NATIONAL SEMICONDUCTOR 4615

CONTENTS

Getting Started

Turn your calculator on with the switch on the left side of the machine. The calculator is automatically cleared and the display should now show 0.
If it does, not, check to see of the battery needs recharging by connecting the AC charger.

AC Charger

Your calculator is powered by rechargeable NiCd batteries which should give you about eight hours use with normal operation. To charge your batteries, connect the AC charger to the jack on top of the calculator. A typical full charge takes five hours.
It is recommended that you plug your machine in to charge each night. You can operate your calculator with the charger connected and the calculator will not overcharge. BE SURE CALCULATOR IS TURNED OFF BEFORE CONNECTING THE CHARGER.

Operations

Display
Your calculator will accept and display any positive or negative number between 0.0000001 and 99999999. Any result larger than 99999999 or smaller than −99999999 will result in an overflow indicated by all zeros and decimal points being displayed.
Note: *See Calculations with Large Numbers section.*

Automatic Display Shutoff
To save battery life, your calculator automatically shuts off the display and shows all decimal points if no key has been touched for approximately 25 seconds. No data has been changed and further entries or operations will bring back the display. To restore the display without changing its contents, touch CHS twice.

Reverse Polish Logic and the Stack Principle

Your calculator uses Reverse Polish logic with three registers called X, Y and Z. A register is an electronic element used to store data while it is being displayed, processed or waiting to be processed. The three registers are arranged in a "stack" as follows: (To avoid confusion between the name of a register and its contents, the registers in this diagram and the diagrams in Appendix A are represented by capital letters X, Y and Z and the contents of the registers by lowercase letters x, y and z).

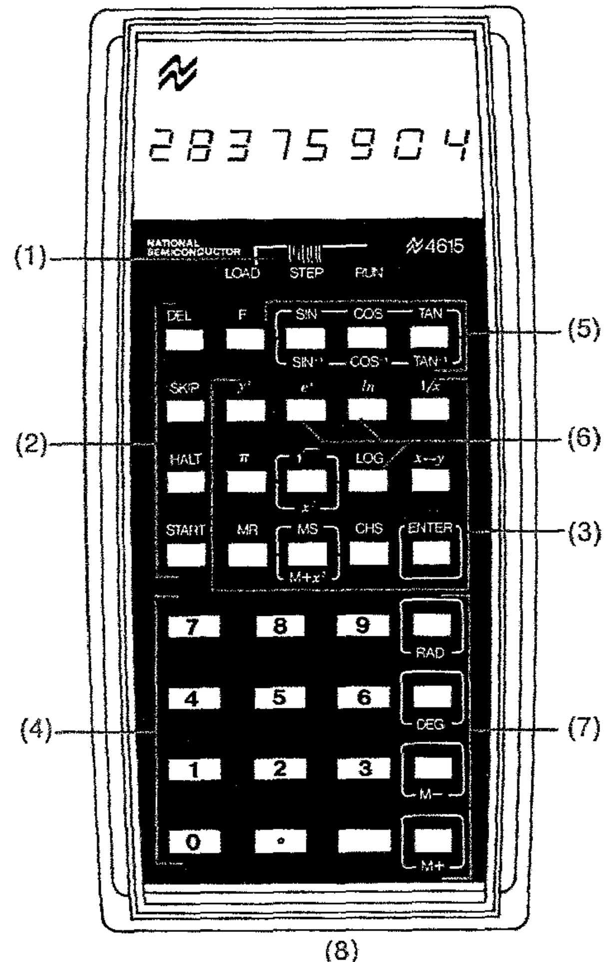
CONTENTS	LOCATION
z	Z
y	Y
x	X

The display always shows the contents (x) of register X. See Appendix A for diagrams showing what happens to the stack for each operation on the calculator.

Keyboard Layout

(1) LOAD/STEP/RUN switch
(2) Program control keys
(3) Special mathematical functions and memory keys
(4) Number entry keys
(5) Trig function keys
(6) Logarithmic function keys.
(7) Basic function, radian degree conversion, accumulating memory keys
(8) Clear key

Keyboard Layout



Keyboard Callout

F — Accesses lower functions on these keys: ($\sin^{-1}$) ($\cos^{-1}$) ($\tan^{-1}$) (x^2) ($M+x^2$) (rad) (deg) (M+) (M−)

sin — Upper function: Computes the sine of the angle in the display.

($\sin^{-1}$) — Lower function: Computes the inverse sine (arc sine) of the number in the display.

cos Upper function: Computes the cosine of the angle in the display.

($\cos^{-1}$) Lower function: Computes the inverse cosine (arc cosine) of the number in the display.

tan Upper function: Computes the tangent of the angle in the display.

($\tan^{-1}$) Lower function: Computes the inverse tangent (arc tangent) of the number in the display.

y^x Raises "y" to the "x" power.

e^x Computes the natural antilogarithm of the number in the display (raises e = 2.718281 to the "x" power).

ln Computes the natural logarithm of the number in the display.

1/x Computes the reciprocal of the number in the display (divides 1 by "x").

π Enters Pi (π) = 3.1415926 into the display.

$\sqrt{}$ Upper function: Computes the square root of the number in the display.

(x^2) Lower function: Squares the number in the display.

log Computes the common logarithm of the number in the display.

x–y Exchanges the number now in the display with the number previously in the display.

MR Recalls the contents of memory to the display.

MS Upper function: Stores the number in the display in memory.

(M+x^2) Lower function: Adds the square of the number in the display to the contents of memory.

CHS Changes the sign of the number in the display.

ENT Enters the number in the display into a working register ("y").

÷ Upper function: Divides "y" by "x."

(rad) Lower function: Converts the number of degrees in the display to radians.

× Upper function: Multiplies "y" by "x."

(deg) Lower function: Converts the number of radians in the display to degrees.

− Upper function: Subtracts "x" from "y."

(M−) Lower function: Subtracts the number in the display from the contents of memory.

+ Upper function: Adds "x" to "y."

(M+) Lower function: Adds the number in the display to the contents of memory.

Keying In and Entering Numbers

To enter the first number in a 2-factor calculation, key in the number and touch ENT. If your number includes a decimal point, key it in with the number. If a decimal is keyed in more than once in a number entry, the calculator will use the last decimal keyed in. You do not have to key in the decimal in whole numbers.

To enter a negative number, key in the number and touch CHS.

Correcting Wrong Number Entries

To clear a wrong number entry, touch C. Touching C clears the X register (display) and drops the stack down.

Performing Calculations

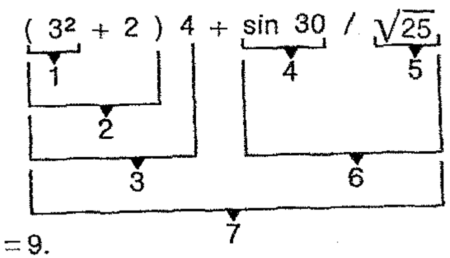
In addition to the separate memory, there are three locations where numbers can be kept and operated on. These locations are called registers and in your calculator these have been combined into an automatic stack. Your calculator uses the three level stack along with Reverse Polish logic to enable you to perform calculations according to mathematical hierarchy.

Mathematical Hierarchy and Reverse Polish Logic

"Hierarchy" is a term for the rules of mathematics which tell you in which order to perform operations on numbers. Those rules are:

1. Try to do the problem left to right (this may not always be possible).
2. Do all operations within parentheses, if any, first.
3. Perform operations in the following order:
   a. raising to powers, taking roots, trig, log and reciprocal functions.
   b. multiplication and division.
   c. addition and subtraction.
4. Repeat steps 1 through 3 until the calculation is complete.

Example: The equation $(3^2+2)4+\sin 30/\sqrt{25} = 44.1$ is solved according to the rules of hierarchy as follows:



1. $3^2 = 9.$
2. $9+2=11.$
3. $11 \times 4 = 44.$
4. $\sin 30 = .5$
5. $\sqrt{25} = 5.$
6. $.5 \div 5 = .1$
7. $44 + .1 = 44.1.$

If you remember the following three steps in applying Reverse Polish logic to the rules of hierarchy, you will quickly master your calculator and have confidence in its answers.

1. Starting at the left and working right, key in the next number (or the first if this is the beginning of a new problem).
2. Ask yourself: "Can an operation be performed according to the rules of hierarchy?" If so, perform all operations possible. If not, touch ENT.
3. Repeat steps 1 and 2 until your calculation is complete.

Example: Following these three steps, you can calculate the equation $(3^2+2)4+\sin 30/\sqrt{25}$ using Reverse Polish logic as follows:

KEY IN	DISPLAY SHOWS	COMMENTS
3	3	
F (x^2)	9.	3^2.
2	2	
+	11.	3^2+2.
4	4	
×	44.	$(3^2+2)4$.
30	30	
sin	.5	sin 30.
25	25	
√	5.	$\sqrt{25}$.
÷	.1	$\sin 30/\sqrt{25}$.
+	44.1	$(3^2+2)4+\sin 30/\sqrt{25}$.

Calculation is complete and performed according to the rules of hierarchy.

Double Functions

Some keys on your calculator have double functions; that is, touching the same key will perform two different operations. Double functions have their function defined in lettering in front of the keys. They will be referred to by parentheses ( ) in this manual. Primary functions will be referred to by a color bar

F — Accesses double function keys.
If F is touched accidentally, touch C to cancel the effect. The stack is not affected.

One-Factor Calculations

One-factor functions work directly on the number in the display. There is no need to touch ENT before performing the function.

Square, Square Root and Reciprocal Functions

(x^2) Touched after the F key, squares the number in the display.

$\sqrt{\ }$ Computes the square root of the number in the display.

1/x Computes the reciprocal of the number in the display.

Example: Key in 2 1/x ; display shows: .5.

Logarithmic Functions

ln Computes the natural logarithm of any positive number in the display.

e^x Computes the natural antilog of the number in the display by raising "e" (2.718281) to the power in the display.

log Computes the common logarithm of any positive number in the display.

To compute the common antilogarithm of the number in the display, key in: 10 x-y y^x.

Trigonometric Functions

sin Computes the sine of the angle (in degrees) in the display.

cos Computes the cosine of the angle (in degrees) in the display.

tan Computes the tangent of the angle (in degrees) in the display.

($\sin^{-1}$) Touched after F, computes the arc sine (in degrees) of the number in the display.

($\cos^{-1}$) Touched after F, computes the arc cosine (in degrees) of the number in the display.

($\tan^{-1}$) Touched after F, computes the arc tangent (in degrees) of the number in the display.

Note: To compute hyperbolic and inverse hyperbolic functions, see Appendix B—Part 2.

Example: Find the cotangent, secant and cosecant of 30°. Using the formulae:

$$\cot = \frac{1}{\tan},\ \sec = \frac{1}{\cos},\ \csc = \frac{1}{\sin}$$

KEY IN	DISPLAY SHOWS	COMMENTS
30	30	
MS	30.	Store for further use without having to re-enter.
tan	.5773502	
1/x	1.732051	Cotangent 30°
MR	30.	Re-enter 30°
cos	.8660255	
1/x	1.1547004	Secant 30°
MR	30.	Re-enter 30°
sin	.5	
1/x	2.	Cosecant 30°

Example: Find the arc cotangent of 1.7320508.
arc cot 1.7320508 = 30°

KEY IN	DISPLAY SHOWS
1.7320508	1.7320508
1/x	.57735027
F	.57735027
($\tan^{-1}$)	30.

Degree/Radian Conversion

(rad) Touched after F, converts the contents of the display from degrees to radians.

(deg) Touched after F, converts the contents of the display from radians to degrees.

Example: How many radians is 30°?
Key in 30 F (rad) ;
display shows: .52359877.

Example: What is the sine of .6 radians?
Key in .6 F (deg) sin ;
display shows: .5646425.

