ANS	
15	ANS	8	code 8 total	ANS	
16	ANS	9	code 9 total	ANS	
17	ANS	0	all data total	end of calculation AC	
18					
19					
20					
21					
22					
23					
24					
25					

CASIO *fx-201P* PROGRAM SHEET

Program name Prediction by regression analysis (primary regression)	Date	No. 2

Formula (examples using actual values are on the following page)

If the regressing straight line is

$$Y = A + Bx,$$

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

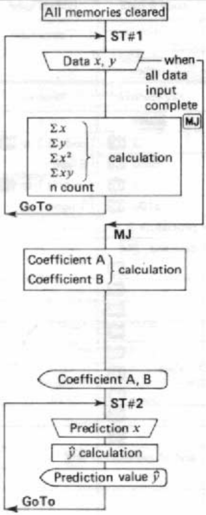
$$A = \frac{1}{n}(\Sigma y - B \cdot \Sigma x)$$

Predicted value $\hat{y}$ is $\hat{y} = A + Bx$

Memory content

①	data x	⑥	Σxy
②	data y	⑦	count n
③	Σx	⑧	determined coefficient A
④	Σy	⑨	determined coefficient B
⑤	Σx^2	⓪	

Calculation flow chart



PROGRAM

ST#	MJ	Calculation message												Step
		MAC												
ST#1:														
		ENT	1	:	2	:								9
		3	=	3	+	1	:							
		4	=	4	+	2	:							21
		5	=	1	x	1	+	5	:					
		6	=	1	x	2	+	6	:					37
		7	=	7	+	k	1	:						
		GoTo	1	:										47
	MJ													
		8	=	3	x	3	:							
		8	=	7	x	5	−	8	:					62
		9	=	3	x	4	:							
		9	=	7	x	6	−	9	÷	8	:			78
		8	=	9	x	3	+/−	+	4	÷	7	:		
		ANS	8	:	9	:								94
ST#2:														
		ENT	1	:										100
		2	=	1	x	9	+	8	:					
		ANS	2	:										
			2	:										114

* Performing program calculations is explained on the next page.

CASIO *fx-201P* Program calculation operation manual

Calculation example . . . actual calculation table, values, etc.

Year (x)	Sales amount (y)	(Unit $1,000)
1966	813.6	
1967	781.3	
1968	855.1	
1969	1,228.7	
1970	1,432.4	
1971	1,574.9	
1972	1,697.2	
1973	2,069.5	
1974	1,986.0	
1975	2,290.8	
1976	(2,451.2)	Prediction
1977	(2,629.1)	
1978	(2,806.9)	

$y = A + Bx$ A = (−6,619.9) . . . for 0 year
B = (177.86) . . . trend for each year

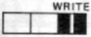
Memory no.	When ENT lamp is lit (data)	When ANS lamp is lit (answer)
1	year	
2	amount	predicted amount
3		
4		
5		
6		
7		
8		determined coefficient A
9		determined coefficient B
0		
E (I-memory)		

Preparation:

1. Write-in program

- Program switch

WRITE



- Key-in in the sequence of the program

2. Program calculation

- Program switch

COMP

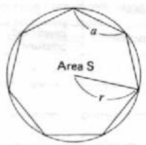
- The following in the sequence given at the right

Procedure	Lamp	Memory No.	Display meaning	Key operation	Remarks
1				START	
2	ENT	1	year input	66 ENT	
3	ENT	2	amount input	813.6 ENT	
4	ENT	1	year input	67 ENT	
5	ENT	2	amount input	781.3 ENT	
6	ENT	1	repeat from procedure 2		
7				When all data end, MJ	
8	ANS	8	determined coefficient A display	ANS	
9	ANS	9	determined coefficient B display	ANS	
10	ENT	1	prediction year input	76 ENT	
11	ANS	2	prediction amount display	ANS	
12	ENT	1	prediction year input	77 ENT	
13	ANS	2	prediction amount display	ANS	
14	ENT	1	repeat from procedure 10		
15				end of calculation AC	
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					

CASIO *fx-201P* PROGRAM SHEET

Program name Area and length of one side of an n-sided regular polygon	date	No. 3

Formula



The length a) of one side and area S) of an insided regular polygon fitting inside a circle with radius r):