Two-Factor Calculations

To perform two-factor calculations, key in the first number, touch ENT, then key in the second factor and touch the desired function key.

Basic Functions (+ — × ÷)

+ Adds "x" to "y".

— Subtracts "x" from "y".

Example: Key in 5 ENT 3 + ;
display shows: 8.

× Multiplies "y" by "x".

÷ Divides "y" by "x".

Example: Key in 36 ENT 12 ÷ ;
display shows: 3.

Power and Root Functions

y^x Raises "y" to the "x" power.

Example: Key in 5 ENT 3 y^x;
display shows: 124.9999.*

Since taking the x^{th} root of y is the same as raising y to the 1/x power, roots are obtained by touching 1/x before touching y^x. Example: key in 125 ENT 3 1/x y^x; display shows: 4.999995.*

**Note:* The reason for the small variation from the absolute answer is that the calculator uses a log, antilog method of raising to powers; i.e., $y^x = e^{x \ln y}$. See Appendix A for a diagram of how this function works on the stack.

Chain Calculations

The number in the display is always ready to have calculations performed on it.

Example: (2 + 3) x (4 + 5) = 45.

KEY IN	DISPLAY SHOWS
2	2
ENT	2.
3	3
+	5.
4	4
ENT	4.
5	5
+	9.
×	45.

Memory

MS Stores the number in the display in memory (register M).

MR Recalls the contents of memory (register M) to the display (register X).

lemory Clear

o clear memory, key in: 0 MS .

ccumulating Memory

/I+) Touched after F , adds contents of display to contents of memory. Display (register X) is unaffected.

/I–) Touched after F , subtracts contents of display from contents of memory. Display (register X) is unaffected.

/I+x²) Touched after F , adds square of contents of display to contents of memory. Display register X) is unaffected.

xample: Compute the following: $\Sigma x = 1 + 2 + 3 = 6$; $x^2 = 1^2 + 2^2 + 3^2 = 14$.

KEY IN	DISPLAY SHOWS	COMMENTS
0 MS	0.	Clear memory.
1	1	

KEY IN	DISPLAY SHOWS	COMMENTS
F (M+x^2)	1.	x^2 summed in memory.
ENT	1.	x summed in register X.
2	2	
F (M+x^2)	2.	$1^2 + 2^2$ in memory.
+	3.	1 + 2 in register X.
3	3	
F (M+x^2)	3.	$1^2 + 2^2 + 3^2$ in memory.
+	6.	1 + 2 + 3 in register X.
MR	14.	Recall x^2.

Calculations with Large Numbers

The capacity of your calculator is ±99999999. On occasion, however, you may be doing a calculation which involves numbers expressed in scientific notation or whose result would exceed the capacity of the machine.

There are basically three types of multiplication and division problems involving large numbers that can be handled on your calculator: straight multiplication and division problems; fractional multiplication and division problems; and scientific notation multiplication and division problems. All three kinds are handled by adding and subtracting common logs, then finding the common antilog.

Straight Multiplication and Division

The rule here is: If you are multiplying by a number, add its common log; if you are dividing by a number, subtract its common log.

Example: Multiply 445662 x 776550 x 8876.25:

KEY IN	DISPLAY SHOWS	COMMENTS
445662	445662	First number.
log	5.649006	Common log.

KEY IN	DISPLAY SHOWS	COMMENTS
MS	5.649006	Store first log in memory.
776550	776550	Second number.
log	5.890169	Common log.
F (M+)	5.890169	Add to memory.
8876.25	8876.25	Third number.
log	3.94823	Common log.
F (M+)	3.94823	Add to memory.
MR	15.487405	Recall summed logs.
15	15	Key in the characteristic (number to left of decimal) of summed logs. This is your power of 10.
MS	15.	Store characteristic.
−	.487405	Subtract characteristic from mantissa (number to right of decimal point) leaving mantissa.
10	10	
x–y	.487405	
y^x	3.071883	Take common antilog. This number is the mantissa portion of answer in scientific notation.
MR	15.	This number is the exponent portion of answer in scientific notation. Hence, the answer is 3.071883×10^{15}.

Note: Touch x–y to see either part again.

Example: 2163 x .067 ÷ 8766500:

KEY IN	DISPLAY SHOWS	COMMENTS
2163	2163	
log MS	3.335056	Compute, store first log
.067	.067	
log F (M+)	−1.173925	Compute and add second log (multiplication)
8766500	8766500	
log F (M−)	6.942826	Compute and subtract third log (division).
MR	−4.781695	Recall summed logs.
4 CHS	−4	
MS —	−.781695	Store and subtract characteristic.
10	10	Compute common antilog.
x-y y^x	.1653123	Mantissa portion of scientific notation answer.
MR	−4.	Exponent portion of scientific notation answer. Answer .1653123 x 10^{-4}

Fractional Multiplication and Division

These types of equations are in the form: $\frac{a \times b}{c \times d}$. The rule here is: For numerators, add logs for multiplication and subtract logs for division. For denominators, subtract logs for multiplication and add logs for division.

Example: Calculate $\frac{33882 \div 0.75 \times 88644}{526 \times .028 \div 126}$:

KEY IN	DISPLAY SHOWS	COMMENTS
33882	33882	On top of line:
log MS	4.529969	Compute, store first log.
.75	75	

KEY IN	DISPLAY SHOWS	COMMENTS
log F (M−)	−.1249387	Compute and subtract second log (division).
88644	88644	
log F (M+)	4.947649	Compute and add third log (multiplication).
526	526	Under the line:
log F (M−)	2.720986	Compute and subtract first log.
.028	.028	
log F (M−)	−1.552842	Compute, subtract second log (multiplication).
126	126	
log F (M+)	2.10037	Compute and add third log (division).
MR	10.534782	Recall summed logs.
10 MS —	.534782	Store and subtract characteristic.
10 x-y y^x	3.425955	Compute common antilog to obtain mantissa portion of scientific notation answer.
MR	10	Recall exponent portion of scientific notation answer. Answer is 3.425995 x 10^{10}.

Scientific Notation Multiplication and Division

These types of equations are in the form: a x 10^i x b x 10^j. The rule here is: When multiplying, add the logs of the mantissas and the actual value of the exponents (this is because what you are really adding is log 10 x exponent = 1 x exponent = exponent); if you are dividing, subtract the logs of the mantissas and the actual value of the exponents. If you have equations that are in fractional form that involve scientific notation, combine these rules with those for fractional notation.

Example: Calculate 9.886 x 10^{13} x 7.654 x 10^5:

KEY IN	DISPLAY SHOWS	COMMENTS
9.886	9.886	
log MS	.9950206	Compute, store first log.
13	13	
F (M+)	13.	Add actual value of exponent.
7.654	7.654	
log F (M+)	.8838885	Compute and add log of next mantissa.
5	5	
F (M+)	5.	Add actual value of next exponent.
MR	19.878908	Recall summed logs.
19 MS —	.878908	Store and subtract characteristic.
10 x–y y^x	7.56672	Compute common anti-log to obtain mantissa portion of answer.
MR	19.	Recall exponent portion of answer. Answer is 7.56672×10^{19}.

Remember, touching x–y will recall the other portion of the answer.

Example: Calculate $9.886 \times 10^{13} \div 7.654 \times 10^{-5}$:

KEY IN	DISPLAY SHOWS	COMMENTS
9.886	9.886	
log MS	.9950206	Compute, store first log.
13	13	
F (M+)	13.	Add actual value of exponent.
7.654	7.654	
log F (M–)	.8838885	Compute and subtract log of next mantissa (division).
5 CHS	–5	

KEY IN	DISPLAY SHOWS	COMMENTS
F (M–)	–5.	Subtract actual value of exponent (division).
MR	18.111132	Recall summed logs.
18 MS —	.111132	Store and subtract characteristic.
10 x–y y^x	1.291611	Compute common anti-log to obtain mantissa portion of answer.
MR	18.	Recall exponent portion of answer. Answer is 1.291611×10^{18}.

Error Conditions

In the event of a logic error (e.g., division by zero), your calculator will display all zeros and decimal points. An error condition can be reset by touching any key. If the error condition is cleared by any key except C, the machine will assume that the contents of register X = 0.

Programming

The addition of "learn-mode" programming to your already powerful calculator provides you with a unique time saving approach to the evaluation of long equations or those which require iterative solutions.

"Learn-mode" programming is essentially automatic key pressing. One of the benefits of the "learn-mode" programmer is its inherent simplicity in developing and using a program. Effective use of the programmer does not require any special skills or knowledge of programming languages. If you can use the calculator, you can use the programmer!

The basic technique of "learn-mode" programming is that the programmer remembers the sequence of key depressions used to solve a problem. Therefore, to program the calculator, all you must do is solve the problem once correctly with the programmer in LOAD mode. Even if mistakes are made, when the proper corrections are made, the programmer will learn the corrections and yield the proper solution.

Note: Be sure you switch your calculator on in RUN position, then slide the LOAD/STEP/RUN switch to LOAD before entering programs.

LOAD/STEP/RUN Switch

LOAD — Allows loading of program steps into the program storage area.

STEP — Executes one step of a stored program for each touch of start.

RUN — Permits execution of programs by use of the start or skip keys.

Programming Keys

start The start key has functions in both the LOAD and RUN modes. In LOAD mode, touching start will erase all previously stored information in the program storage area, write a START code and mark the beginning of the first program. In the RUN mode, touching start begins execution of the first program. If the programmer is stopped at a HALT code (explained below), touching start continues the program to the next HALT or to the end of the program. After reaching the end of a program, the programmer always returns to the START code at the beginning of the first program.

halt The halt key functions only in the LOAD mode and is used to insert a HALT code in the program sequence. In the RUN mode, when the programmer encounters a HALT code, it stops the playback of the program and returns control of the calculator back to the user. halt is usually used as a pause in the program execution to allow the reading of an intermediate result and/or to input a variable for further processing. Normally, start is used to leave the HALT condition and continue execution of the program, but it will also allow branching to the next or subsequent programs if skip is touched.

skip The skip key has functions in both the LOAD and RUN modes. In the LOAD mode, touching skip marks the beginning of programs other than the first. It writes a SKIP code for each subsequent program. In RUN mode, touching skip causes the programmer to jump from the beginning of a program, or from a HALT point, to the beginning of the next program and begin execution of that program. Execution continues to the first HALT or to the end. If only one program is stored and the programmer is stopped at a HALT, touching skip will jump over the remaining part of the program and return to the beginning of the program. This feature may be used to create a "loop" within the main program. When only two programs are stored, touching skip effectively executes the second program and touching start executes the first. When more than two programs are stored, a HALT code must be programmed in somewhere in all programs except the first. To execute the second program, touch skip; to execute the third program, touch skip twice; to execute the n^{th} program, touch skip n-1 times.

del The del key (delete) functions only in the LOAD mode and is used for editing the program. Its purpose is to remove entries from the program memory. Touching del always starts with the last program step entered and removes one entry each time it is touched. It is essentially a backspace key. Using the del key can cause the error alarm to come on if an attempt is made to delete a START or SKIP code. When a SKIP code is deleted, the alarm means that an entire program has been removed. Touching del again will turn off the error alarm and delete the last step of the program preceding the deleted SKIP. If the alarm does not go off with the next touch of del , it means that the START code is the only code left in the program memory and all programs have been cleared. The START code cannot be deleted. If a SKIP code is deleted accidentally, re-entering skip will reinitiate that program.

Error Alarm

Your programmable calculator has a program memory capacity of 100 steps. If more than 100 program steps are entered, or if a SKIP code is deleted, or an attempt is made to delete a START code, an error alarm consisting of all decimal points will be displayed. This alarm differs from the automatic display shutoff in that the decimals appear with the number in the display. For example, .1.2.3.4.5. . . would indicate an error alarm.

Entering Variables

With the LOAD/STEP/RUN switch in the LOAD position, key in halt wherever a variable is going to appear in your program. With the LOAD/STEP/RUN switch in the RUN position, when the programmer encounters a HALT code, control of the calculator is returned to the user. A variable can be entered for further processing at that time. A variable consists of any number entry key (0-9), • , CHS or π. Touching start after the variable has been entered continues the program from the HALT code. The variable does not become part of the program. It consists of the one program step halt .

Entering Constants

With the LOAD/STEP/RUN switch in LOAD position, keying in any number entry key (0-9), • , CHS or π WITHOUT preceding the number with a halt enters

the number in the program as a constant. This constant will automatically be keyed in and used each time the program is run. The constant takes as many program steps as there are digits (including decimal and CHS) in the number.

Programmer Control Operations

The following table summarizes the operations of the programming switch and keys.

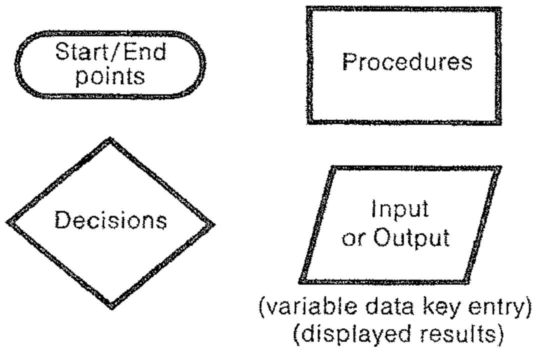
KEY	LOAD MODE	RUN MODE
start	Clears program area and writes START code.	Starts first program or continues from a HALT.
skip	Marks beginning of programs subsequent to the first.	Jumps over remainder of a program and begins execution of the next program.
del	Deletes the last step entered in a program.	No function.
halt	Writes a HALT code in the program and returns control of calculator to user.	No function.

SWITCH	FUNCTION
RUN position	Permits execution of programs by use of start or skip .
LOAD position	Allows loading of programs into program storage area.
Switching from LOAD to RUN position	Positions programmer control to begin execution of the first program.
Switching from RUN to LOAD position	Positions programmer control to begin entering more program steps to the end of the last program entered. Previous program data is not affected. Multiple programs can be added at this point by touching skip .

Switching from RUN to LOAD to RUN position	Positions programmer control to begin execution of the first program entered. This feature can be useful if the user is interrupted during his calculations and forgets which portion of the program was last executed.
STEP position	Executes one step of a stored program for each touch of the start key.

Programming Examples

The following examples will illustrate the versatility of your "learn-mode" programmer. The following programming symbols will be used to help define the flow of execution of sample programs:

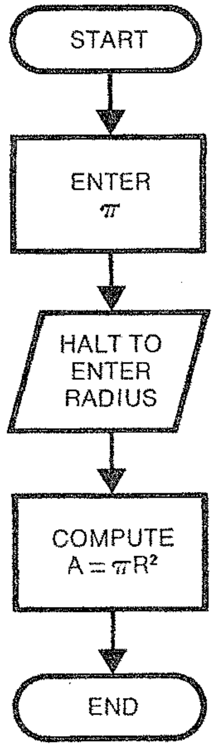


Entering Single Programs

Your "learn-mode" programmer functions as a calculator while it is being programmed, thus enabling you to use actual data to get a meaningful result as you program. While it is not necessary to use actual data while keying in a sequence of program steps, doing so for simple, non-iterative programs where a result can be predicted can be quite helpful. This feature lets you "debug" your program by seeing if the calculated results displayed at the end of programming are the same as your predicted results for the calculations involved. If the results are the same, you have keyed in the program correctly.

Example: Program your calculator to find the area of circles of varying radii using the formula $A = \pi r^2$. We can use a "test" variable of 1 in this program. Thus, if after keying in the program we come up with a calculated result of π, we know that the program was keyed in correctly.

Desired flow of execution:



Loading the program: Using 1 as a "test" variable:

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
1	start						Mark beginning of program.
2	π		π				
3	halt		π				Halt to enter radius.
		1	1	π			r.*
4	F		1	π			
5	(x^2)		1.	π			r^2.
6	×		π				πr^2. End of program.

Running the program: Switch: RUN position.
What is the area of a circle of radius 5? 2.25? 8.73?

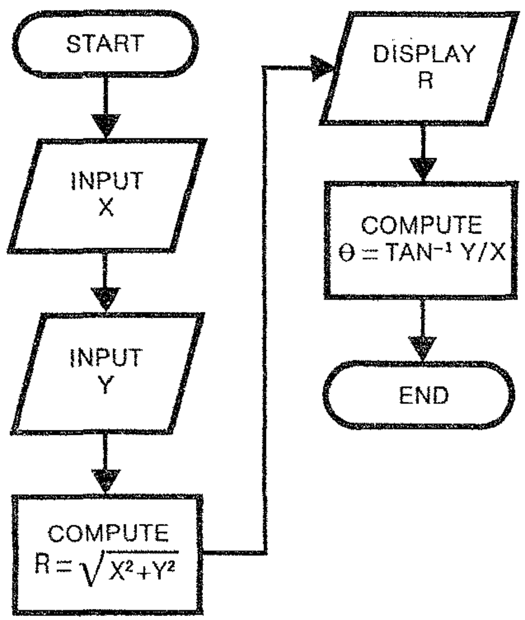
KEY IN	DISPLAY SHOWS	COMMENTS
start	3.1415926	Start program and execu to first HALT.
5	5	First radius.
start	78.539815	Program continues to en First area computed.
start	3.1415926	Start program and execu to first HALT.
2.25	2.25	Next radius.
start	15.904312	Next area computed.
start	3.1415926	Start program.
8.73	8.73	Next radius.
start	239.42988	Next area.

Example: Program the calculator to convert rectangular coordinates to polar coordinates ,using the formulas:

$$R = \sqrt{x^2 + y^2}$$
$$\theta = \tan^{-1}(y/x.)$$

By the example in Appendix B — Part 1, we see that if we use "test" variables of $x = 6$, $y = 8$, we can predict $R = 10$ and $\theta = 53.1301$.

Desired flow of execution:



Loading the program:
Using x = 6, y = 8 as "test" variables:

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
1	start						Mark beginning of program.
2	halt						Halt for x.
		6	6				x.
3	ENT		6.	6.			
4	ENT		6.	6.	6.		
5	×		36.	6.			x^2.
6	halt						Halt for y.
		8	8	36.	6.		y.
7	MS		8	36	6	8	
8	F		8	36	6	8	
9	(x^2)		64.	36	6	8	y^2.
10	+		100.	6		8	$x^2 + y^2$.
11	$\sqrt{\ }$		10.	6		8	$\sqrt{x^2 + y^2} = r$.
12	halt		10.	6		8	Halt to display r.
13	x–y		6	10		8	
14	MR		8	6	10	8	
15	x–y		6	8	10	8	
16	÷		1.3333333	10		8	
17	F		1.3333333	10		8	
18	$(\tan^{-1})$		53.1301	10			θ calculated. End of program.

Running the program: Switch: RUN position.

Convert the following rectangular coordinates to polar coordinates: $x = 4$, $y = 7$; $x = 16.223$, $y = 14.56$

KEY IN	DISPLAY SHOWS	COMMENTS
start	53.1301	Start program, execute t first HALT. (Display sho result of programming).
4	4	x.
start	16.	Program continues to next HALT.
7	7	y.
start	8.0622577	Program continues to next HALT. R calculated and displayed.
start	60.25511	Program continues to en θ calculated, displayed.
start	60.25511	Start program, execute t first HALT. (Display sho result of last program).
16.223	16.223	x.
start	263.18572	Program continues to next HALT.
14.567	14.567	y.
start	21.803284	Program continues to next HALT. R calculated and displayed.
start	41.92139	Program continues to end. θ calculated, displayed.

Saving Programming Steps

While you have 100 programming steps on your calculator, it is often beneficial to program equations in as few steps as possible. Re-arranging equations and setting up programs to enable the machine to begin calculations with the number currently in the display can lead to saving both programming steps in LOAD position and touches of start in RUN position.

Let's rewrite the first two programs using some step-saving techniques. In the first program involving the area of a circle, we can see that if we set up the equation to square whatever is in the display and then multiply it by π, we can eliminate one touch of start.

Loading the program:

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
		1	1				"Test" variable.
Switch: LOAD position							
1	start		1				Mark beginning of program.
2	F		1				
3	(x^2)		1				
4	π		π	1			
5	×		π				End of program.

Running the program: Switch: RUN position.

The program will now take whatever is in the displa at the time you touch start, square it and multiply i by π; hence, πr^2.

Example: What is the area of a circle of radius 5? Of radius 7.7? Of radius 83.6?

KEY IN	DISPLAY SHOWS	COMMENTS
5	5	First radius.
start	78.539815	Start program and exec to end. Area calculated and displayed.
7.7	7.7	Next radius.
start	186.26502	Next area.
83.6	83.6	Next radius.
start	21956.465	Next area.

Loading the program:

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTE ONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
		0	0				
	MS		0				Clear memory.
		8	8				y.
Switch: LOAD position							
1	start		8				Mark beginning of program.
2	F		8				
3	(M+x²)		8.			64.	Add y² to memory (0).
4	halt		8			64.	Halt for x.
		6	6	8		64.	
5	F		6.	8		64.	
6	(M+x²)		6	8		100.	Add x² to memory (y²).
7	÷		1.3333333			100.	
8	F		1.3333333			100.	
9	(tan⁻¹)		53.1301			100.	Θ calculated.
10	MR		100.	53.1301		100.	Recall y² + x².
11	√		10.	53.1301		100.	r calculated. End of program.

Programming the equation in this manner saved one programming step in the program and eliminated one touch of start while running the program.

In the second program we can use the $(M+x^2)$ feature to save programming steps and the x↔y feature to eliminate one touch of start.

Running the program: Switch: RUN position.

Convert the following rectangular coordinates to po
coordinates: $x=4, y=7$; $x=35.6, y=75.25$.
We have to clear memory each time before running
this program and input y first.

KEY IN	DISPLAY SHOWS	COMMENTS
0 MS	0.	Clear memory.
7	7	y.
start	7.	Start program and exec to first HALT.
4	4	x.
start	8.0622577	R calculated, displayed.
x-y	60.25511	Recall Θ from register Y
0 MS	0.	Clear memory.
35.6	35.6	y.
start	35.6	
75.25	75.25	x.
start	83.246156	R calculated, displayed.
x-y	25.31835	Recall Θ from register Y

Programming the equation in this manner saved sev programming steps and one touch of start .

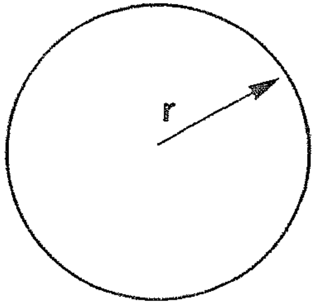
Entering Multiple Programs

Your "learn-mode" programmer has 100 programming steps. You can enter as many programs as will fit into 100 steps. The use of the skip key enables you to enter more than one program. While in LOAD position, touching skip terminates one program and marks the beginning of a new program. While in RUN position, touching skip makes the programmer jump from the beginning of the first program to the beginning of the second program and start execution of that program; or, touching skip makes the programmer jump from a HALT code in any program to the beginning of the next program and start execution of that program. If you plan to have more than two programs in the machine, make sure there is a HALT code programmed somewhere in all programs except the first. Since the programmer jumps from a HALT code on all programs except the first, it will not be able to access the third, fourth, etc. programs unless a HALT code is built into those programs. If you want the programmer to stop at the beginning of each program, build the HALT code in as the first program step after skip .

The programmer will return to the beginning of the first program after completing any program. Thus, after executing program 3, the programmer returns to the top and gets ready to execute program 1 again.

Example: Program the calculator to compute the area and circumference of a circle of radius r and the volume of a sphere of radius r.