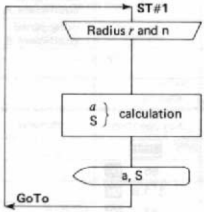
$a = 2r \cdot \sin\frac{\alpha}{2}$

$S = \frac{nr^2}{2} \sin\alpha$

$(\alpha = \frac{360°}{n})$

Memory content			
①	radius of circle r	⑥	
②	number of sides of polygon n n	⑦	
③	$\alpha\ (= \frac{360}{n})$	⑧	
④	length of one side a)	⑨	
⑤	area S) of regular polygon	⓪	

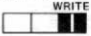
Calculation flow chart



PROGRAM

ST#	MJ	Calculation message											Step
ST#1:													
	ENT	1	:	2	:								
	3	=	K	3	6	0	÷	2	:				17
	4	=	3	÷	K	2	:						
	4	=	K	2	x	1	x	4	sin	:			34
	5	=	1	x	1	x	2	x	3	sin	÷		
	k	2	:										
	ANS	4	:	5	:								
	GoTo	1	:										56

Program calculation operation manual

Preparation	Procedure	Lamp	Memory No.	Display meaning	Key operation	Remarks
Preparation: **1. Write-in program** * Program switch WRITE  ** Key-in in the sequence of the program **2. Program calculation** * Program switch COMP * Angular mode selector: "DEG" * The following in the sequence given at the right	1				START	
	2	ENT	1	r input	10 ENT	radius 10 cm
	3	ENT	2	n input	7 ENT	for regular heptagon
	4	ANS	4	a display	ANS	a= 8.6776748 cm
	5	ANS	5	S display	ANS	S= 273.641018 cm²
	6	ENT	1	r input	12 ENT	radius 12 cm
	7	ENT	2	n input	8 ENT	for regular octagon
	8	ANS	4	a display	ANS	a= 9.18440232 cm
	9	ANS	5	S display	ANS	S=407.2935052cm²
	10	ENT	1	repeat form procedure 2		
	11				end of calculation AC	
	12					

CASIO *fx-201P* PROGRAM SHEET

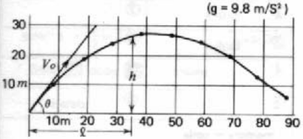
Program name	Parabolic motion (graph showing relationship of elevation and distance to time)	Date	4

Formula

$$h = V_0 t \cdot \sin\theta - \frac{1}{2}gt^2$$

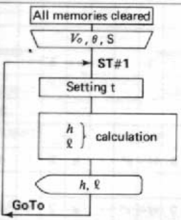
$$\ell = V_0 t \cdot \cos\theta$$

$(g = 9.8\ m/S^2)$



Memory content	
① initial velocity V_0 m/second	⑥ horizontal distance ℓm
② angle θ (degree)	⑦
③ time period S second	⑧
④ time elapsed t second	⑨
⑤ height *h*m	⓪

Calculation flow chart



PROGRAM

ST#	MJ	Calculation message											Step
	MAC												
	ENT	1	:	2	:	3	:						
ST#1:													11
	4	=	4	+	3	:							
	7	=	4	x	4	x	k	x	•	9	:		28
	5	=	1	x	4	x	2	sin	—	7	:		
	6	=	1	x	4	x	2	cos	:				48
	ANS	5	:	6	:								
	GoTo	1	:										56

Program calculation operation manual

Preparation:

1. Write-in program
   - Program switch WRITE
   - Key-in in the sequence of the program
2. Program calculation
   - Program switch COMP
   - Angular mode selector: "DEG"
   - The following in the sequence given at the right

Procedure	Lamp	Memory No.	Display meaning	Key operation	Remarks
1				START	
2	ENT	1	V₀ input	30 ENT	initial velocity 30m/s angle 50°, time period 0.5 sec.
3	ENT	2	θ input	50 ENT	
4	ENT	3	time period S input	⊡ 5 ENT	
5	ANS	5	height after S seconds	ANS	h_1 = 10.2656666
6	ANS	6	distance after S seconds	ANS	l_1 = 9.64181415
7	ANS	5	height after 2S seconds	ANS	h_2 = 18.0813332
8	ANS	6	distance after 2S seconds	ANS	l_2 = 19.2836283
9	ANS	5	height after 3S seconds	ANS	h_3 = 23.4469998
10	ANS	6	distance after 3S seconds	ANS	l_3 = 28.92544245
11	ANS	5	repeat from procedure 5		
12				end of calculation AC	