Program 1 will calculate the area;
Program 2 will calculate the circumference; and
Program 3 will calculate the volume.



$A = \pi r^2$;

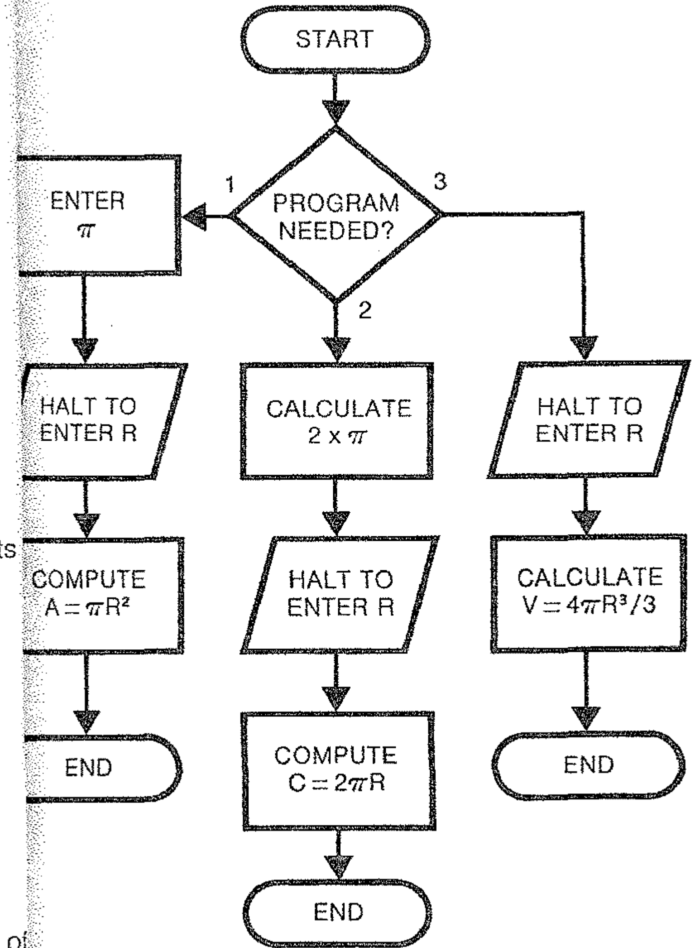
$C = 2\pi r$;

$V = 4/3\pi r^3$.

Using a "test" variable of 1, our predicted results should be:

Program 1: $A = 3.1415926$ (π);
Program 2: $C = 6.2831852$ (2π); and
Program 3: $V = 4.18879$ ($4\pi/3$).

esired flow of execution:



Loading the programs: Using a "test" variable of

LINE NO.	KEY ENTRY	DATA ENTRY	REGIS NTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
1	start						Mark beginning of program 1.
2	π		π				
3	halt						Halt for r.
		1	1	π			r.

Loading the programs: Using a "test" variable of

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
4	F		1	π			
5	(x^2)		1.	π			
6	×		π				A calculated. End of program 1.
7	skip		π				Mark beginning of program 2.
8	2		2	π			
9	ENT		2.	2	π		
10	π		π	2	π		
11	×		2π	π			
12	halt		2π	π			Halt for r.
		1	1	2π	π		
13	×		2π				C calculated. End of program 2.
14	skip		2π				Mark beginning of program 3.
15	halt		2π				Halt for r.
		1	1	2π			
16	ENT		1	1	2π		
17	ENT		1	1	1		
18	×		1	1			
19	×		1				r^3.
20	π		π	1			
21	×		π				πr^3.
22	4		4	π			
23	×		4π				
24	3		3	4π			
25	÷		$4\pi/3$				$4\pi r^3/3$. V calculated. End pgm 3.

Running the program: Switch: RUN position.

Compute:

1. The area of a circle of radius 5;
2. The circumference of a circle of radius 6.6;
3. The area of a circle of area 7.25; and
4. The volume of a sphere of radius 4.5.

For problem 1, we need program 1. For problem 2 we need program 2. For problem 3, we need progra[m] 1 and for problem 4, we need program 3.

KEY IN	DISPLAY SHOWS	COMMENTS
start	3.1415926	Start program 1 and exe- cute to first HALT.
5	5	R for problem 1.
start	78.539815	Program continues to en[d of] program 1, area calcula[ted].
skip	6.2831852	Skip program 1, begin program 2 and execute [to] first HALT.
6.6	6.6	R for problem 2.
start	41.469022	Program continues to e[nd] of program 2, circumfer- ence calculated.
start	3.1415926	Start program 1 and exe- cute to first HALT.
7.25	7.25	R for problem 3.
start	165.12996	Program continues to end of 1, area calculated.
skip	6.2831852	Skip program 1, begin program 2 and execute to first HALT.
skip	6.2831852	Skip program 2, begin pro- gram 3 and execute to first HALT (at the beginning of the program).
4.5	4.5	R for problem 4.
start	381.70346	Program continues to end of program 3, volume calculated.

By re-arranging the area, circumference and volume programs to work on what is already in the display, we can eliminate some touches of start. We will not use a "test" variable while loading the programs this time, however, so the displayed results while programming will be unpredictable. Remember, if while you are programming without "test" variables the display shows an error indication, it doesn't matter. Keep on keying in the program. The next key entry will reset the error condition and it will not affect your program. You may "debug" your program with test variables after they are programmed.

Loading the programs:

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
1	start		0				Mark beginning of program 1.
2	F		0				
3	(x^2)		0				
4	π		π				

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
							End of program 1.
5	×		0				Mark beginning of program 2.
6	skip		0				Halt at beginning of program 2.
7	halt		0				
8	ENT		0				
9	π		π				
10	×		0				
11	2		2				End of program 2.
12	×		0				Mark beginning of program 3.
13	skip		0				Halt at beginning of program 3.
14	halt		0				
15	ENT		0				
16	ENT		0				
17	×		0				
18	×		0				
19	π		π				
20	×		0				
21	4		4				
22	×		0				
23	3		3				End of program 3.
24	÷		0				

Running the program: Switch: RUN position.

Compute:

1. The area of a circle of radius 7.75;
2. The volume of a sphere of radius 13; and
3. The circumference of a circle of radius 9.2.

KEY IN	DISPLAY SHOWS	COMMENTS
7.75	7.75	R for problem 1.
start	188.6919	Start program 1. Program continues to end, area calculated.
13	13	R for problem 2.
skip	13	Skip program 1.
skip	13	Skip program 2.
start	9202.7716	Start program 3, program continues to end, volume calculated.
9.2	9.2	R for problem 3.
skip	9.2	Skip program 1.
start	57.805302	Start program 2, progra continues to end, circum ference calculated.

Programming this way eliminates one program step and makes it easier to tell which program you are running. Remember, the programmer returns to the beginning of program 1 after executing any progra Thus, because the programmer returns to the begin ning of program 1, to execute program 3 after exec ing program 2, you must touch skip to skip progra 1, and skip to skip program 2.

"Indexing" Your Programs

"Index" numbers can be built into programs in such a way to indicate which program is about to be executed. There is no set way to "index" programs. Try a few sample programs out and some of your own to get a feel for it. Just remember to watch your registers so as not to destroy data that you may need in a particular program.

"Indexing" With Memory

One way to "index" a program is to build the index number in memory. Obviously, these kinds of programs would have to be those which do not use the contents of memory from another program.

Example: Write two programs:

Program 1: Converts centigrade to fahrenheit using the formula: $F° = 9/5\ C° + 32$

Program 2: Converts fahrenheit to centigrade using the formula: $C° = 5/9\ (F° - 32)$.

These programs will be written in two ways. One way will illustrate a method of indexing where the variable is keyed in immediately after the index number is displayed before touching start, the other method will illustrate touching start after the index number is displayed to start the program before entering the variable.

Method 1: With this method, you enter the variable just after the index number is displayed, then touch start to start the program. This saves a touch of start.

Loading the programs:

No "test" variables will be used.

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
1	start						Mark beginning of program 1.
2	1		1				"Index" number.
3	MS		1.			1.	Store in memory.
4	halt		1.			1.	Halt for °C.
5	ENT		1.	1.		1.	
6	9		9	1.		1.	
7	×		9.			1.	
8	5		5	9.		1.	
9	÷		1.8			1.	
10	3		3	1.8		1.	
11	2		32	1.8		1.	
12	+		33.8			1.	°F calculated.
13	skip		33.8			1.	Mark beginning of program 2.
14	2		2	33.8		1.	"Index" number.
15	MS		2.	33.8		2.	Store in memory.
16	halt		2.	33.8		2.	Halt for °F.
17	ENT		2.	2.	33.8	2.	
18	3		3	2.	33.8	2.	
19	2		32	2.	33.8	2.	
20	−		−30.	33.8		2.	
21	5		5	−30.	33.8	2.	
22	×		−150.	33.8		2.	
23	9		9	−150.	33.8	2.	
24	÷		−16.666666	33.8		2.	°C calculated.

Running the program: Switch: RUN position.

Problem 1: Convert 100°C to °F;
Problem 2: Convert 212°F to °C;
Problem 3: Convert 78°F to °C.

KEY IN	DISPLAY SHOWS	COMMENTS
start	1.	Program 1 is ready to execute.
100	100	°C.
start	212.	Program executes to er °F computed.
skip	2.	Program 2 is ready to execute.
212	212	°F.
start	100.	Program executes to er °C computed.
skip	2.	Program 2 is ready to execute.
78	78	°F.
start	25.555555	°C. computed.

Method 2: With this method, you start the program after seeing the "index" number and before entering the variable. While this involves an extra touch of start, it can be useful if the first step in your program involves entering a constant. You have to have a HALT code built in after the index number in order for the machine to stop and display the index number. If the next step in the program is to enter a constant, you have to have some way to terminate the HALT condition so the constant will not be treated as a variable. Touching F does not affect the stack, does not affect single function keys, but DOES terminate a HALT condition. Using this feature, constants can be programmed in following a HALT.

Loading the program:
Again, no "test" variables will be used.

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
1	start						Mark beginning of program 1.
2	1		1				"Index" number.
3	MS		1.			1.	Store in memory.
4	halt		1.			1.	Halt to display "index."
5	F		1.			1.	Clear HALT code.
6	9		9	1.		1.	Constant.
7	ENT		9.	9.	1.	1.	
8	halt		9.	9.	1.	1.	Halt for °C.
9	×		81.	1.		1.	
10	5		5	81	1.	1.	
11	÷		16.2	1.		1.	
12	3		3	16.2	1.	1.	
13	2		32	16.2	1.	1.	
14	+		48.2	1.		1.	°F calculated.
15	skip		48.2	1.		1.	Mark beginning of program 2.
16	2		2	48.2	1.	1.	"Index" number.
17	MS		2.	48.2	1.	2.	Store in memory.
18	halt		2.	48.2	1.	2.	Halt to display "index."
19	F		2.	48.2	1.	2.	Clear HALT code.
20	5		5	2.	48.2	2.	Constant.
21	ENT		5.	5.	2.	2.	
22	halt		5.	5.	2.	2.	Halt for °F.
23	ENT		5.	5.	5.	2.	
24	3		3	5.	5.	2.	

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
25	2		32	5.	5.	2.	
26	—		−27.	5.		2.	
27	×		−135.			2.	
28	9		9	−135.		2.	
29	÷		−15.			2.	°C calculated.

Running the program: Switch: RUN position.

Problem 1: Convert 212°F to °C;
Problem 2: Convert 13°C to °F;
Problem 3: Convert 69°F to °C.

KEY IN	DISPLAY SHOWS	COMMENTS
skip	2.	Program 2 is ready to execute.
start	5.	Start program 2 and execute to first HALT.
212	212	°F.
start	100.	°C computed.
start	1.	Program 1 is ready to execute.
start	9.	Start program 1 and execute to first HALT.
13	13	°C.
start	55.4	°F calculated.
skip	2.	Program 2 is ready to execute.
start	5.	Start program 2.
69	69	°F.
start	20.555555	°C calculated.