CASIO *fx-201P* PROGRAM SHEET

Program name	date	No.
Hyperbola function		5

Formula

$$\sinh x = \frac{e^x - e^{-x}}{2} \quad \ldots\ldots\ldots\ldots \quad \text{code } 1$$

$$\cosh x = \frac{e^x + e^{-x}}{2} \quad \ldots\ldots\ldots\ldots \quad \text{code } 2$$

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \ldots \quad \text{code } 3$$

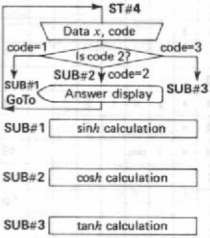
- Data ENT code ENT → answer

Memory content

①	sinh x	⑥	
②	cosh x	⑦	
③	tanh x	⑧	
④		⑨	
⑤		⓪	data x

I-memory → code

Calculation flow chart



PROGRAM

Calculation message

ST#	MJ												Step
ST#4:													
	ENT	0	:	I	:								
	IF	I	=	k	2	:	1	:	2	:	3	:	
	ANS	IM	:										
	GoTo	4	:										26
SUB#1:													
	1	=	0	e^x	−	0	+/−	e^x	÷	k	2	:	41
SUB#2:													
	2	=	0	e^x	+	0	+/−	e^x	÷	k	2	:	56
SUB#3:													
	9	=	0	e^x	+	0	+/−	e^x	:				
	3	=	0	e^x	−	0	+/−	e^x	÷	9	:		79

Remarks: 1/x can not be put into program.

Program calculation operation manual

Preparation :

1. **Write-in program**
   - Program switch WRITE
   - Key-in in the sequence of the program
2. **Program calculation**
   - Program switch COMP
   - The following in the sequence given at the right

Procedure	Lamp	Memory No.	Display meaning	Key operation	Remarks
1				START	
2	ENT	0	data x input	1.2 ENT	answer is
3	ENT	E	code input	1 ENT	1.509461345 for
4	ANS	1	sin hx input	ANS	sin$h$ 1.2
5	ENT	0	data x input	2.5 ENT	answer is
6	ENT	E	code input	2 ENT	6.132289499 for
7	ANS	2	cos hx display	ANS	cos$h$ 2.5
8	ENT	0	data x input	· 9 ENT	answer is
9	ENT	E	code input	3 ENT	0.716297868 for
10	ANS	3	tan h display	ANS	tan$h$ 0.9
11	ENT	0	repeat from procedure 2		
12				end of calculation AC	

CASIO *fx-201P* PROGRAM SHEET

Program name	Date	No.
Inverse hyperbola function		6

Formula

$\sinh^{-1} x = \ln(x + \sqrt{x^2 + 1})$ ..... code 1

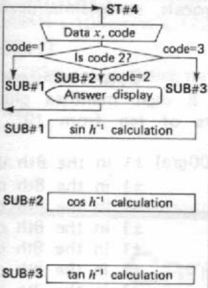
$\cosh^{-1} x = \ln(x + \sqrt{x^2 - 1})$ ..... code 2 $[x > 1]$

$\tanh^{-1} x = \frac{1}{2} \ln \frac{1+x}{1-x}$ ..... code 3 $[|x| < 1]$

- Data ENT code ENT → Answer

Memory content			
①	sin h^{-1} x	⑥	
②	cos h^{-1} x	⑦	
③	tan h^{-1} x	⑧	
④		⑨	
⑤		⓪	data x
I-memory → code			

Calculation flow chart



PROGRAM

ST#	MJ				Calculation message								Step
ST#4:	ENT	0	:	I	:								
	IF	I	=	k	2	:	1	:	2	:	3	:	
	ANS	IM	:										
	GoTo	4	:										26
SUB#1:	9	=	0	x	0	+	k	1	:				
	9	=	9	√	+	0	:						
	1	=	9	ln	:								50
SUB#2:	9	=	0	x	0	−	k	1	:				
	9	=	9	√	+	0	:						
	2	=	9	ln	:								72
SUB#3:	9	=	k	1	−	0	:						
	9	=	k	I	+	0	÷	9	:				
	3	=	9	ln	÷	k	2	:					101

Program calculation operation manual

Preparation:

1. **Write-in program**
   - Program switch WRITE
   - Key-in in the sequence of the program
2. **Program calculation**
   - Program switch COMP
   - The following in the sequence given at the right