"Indexing" Without Memory

If you want to use the contents of memory in a program, perhaps because a variable is stored there, you can "index" your program using the stack. Be aware of what your index numbers will do to the stack because they do take up a position in the X register. It is advisable to clear the X register after displaying the "index" number.

Example: Let's reprogram the area, circumference and volume programs this time putting the variable (r) which is common to all three programs in memory and "indexing" the programs to tell which is which.

Program 1 = Area of circle = πr^2;
Program 2 = Circumference of circle = $2\pi r$;
Program 3 = Volume of sphere = $4/3\pi r^3$.

Loading the program:

A "test" variable of 1 will be used. Key in: 1 MS.

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
1	start					1.	Mark program 1.
2	C					1.	Clear previous "index" number.
3	C					1.	
4	1		1			1.	"Index" number.
5	halt		1			1.	Halt to display "index."
6	MR		1.	1.		1.	Recall variable (r).
7	F		1.	1.		1.	
8	(x^2)		1.	1.		1.	r^2.
9	π		π	1.	1.	1.	
10	×		π	1.		1.	πr^2.
11	skip		π	1.		1.	Mark program 2.
12	C		1.			1.	Clear previous "index" number.
13	C		0.			1.	
14	2		2			1.	"Index."
15	halt		2			1.	Halt to display "index."
16	MR		1.	2.		1.	Recall r.
17	ENT		1.	1.	2.	1.	
18	π		π	1.	1.	1.	
19	×		π	1.		1.	
20	2		2	π	1.	1.	
21	×		2π	1.		1.	$2\pi r$.
22	skip		2π	1.		1.	Mark program 3.
23	C		1.			1.	Clear previous index.
24	C		0.			1.	

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
25	3		3			1.	"Index."
26	halt		3			1.	Halt to display "index."
27	MR		1.	3.		1.	
28	ENT		1.	1.	3.	1.	
29	ENT		1.	1.	1.	1.	
30	×		1.	1.		1.	
31	×		1.			1.	r^3.
32	π		π	1.		1.	
33	×		π			1.	
34	4		4	π		1.	
35	×		4π			1.	
36	3		3	4π		1.	
37	÷		$4/3\pi$			1.	$4/3\pi r^3$.

Running the program: Switch: RUN position.

Problem 1: What is the area of a circle of radius

Problem 2: What is the volume of a sphere of radius 5?

Problem 3: What is the circumference of a circle of radius 9.6?

Problem 4: What is the area of a circle of radius 9.6?

KEY IN	DISPLAY SHOWS	COMMENTS
5 MS	5.	R for problem 1.
start	1	Program 1 is ready to execute.

KEY IN	DISPLAY SHOWS	COMMENTS
start	78.539815	Area for problem 1.
skip	2	Skip program 1.
skip	3	Skip program 2. Program 3 is ready to execute. R for problem 2 is already in memory.
start	523.59873	Volume for problem 2.
9.6 MS	9.6	R for problem 3.
skip	2	Skip program 1, program 2 is ready to execute.
start	60.318576	Circumference for problem 3.

KEY IN	DISPLAY SHOWS	COMMENTS
start	1	Program 1 is ready to execute. R for problem 4 is already in memory.
start	289.52917	Area for problem 4.

"Looping" and "Branching"— Iterative Programs

There are two methods of making your "learn-mode programmer "loop." One is using the skip key feature in single programs; the other is using the start key feature in multiple programs. The following examples will illustrate both methods.

"Looping" With SKIP

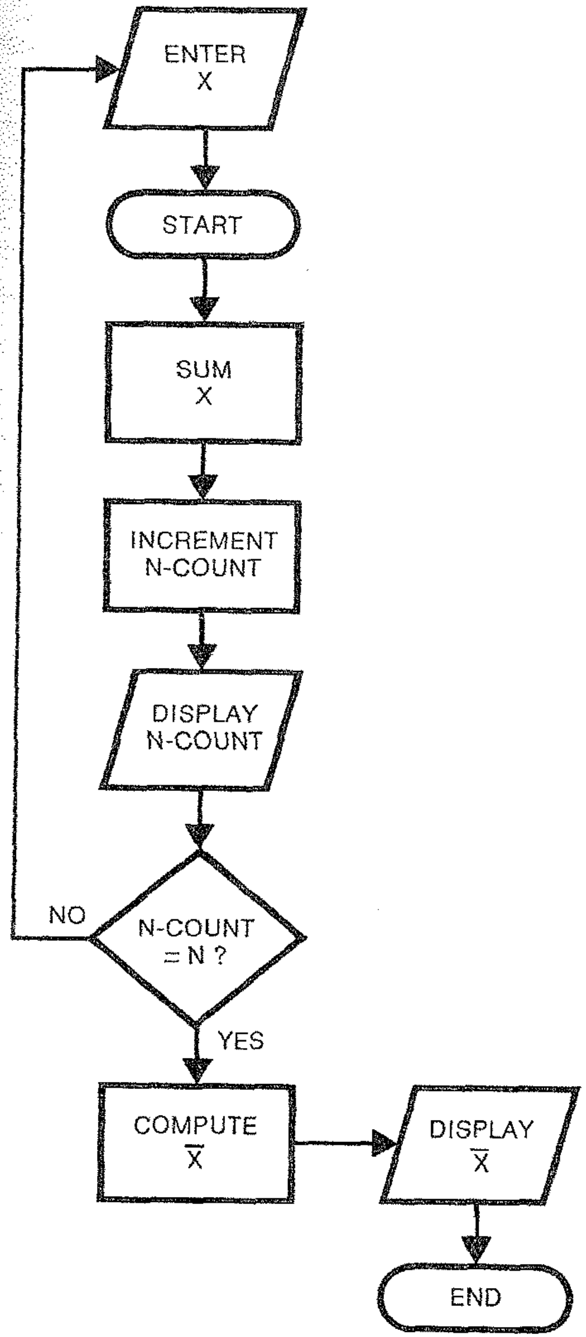
If only one program is stored and the programmer is stopped at a HALT, touching skip will jump over the remaining part of the program and start execution at the beginning of the program automatically. This feature may be used to create a "loop" within the main program.

Example: Find the mean of a list of numbers using the formula:

$$\overline{x} = \frac{\Sigma x}{n}.$$

Clearly, we need a "loop" to sum up "x." In this program, we will build an "n" count into the loop so we can always tell how many numbers we have entered. Keying in a number and touching skip will enter the number into the loop and add it to Σx. When the "n" count reaches the number of entries we have, we can "branch" out of the loop by touching start.

Desired flow of execution:



Loading the program: To program iterative progr... act as if you are programming the second, third o... n^{th} iteration and think of what you would like to ha... in each register when the iteration is next per-formed. This program has been written as if we were about to perform the n^{th} iteration. The results of iteration n-1 are currently in the registers, when we start the iteration, "x" will be in the display.

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
			Σn	Σx		Σn	Result of iteration n-1.
1	start		x	Σn	Σx	Σn	Start program. "x" is in display.
2	x–y		Σn	x	Σx	Σn	
3	C		x	Σx		Σn	Clear displayed n-count.
4	+		Σx			Σn	Sum x.
5	1		1	Σx		Σn	
6	F		1	Σx		Σn	
7	(M+)		1.	Σx		Σn	Increment n-count in memory.
8	C		Σx				Clear increment.
9	MR		Σn	Σx		Σn	Recall new n-count.
10	halt		Σn	Σx		Σn	Build in "loop" point.
11	÷		Σx/Σn				$\overline{x}$ calculated.

Running the program: Switch: RUN position.

Calculate the mean of the following five numbers: ... 5, 7, 6, 3). We need to clear all registers and memo... before using this program. Key in: C C C MS.

KEY IN	DISPLAY SHOWS	COMMENTS
2	2	x_1.
skip	1.	N-count.
5	5	x_2.
skip	2.	N-count.
7	7	x_3.
skip	3.	N-count.
6	6	x_4.
skip	4.	N-count.
3	3	x_5.
skip	5.	N-count. N-count = 5, so we "branch" to calculate mean.
start	4.6	Mean calculated.

Example: Average the following bowling scores: (212, 243, 198).

KEY IN	DISPLAY SHOWS	COMMENTS
C C		
C MS	0.	Clear registers, memory.
212	212	First score.
skip	1.	N-count.
243	243	Second score.
skip	2.	N-count.
198	198	Third score.
skip	3.	N-count = 3, "branch" to find average.
start	217.66666	Average score.

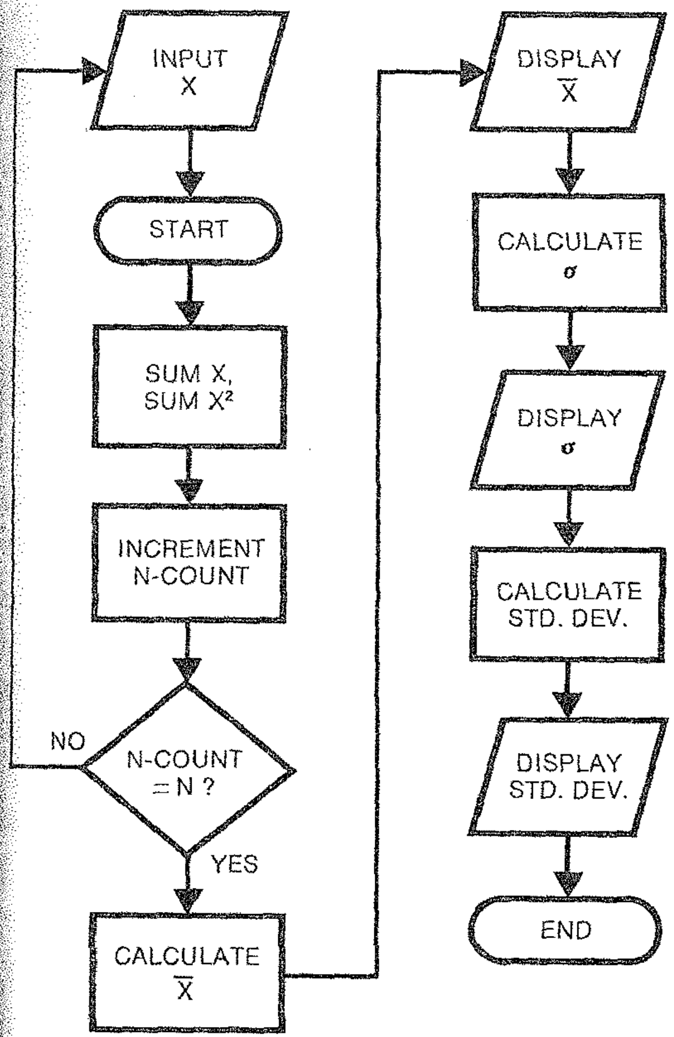
"Looping" With START

Since after executing any program, the programmer returns to the top of the program memory and gets ready to execute the first program again, we can "loop" through the first program of a multiple program set by touching start and "branch" to another program after the iterations have been completed by touching skip .

Example: Program the calculator to compute the mean, variance and standard deviation of a group of data, using the equations:

$$\overline{x} = \frac{\Sigma x}{n}; \ \sigma^2 = \frac{\Sigma x^2 - (\Sigma x)^2/n}{n-1}; \ SD = \sqrt{\sigma}$$

Desired flow of execution:



Loading the program: Assume "x" is in the display and that you are programming the n^{th} iteration of program 1. Program 1 will be the iterative program summing x, x^2 and incrementing the n-count; program 2 will compute $\overline{x}$, the variance and the standard deviation.

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
Switch: LOAD position							
			Σx	Σn		Σx^2	Results of iteration n-1.
1	start		x	Σx	Σn	Σx^2	Mark program 1.
2	F		x	Σx	Σn	Σx^2	
3	$(M{+}x^2)$		x	Σx	Σn	Σx^2	Sum x^2 in memory.
4	+		Σx	Σn		Σx^2	Sum x in stack.
5	x–y		Σn	Σx		Σx^2	Retrieve n-count.
6	1		1	Σn	Σx	Σx^2	
7	+		Σn	Σx		Σx^2	Increment n-count.
8	x–y		Σx	Σn		Σx^2	Retrieve Σx for next iteration.
9	skip		Σx	Σn		Σx^2	Mark program 2.
10	x–y		Σn	Σx		Σx^2	
11	MR		Σx^2	Σn	Σx	Σx^2	
12	x–y		Σn	Σx^2	Σx	Σx^2	
13	MS		Σn	Σx^2	Σx	Σn	Exchange Σx^2 & Σn in memory.
14	C		Σx^2	Σx		Σn	
15	x–y		Σx	Σx^2		Σn	
16	MR		Σn	Σx	Σx^2	Σn	Recall n.
17	÷		$\Sigma x/n$	Σx^2		Σn	$\overline{x}$ calculated.
18	halt		$\Sigma x/n$	Σx^2		Σn	Halt to display $\overline{x}$.
19	F		$\Sigma x/n$	Σx^2		Σn	
20	(x^2)		$(\Sigma x/n)^2$	Σx^2		Σn	$(\Sigma x/n)^2$.
21	MR		Σn	$(\Sigma x/n)^2$	Σx^2	Σn	Recall n.
22	X		$(\Sigma x)^2/n$	Σx^2		Σn	$(\Sigma x)^2/n$.

LINE NO.	KEY ENTRY	DATA ENTRY	REGISTER CONTENTS X (Display)	Y	Z	M (Memory)	COMMENTS
23	—		$\Sigma x^2 - (\Sigma x)^2/n$			Σn	$\Sigma x^2 - (\Sigma x)^2/n$.
24	MR		Σn	$\Sigma x^2 - (\Sigma x)^2/n$		Σn	Recall n.
25	1		1	Σn	$\Sigma x^2 - (\Sigma x)^2/n$	Σn	
26	—		n-1	$\Sigma x^2 - (\Sigma x)^2/n$		Σn	n-1.
27	÷		σ			Σn	Variance calculated.
28	halt		σ			Σn	Halt to display σ.
29	$\sqrt{\ }$		S.D.			Σn	Standard deviation calculated.

Running the program: Switch: RUN position.

Calculate the mean, variance and standard deviation of the following data: (2, 7, 3, 5, 2). We can iterate program 1 by entering "x" and touching start. This program does not display the n-count, so after entering all x's, we "branch" to program 2 to calculate the mean, variance and standard deviation. We have to clear all registers and memory before using this program. Key in: C C C MS.

KEY IN	DISPLAY SHOWS	COMMENTS
2	2	x_1.
start	2.	Σx.
7	7	x_2.
start	9.	Σx.
3	3	x_3.
start	12.	Σx.
5	5	x_4.
start	17.	Σx.
2	2	x_5.

KEY IN	DISPLAY SHOWS	COMMENTS
start	19.	Σx. We are through summing x, x^2 and n. "Branch" to program 2.
skip	3.8	Mean calculated and displayed.
start	4.7	Variance calculated and displayed.
start	2.1679483	Standard deviation calculated and displayed.
MR	5.	Touch MR to see n-count.

Example: Calculate the mean, variance and standard deviation of the following data: (3.44, 7.86, 4, 5, 9).

KEY IN	DISPLAY SHOWS	COMMENTS
C C C MS	0.	Clear registers, memory.
3.44 start	3.44	x_1.
7.86 start	11.3	x_2.
4 start	15.3	x_3.
5 start	20.3	x_4.

KEY IN	DISPLAY SHOWS	COMMENTS
9 start	29.3	xs. Sum of x displayed. "Branch" to program 2.
skip	5.86	Mean.
start	5.9788	Variance.
start	2.4451584	Standard deviation.

A Recap of Programming Tips

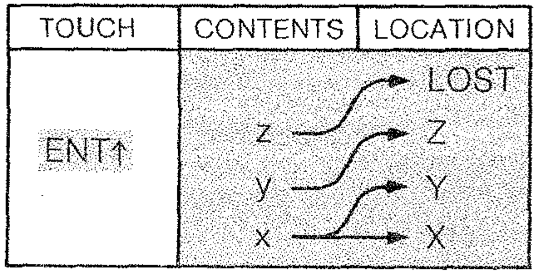
1. Always turn your machine on with the LOAD/STEP/RUN switch in RUN position. Slide it to LOAD to enter a program.
2. To clear display before starting a program, touch C until a zero appears in the display. To clear memory, touch 0 MS.
3. Touching start while in LOAD position always erases what you have already keyed in. If you change your mind in the middle of a program an want to start again, touch start.
4. The del key deletes the last step entered.
5. Mark the beginning of the first program with sta Mark the beginning of all subsequent programs with skip. If more than two programs are being stored, all except the first must have a HALT coc as part of the program to permit accessing of those programs.
6. To interrupt a program, whether to enter a variab or display a result, touch halt.
7. To enter a constant, key in the desired number. It becomes part of the program.
8. To enter a variable, key in halt. Any number following the keying in of halt will not be reme bered by the programmer and will be treated as a "test" variable. Remember that π and CHS keyed in after a halt are considered part of the "test" variable.

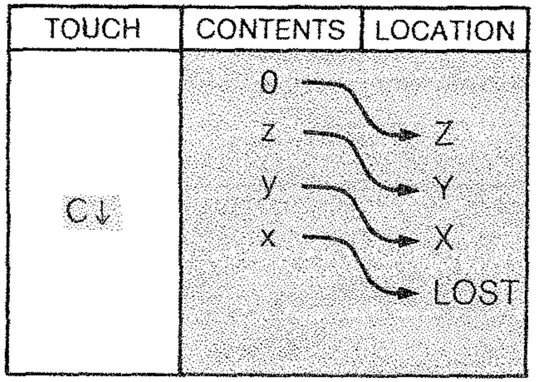
9. To start the first program, touch start.
To start the second program, touch skip.
To start the n^{th} program, touch skip n-1 times.

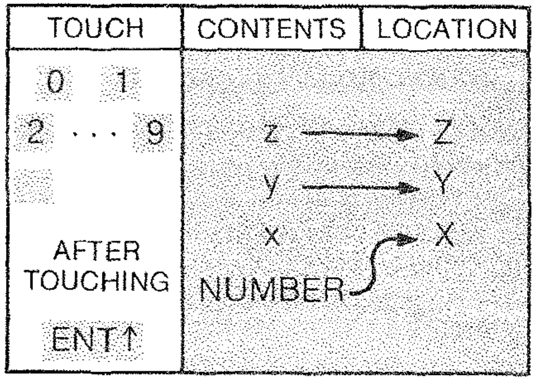
APPENDICES

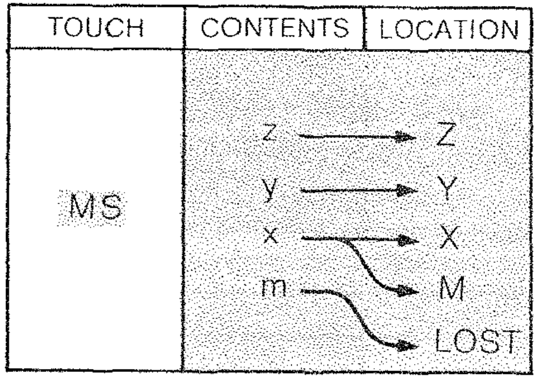
Appendix A — Stack Diagrams

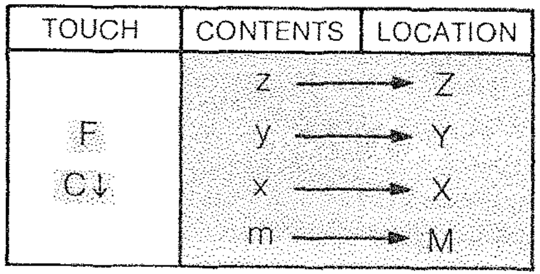
The following diagrams show what happens to the stack for each operation of your calculator. Contents of registers are indicated by lower-case letters x, y and z. Locations are indicated by capital letters X, Y and Z. The display always shows the contents of register X. Memory is register M.

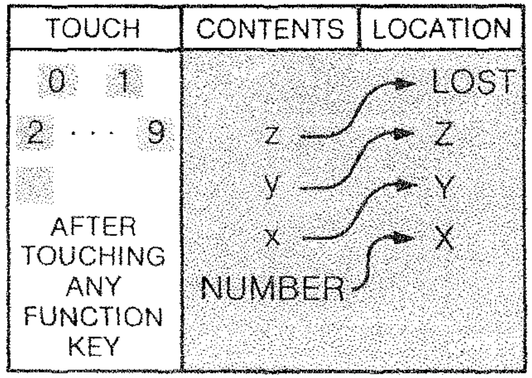


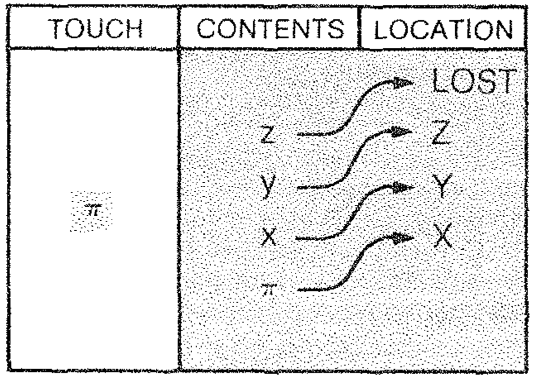


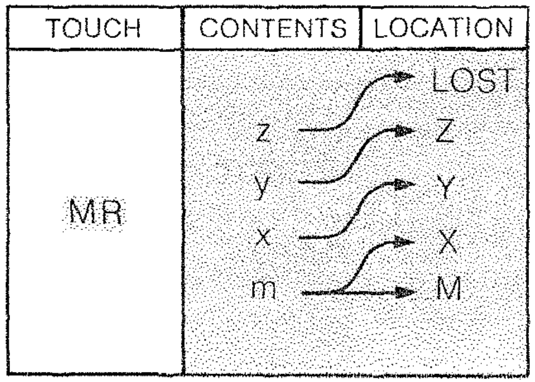

TOUCH
CONTENTS
LOCATION
0 1
2 · · · 9
AFTER TOUCHING
ENT↑
z
y
x
NUMBER
Z
Y
X


TOUCH
CONTENTS
LOCATION
MS
z
y
x
m
Z
Y
X
M
LOST


TOUCH
CONTENTS
LOCATION
F
C↓
z
y
x
m
Z
Y
X
M


TOUCH
CONTENTS
LOCATION
0 1
2 · · · 9
AFTER TOUCHING ANY FUNCTION KEY
z
y
x
NUMBER
LOST
Z
Y
X


TOUCH
CONTENTS
LOCATION
π
z
y
x
π
LOST
Z
Y
X


TOUCH
CONTENTS
LOCATION
MR
z
y
x
m
LOST
Z
Y
X
M

TOUCH	CONTENTS	LOCATION
SIN	0	LOST
COS	z	Z
TAN	y	Y
ln	x	X
LOG	f(x)	LOST
e^x		

TOUCH	CONTENTS	LOCATION
	0	LOST
	z	Z
Y^X	y	Y
	x	X

*Note: Performing any trig, log or antilog function clears register Z. f(x) is transferred to register X, and register Y remains unchanged. Performing the Y^X function clears register Z. The contents of register X are transferred to register Y and Y^X are transferred to register X.