Procedure	Lamp	Memory No.	Display meaning	Key operation	Remarks
1				START	
2	ENT	0	data x input	1.5 ENT	answer is
3	ENT	E	code input	1 ENT	1.1947632
4	ANS	1	sin h^{-1} x display	ANS	for sinh^{-1} 1.5
5	ENT	0	data x input	5.3 ENT	answer is
6	ENT	E	code input	2 ENT	2.3518328 for
7	ANS	2	cos h^{-1} x display	ANS	cosh^{-1} 5.3
8	ENT	0	data x input	· 6 ENT	answer is
9	ENT	E	code input	3 ENT	0.6931472 for
10	ANS	3	tan h^{-1} x display	ANS	tanh^{-1} 0.6
11	ENT	0	repeat from procedure 2		
12				end of calculation AC	

Specifications

NORMAL OPERATION

Capabilities:

4 basic functions, chain & mixed operations, constant calculations for five functions, powers and reciprocals, automatic accumulation in four functions, direct access to the memory, true credit balance and various kinds of practical calculations.

SCIENTIFIC FUNCTION

Trigonometric/Inverse trigonometric functions, common/natural logarithmic functions, Exponentiations, square roots, reciprocals, sexagesimal/decimal conversion, Pi entry and scientific notation.

Capacity :

	Input range	Output accuracy
Entry/basic operations	10 digit mantissa or 8 digit mantissa plus 2 digit exponent (powers of ten from 10^{99} to to 10^{-99}).	
$\sin x$ /$\cos x$ /$\tan x$	$\mid x \mid \leqq 1440°$ (8πrad,1600gra)	±1 in the 8th digit
$\sin^{-1} x$ /$\cos^{-1} x$	$\mid x \mid \leqq 1$	±1 in the 8th digit
$\tan^{-1} x$	$\mid x \mid < 1 \times 10^{100}$	±1 in the 8th digit
$\log x$ /$\ln x$	$0 < x < 1 \times 10^{100}$	±1 in the 8th digit
10^x	$\mid x \mid < 100$	±1 in the 8th digit
e^x	$\mid x \mid \leqq 230$	±1 in the 8th digit
x^y	$0 < x < 1 \times 10^{100}$	±1 in the 7th digit
$\sqrt{x}$	$0 \leqq x < 1 \times 10^{100}$	±1 in the 10th digit
$1/x$	$\mid x \mid < 1 \times 10^{100}$, $x \neq 0$	±1 in the 10th digit
o ′ ″	Up to second	±1 in the 10th digit
π	10 digit	

DECIMAL POINT Full floating mode with underflow

NEGATIVE NUMBER

Indicated by the floating minus (–) sign for mantissa.
The minus sign appears in the 3rd column for a negative exponent.

OVERFLOW Indicated by an *"E "* sign, locking the calculator.

MEMORY 1 independent memory and 10 data memories.

PROGRAM

Number of steps: 127 steps, stored system

Memory: 10 memories for calculation and data totaling plus 1 indirect address memory

Conditional and unconditional jump: max. of 10 jumps possible

Subroutines: max of 10 subroutines, 1 deep

Other functions: Manual jump, multiple assembly of one constant, program writing and check command display, and back-step.

READ-OUT

Zero suppression, Digitron tube panle, and LED for signs

POWER CONSUMPTION 0.6 W

POWER SOURCE

AC: 100, 117, 220 or 240V (±10V), 50/60 Hz with applicable AC Adaptor

DC: Four AA size Manganese dry batteries (SUM–3) operate about 8 hours continuously.

Four AA size Alkaline dry batteries (AM–3) operate about 19 hours continuously.

USABLE TEMPERATURE

0°C ~ 40°C (32°F ~ 104°F)

DIMENSIONS:

34.3mmH x 104mmW x 172mmD (1-3/8"H x 4"W x 7"D)

WEIGHT

364 g (12.8 oz) including batteries

Care of your new electronic calculator

This calculator is a durable, precision-made instrument which will provide you with years of trouble free service.

To help ensure this we recommend that the inside of the calculator not be touched. It is also inadvisable to subject the calculator to hard knockes, and unduly strong key pressing.

Extreme cold (below 32°F or 0°C), heat (above 104°F or 40°C) and humidity may also effect the function of the calculator. When you do not use the calculator for a long period, take out the batteries to prevent possible damage from battery leakage. Special care should be taken not to leave dead batteries inside the calculator. Please make sure you switch off the power when you finish your calculations or intend to open the cover to change batteries. Should the calculator need servicing, take the unit to the store where purchased or to a nearby dealer.

CASIO®

Printed in Japan