TOUCH	CONTENTS	LOCATION
F	0	LOST
($\sin^{-1}$)	z	Z
or	y	Y
($\cos^{-1}$)	x	X
or	f(x)	LOST
($\tan^{-1}$)		

TOUCH	CONTENTS	LOCATION
	z	Z
1/x	y	Y
$\sqrt{}$	x	X
	f(x)	LOST

TOUCH	CONTENTS	LOCATION
	z	Z
F	y	Y
(X^2)	x	X
	x^2	LOST

TOUCH	CONTENTS	LOCATION
	z	Z
F	y	Y
(rad)	x	X
or	f(x)	LOST
(deg)	(RADIANS TO DEGREES OR DEGREES TO RADIANS)	

TOUCH	CONTENTS	LOCATION
F (M+X²)	z →	Z
	y →	Y
	x →	X
	M+X² →	M
	m →	LOST

TOUCH	CONTENTS	LOCATION
X↔Y	z →	Z
	x →	Y
	y →	X

TOUCH	CONTENTS	LOCATION
ERROR INDICATION	z →	LOST
	0 →	Z
	y →	Y
	0 →	X
	x →	LOST
	m →	M

TOUCH	CONTENTS	LOCATION
+ − × ÷	0 →	Z
	z →	Y
	y, x → f(x) →	X

f(x): $y + x \to X$

$y - x \to X$

$y \times x \to X$

$y \div x \to X$

TOUCH	CONTENTS	LOCATION
F (M+) (M−)	z →	Z
	y →	Y
	x →	X
	x, m → f(x) →	M
	m →	LOST
	f(x): $m + x \to M$	
	$m - x \to M$	

Appendix B — Part 1: Some Examples

In the previous sections of this manual, you have seen a summary of how the functions of your calculator work. This appendix demonstrates the versatility of the machine in a variety of disciplines.

MATHEMATICS

Sum of Products and Product of Sums

Sum of products: $(2 \times 3) + (4 \times 5) = 26$

KEY IN	DISPLAY SHOWS
2	2
ENT	2.
3	3
×	6.
4	4
ENT	4.
5	5
×	20.
+	26.

Product of sums: $(2 + 3) \times (4 + 5) = 45$

KEY IN	DISPLAY SHOWS
2	2
ENT	2.
3	3
+	5.
4	4
ENT	4.
5	5
+	9.
×	45.

See Appendix B — Part 4 for stack diagrams of these examples.

Adding Time, or Degrees, Minutes and Seconds

Add 1 hour, 45 minutes and 45 seconds to 1 hour, 30 minutes and 20 seconds. Enter all times as hh.mmss where hh = hours, mm = minutes and ss = seconds (dd.mmss for degrees, minutes and seconds).

KEY IN	DISPLAY SHOWS	COMMENTS
1.4545	1.4545	1 hour, 45 min., 45 sec.
ENT	1.4545	
1.3020	1.3020	1 hour, 30 min., 20 sec.
+	2.7565	Now, look at the "ss" part of the answer (65). If it is above 60, add .004. Repeat this step until the seconds portion of the answer is below 60.
.004	.004	
+	2.7605	Now, look at the "mm" part of the answer (76). If it is above 60, add .4. Repeat as you did with the seconds portion.
.4	.4	
+	3.1605	Final answer: 3 hours, 16 minutes, 5 seconds.

Degrees, Minutes and Seconds to Decimal Degrees Conversion

Example: Convert the following degrees, minutes and seconds to decimal degrees: 56°23′44.5″

KEY IN	DISPLAY SHOWS	COMMENTS
44.5	44.5	Seconds.
ENT	44.5	
60	60	60 seconds/minute.
MS	60.	
÷	.74166666	
23	23	Minutes.
+	23.741666	

KEY IN	DISPLAY SHOWS	COMMENTS
MR	60.	60 minutes/degree.
÷	0.3956944	
56	56	Degrees.
+	56.395694	Decimal degrees.

Polar to Rectangular Coordinate Conversion

Example: Convert the coordinates $\Theta = 35°$, $R = 7$ to rectangular coordinates using the formulas:

$$X = R \cos \Theta,$$
$$Y = R \sin \Theta.$$

KEY IN	DISPLAY SHOWS	COMMENTS
35	35	Θ.
ENT	35.	
7	7	R.
MS	7	Store R in register M.
x–y	35.	
ENT	35.	Store Θ in register Y.
cos	0.8191521	cos Θ.
MR	7.	Recall R.
×	5.7340647	X displayed = R cos Θ.
x–y	35.	Θ.
sin	0.5735765	sin Θ.
MR	7.	Recall R.
×	4.0150355	Y displayed = R sin Θ.

Note: To see X again, touch x–y.

See Appendix B — Part 4 for a stack diagram of this example.

Rectangular to Polar Coordinate Conversion

Example: Convert the coordinates X = 6, Y = 8 to polar coordinates R and Θ. Using the formula:

$$R = \sqrt{X^2 + Y^2}$$
$$\Theta = \tan^{-1} Y/X.$$

KEY IN	DISPLAY SHOWS	COMMENTS
6	6	X coordinate.
ENT	6.	
ENT	6.	Transfer contents of register Y to register Z to save for use in calculating Θ.
×	36.	
8	8	Y coordinate.
MS	8.	Save for use in calculating Θ.
F (x^2)	64.	
+	100.	$X^2 + Y^2$
$\sqrt{\ }$	10.	R calculated.
x–y	6.	Exchange to bring X coordinate back to register X.
MR	8.	Recall Y coordinate.
x–y	6.	Exchange to divide Y by X.
÷	1.3333333	
F ($\tan^{-1}$)	53.1301	Θ calculated.

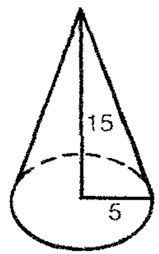
Note: To see R again, touch x–y.

See Appendix B — Part 4 for a stack diagram of this example.

Example: Compute the area of a cone with radius 5 and height 15.

Using the formula: $A = \pi R \sqrt{R^2 + H^2} + \pi R^2$

Substituting: $A = \pi \times 5 \times \sqrt{5^2 + 15^2} + \pi \times 5^2$
$= 326.9045$



KEY IN	DISPLAY SHOWS
π	3.1415926
ENT	3.1415926
5	5
×	15.707963
5	5
F (x^2)	25.
15	15
F (x^2)	225.
+	250.
$\sqrt{\ }$	15.811388
×	248.36469
π	3.1415926
ENT	3.1415926
5	5
F (x^2)	25.
×	78.539815
+	326.9045

Example: Determine the depression of the mercury column in a glass tube of inside diameter 0.6 mm which stands vertically with one end immersed in mercury. The angle of contact with the mercury is 120° and the surface tension is 490 dynes/cm.

Using the formula: $h = 2T/rdg\ (\cos \Theta)$

where: h = height of mercury in tube,
T = surface tension,
r = inside radius of tube (½ diameter),
d = density of the liquid = 13.6 g/cm^3 for mercury,
g = acceleration due to gravity = 980 cm/sec^2.

$$h = \frac{2 \times 490 \text{ dynes/cm}}{0.03 \text{ cm} \times 13.6 \text{ g/cm}^3 \times 980 \text{ cm/sec}^2} \times \cos 120° = -1.225 \text{ cm.}$$

KEY IN	DISPLAY SHOWS	COMMENTS
2	2	
ENT	2.	
490	490	Surface tension.
×	980.	
.03	.03	Inside radius in cm.
ENT	.03	
13.6	13.6	Density of mercury.
×	.408	
980	980	Gravity.
×	399.84	
÷	2.4509803	
120	120	Angle of contact.
cos	−.4999999	
×	−1.2254899	Depression of column in cm.

Example: What is the molarity of a solution that contains 135 grams of calcium chloride, $CaCl_2$, per liter?

Using the formula mass of $CaCl_2$:

$$1\ Ca = 1 \times 40.08\ u = 40.08\ u$$
$$2\ Cl = 2 \times 35.453\ u = 70.906\ u$$
$$110.986\ u = 110.986\ g/mole$$

in the equation: number of moles =

$$\frac{\text{mass of } CaCl_2}{\text{formula mass of } CaCl_2} = \frac{135 \text{ grams}}{110.986 \text{ g/mole}} = 1.21 \text{ moles.}$$

So the concentration of the solution is 1.21 moles per liter.

KEY IN	DISPLAY SHOWS	COMMENTS
40.08	40.08	Atomic mass of Ca.
ENT	40.08	
35.453	35.453	Atomic mass of Cl.
ENT	35.453	
2	2	
×	70.906	Atomic mass of Cl_2.
+	110.986	Formula mass of $CaCl_2$.
135	135	Grams of $CaCl_2$.
x–y	110.986	
÷	1.2163696	Moles/liter.

Example: Calculate the percentage by weight of 10 grams of a substance with normality of 0.15 in 45 milliliters of standard solution with mew of 0.03646.

Using the formula:

$$\%wt = \frac{(mew) \times N \times V \times 10^2}{W}$$

where: %wt = percentage by weight,
mew = millequivalent weight of substance,
N = normality of the substance,
V = volume of standard solution in milliliters, and
W = weight of sample in grams.

Substituting:

$$\%wt = \frac{0.03646 \times 0.15 \times 45 \times 10^2}{10} = 2.46105$$

KEY IN	DISPLAY SHOWS
.03646	.03646
ENT	.03646
.15	.15
×	.005469
45	45
×	.246105
10	10
F (x^2)	100.
×	24.6105
10	10
÷	2.46105

ENGINEERING

Example: What is the equivalent resistance of a 220-ohm resistor, a 145-ohm resistor and a 175-ohm resistor connected in parallel?

Using the equation:

$$R_{eq} = \frac{1}{1/R_1 + 1/R_2 + 1/R_3} = \frac{1}{1/220 + 1/145 + 1/175}$$

KEY IN	DISPLAY SHOWS	COMMENTS
220	220	R_1.
1/x	.00454545	$1/R_1$.
ENT	.00454545	
145	145	R_2.

KEY IN	DISPLAY SHOWS	COMMENTS
1/x	.00689655	$1/R_2$.
+	.011442	
175	175	R_3.
1/x	.00571428	$1/R_3$.
+	.01715628	
1/x	58.287694	$R_{eq} = \frac{1}{1/R_1 + 1/R_2 + 1/R_3}$

Example: What is the tension at the ends of a cable where the span is 700 feet and the sag is 45 feet if each cable of the suspension bridge carries a horizontal load of 620 lbs/ft ?

Using the equation: $T = ½ wa \sqrt{1 + a^2/16d^2}$

where: T = tension,
w = weight (horizontal load),
a = length of span,
d = sag,

$= ½ \times 620 \times 700 \times \sqrt{1 + 700^2/16 \times 45^2}$

$= 871342.25$

KEY IN	DISPLAY SHOWS	COMMENTS
700	700	Length of span (a).
MS	700.	
F (x²)	490000.	a^2.
16	16	
ENT	16.	
45	45	Sag (d).
F (x²)	2025.	d^2.
×	32400	$16d^2$.
÷	15.123456	$a^2/16d^2$.
1	1	
+	16.123456	$1 + a^2/16d^2$.

KEY IN	DISPLAY SHOWS	COMMENTS
√	4.0154023	$\sqrt{1 + a^2/16d^2}$.
MR	700.	a.
×	2810.7816	$a \times \sqrt{1 + a^2/16d^2}$.
620	620	Weight (w).
×	1742684.5	$w \times a \times \sqrt{1 + a^2/16d^2}$.
2	2	
÷	871342.25	$½ \times w \times a \times \sqrt{1 + a^2/16d^2}$.

Example: If the internal pressure of a tank of gas at 295°K is 1500 psi, what is the pressure if the temperature is raised to 303°K?

Using the formula:

$$P_2 = \frac{P_1 T_2}{T_1} = \frac{1500 \times 303}{295} = 1540.6779 \text{ psi.}$$

KEY IN	DISPLAY SHOWS
1500	1500
ENT	1500.
303	303
×	454500.
295	295
÷	1540.6779

Example: What is the equivalent impedance of a 325-ohm resistor and a 15.2-millihenry inductor at a frequency of 1500 Hz?

Using the formula: $Z_{eq} = R/\Theta$

where: $\Theta = \arctan \frac{2\pi fL}{R}$

$= \arctan \frac{2 \times \pi \times 1500 \times .0152}{325}$

$= 23.78739°$ and

$$R = \frac{2\pi fL}{\sin \Theta} = 355.17239$$

KEY IN	DISPLAY SHOWS	COMMENTS
2	2	
ENT	2.	
π	3.1415926	
×	6.2831852	
1500	1500	
×	9424.7778	
.0152	.0152	
×	143.25662	
MS	143.25662	Since you're going to use $2\pi fL$ again to calculate R, store it for further use.
325	325	
÷	.4407896	
F ($\tan^{-1}$)	23.78739	Θ calculated.
sin	.4033439	
MR	143.25662	Recall $2\pi fL$.
x–y	.4033439	Exchange X and Y registers so you can divide what was last in display by what is now in display.
÷	355.17239	R calculated.

STATISTICS

Example: Compute the mean ($\overline{x}$) of the following data: (2, 7, 3, 5, 2).

Using the formula:

$$\overline{x} = \frac{\Sigma x}{n}$$

KEY IN	DISPLAY SHOWS	COMMENTS
2	2	x_1.
ENT	2.	
7	7	x_2.
+	9.	
3	3	x_3.
+	12.	Repeat these steps n-1 times.
5	5	x_4.
+	17.	
2	2	x_5.
+	19.	
5	5	n
÷	3.8	Mean ($\overline{x}$).

Example: Compute the harmonic mean (M_h) of the following data: (2, 7, 3, 5, 2).

Using the formula:

$$M_h = \frac{n}{\Sigma\, 1/x}$$

KEY IN	DISPLAY SHOWS	COMMENTS
2	2	x_1.
1/x	.5	
7	7	x_2.
1/x	.14285714	
+	.64285714	
3	3	x_3.
1/x	.33333333	Repeat these steps n-1 times.
+	.97619047	
5	5	x_4.
1/x	.2	
+	1.1761904	
2	2	x_5.
1/x	.5	
+	1.6761904	
5	5	n
x⇌y	1.6761904	
÷	2.9829546	Harmonic mean (M_h).

Mean, Variance and Standard Deviation

Example: Find the mean, variance and standard deviation of the following values: (2, 7, 3, 5, 2).

Using the formulas:

$$\text{Mean} = \overline{x} = \frac{\Sigma x}{n}$$

$$\text{Variance} = \sigma^2 = \frac{\Sigma x^2 - n(\Sigma x/n)^2}{n-1}$$

$$\text{Standard Deviation} = \sigma = \sqrt{\sigma^2}$$

KEY IN	DISPLAY SHOWS	COMMENTS
0	0	
MS	0.	Clear memory.
2	2	x_1.
F (M+x^2)	2.	
ENT	2.	
7	7	x_2.
F (M+x^2)	7.	
+	9.	
3	3	x_3.
F (M+x^2)	3.	Repeat these steps n-1 times.
+	12.	
5	5	x_4.
F (M+x^2)	5.	
+	17.	
2	2	x_5.
F (M+x^2)	2.	
+	19.	
5	5	n
÷	3.8	Mean ($\overline{x}$).
F (x^2)	14.44	
CHS	−14.44	
5	5	n.
×	−72.2	
MR	91	
+	18.8	
4	4	n-1.
÷	4.7	Variance.
√	2.1679483	Standard deviation.

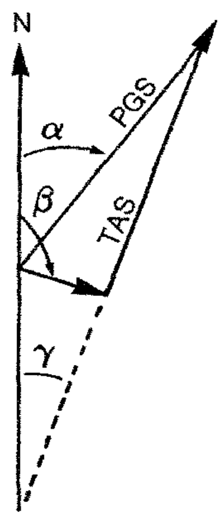
Example: Compute the geometric mean (M_g) of the following data: (2, 7, 3, 5, 2).

Using the formula:

$$M_g \sqrt[n]{(x_1)\ (x_2)\ (x_3)\ \ldots\ (x_n)}$$

KEY IN	DISPLAY SHOWS	COMMENTS
2	2	x_1.
ENT	2.	
7	7	x_2.
×	14.	
3	3	x_3.
×	42.	Repeat these steps n-1 times.
5	5	x_4.
×	210.	
2	2	x_5.
×	420.	
5	5	n
1/x	.2	n^{th} root.
y^x	3.346952	Geometric mean (M_g).

NAVIGATION



Example: Find the predicted ground speed and true heading for a planned flight with the following flight trangle factors known:

$\measuredangle\alpha$ = true course = 30° from North.
$\measuredangle\beta$ = wind direction = 50° from North.
TAS = true air speed = 140 mph.
V = wind velocity = 42 mph.
$\measuredangle\gamma$ = true heading = ?
PGS = predicted ground speed = ?

Predicted Ground Speed

Using the equation:

$$PGS = V\cos(\beta-\alpha) + \sqrt{[V\cos(\beta-\alpha)]^2 - V^2 + TAS^2}$$
$$= 42\cos(50-30) + \sqrt{[42\cos(50-30)]^2 - 42^2 + 140^2}.$$

KEY IN	DISPLAY SHOWS	COMMENTS
42	42	Wind velocity.
ENT	42.	
50	50	Wind direction ($\measuredangle\beta$).
ENT	50.	
30	30	True course ($\measuredangle\alpha$).
−	20.	
cos	.9396927	
×	39.467093	V cos ($\beta-\alpha$).
MS	39.467093	Store for further use.
F (x^2)	1557.6514	$[V\cos(\beta-\alpha)]^2$.
42	42	V.
F (x^2)	1764.	V^2.

KEY IN	DISPLAY SHOWS	COMMENTS
−	−206.3486	$[V\cos(\beta-\alpha)]^2 - V^2$.
140	140	TAS.
F (x^2)	19600.	TAS^2.
+	19393.652	$[V\cos(\beta-\alpha)]^2 - V^2 + TAS^2$.
$\sqrt{\ }$	139.26109	$\sqrt{[V\cos(\beta-\alpha)]^2 - V^2 + TAS^2}$.
MR	39.467093	$V\cos(\beta-\alpha)$.
+	178.72818	$V\cos(\beta-\alpha) + \sqrt{[V\cos(\beta-\alpha)]^2 - V^2 + TAS^2} =$ Predicted ground speed.

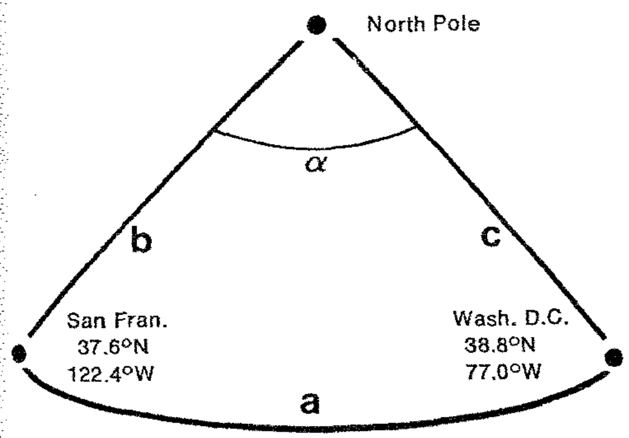
True Heading

Using the equation:

$$\measuredangle\lambda = \alpha - \text{arc sin}\,[V\sin(\beta-\alpha)/TAS]$$
$$= 30 - \text{arc sin}\,[42\sin(50-30)/140]$$

KEY IN	DISPLAY SHOWS	COMMENTS
30	30	True course.
MS	30.	Store.
42	42	Wind velocity (V).
ENT	42.	
50	50	Wind direction (β).
ENT	50.	
MR	30.	Recall α.
−	20.	$\beta-\alpha$
sin	.3420202	Sin $(\beta-\alpha)$.
×	17.10101	V sin $(\beta-\alpha)$.
140	140	TAS.
÷	.12215007	V sin $(\beta-\alpha)$/TAS.

KEY IN	DISPLAY SHOWS	COMMENTS
F ($\sin^{-1}$)	7.016205	arc sin [V sin $(\beta-\alpha)$/TAS].
MR	30.	α.
x-y	7.016205	
−	22.983795	α−arc sin [V sin $(\beta-\alpha)$/TAS].



Example: What is the great circle route between San Francisco and Washington, D.C.?

Using the formula:

$$a = \text{arc cos}(\cos b \cos c + \sin b \sin c \cos\alpha) \times 60$$

where: $\alpha = 122.4° - 77.0° = 45.4°$,
$b = 90° - 37.6° = 52.4°$, and
$c = 90° - 38.8° = 51.2°$.

$$a = \text{arc cos}\,(\cos 52.4 \cos 51.2 + \sin 52.4 \sin 51.2 \cos 45.4) \times 60.$$

KEY IN	DISPLAY SHOWS	COMMENTS
52.4	52.4	b.
cos	.6101452	cos b.
51.2	51.2	c.
cos	.6266039	cos c.
×	.38231936	cos b cos c.
MS	.38231936	Store in memory.

KEY IN	DISPLAY SHOWS	COMMENTS
52.4	52.4	b.
sin	.7922897	sin b.
51.2	51.2	c.
sin	.779338	sin c.
×	.61746147	sin b sin c.
45.4	45.4	α.
cos	.7021531	cos α.
×	.43355248	sin b sin c cos α.
MR	.38231936	Recall memory.
+	.81587184	cos b cos c + sin b sin c cos α.
F	.81587184	
($\cos^{-1}$)	35.32634	arc cos (cos b cos c + sin b sin c cos α).
60	60	
×	2119.5804	Great circle distance.

FINANCE

Example: How much do you have to put in the bank for it to be worth $25,000 in 10 years if the interest rate is 8.5% per year?

Using the formula: $PV = \dfrac{FV}{(1+i)^n}$

where: PV = present value,
FV = future value,
i = interest rate (in decimal),
n = number of years.

Substituting: $PV = \dfrac{25000}{(1+.085)^{10}} = \$11057.15$

KEY IN	DISPLAY SHOWS	COMMENTS
1	1	
ENT	1.	
.085	.085	i

KEY IN	DISPLAY SHOWS	COMMENTS
+	1.085	(1+i)
10	10	n
y^x	2.26098	$(1+i)^n$
25000	25000	
x–y	2.26098	
÷	11057.152	Present Value (PV).

Example: What will $7,000 be worth in 5 years if it is compounded annually at a rate of 8.2% per year?

Using the formula: $FV = PV(1+i)^n$
$= 7000\ (1+.082)^5$

KEY IN	DISPLAY SHOWS	COMMENTS
1	1	
ENT	1.	
.082	.082	i.
+	1.082	
5	5.	n.
y^x	1.482982	$(1+i)^n$.
7000	7000	PV.
×	10380.874	Future value (FV).

Example: Compute the annual rate of return (after taxes) on an investment of $10,000 which after 3½ years is worth $12,550 if the tax rate is 38%.

Using the formula:

$$r = \frac{(FV - PV)\ (1 - \text{tax rate})}{PV} \times n$$

where: r = rate of return,
FV = future value,
PV = present value,
n = number of periods.

KEY IN	DISPLAY SHOWS	COMMENTS
12550	12550	FV.
ENT	12550.	
10000	10000	PV.
MS	10000.	Save for use in dividing.
−	2550.	FV − PV.
1	1	
ENT	1.	
.38	.38	Tax rate.
−	.62	1 − tax rate.
×	1581.	(FV − PV) (1 − tax rate).
MR	10000.	Recall PV.
÷	.1581	$\frac{(FV - PV)(1 - \text{tax rate})}{PV}$
3.5	3.5	n.
×	.55335	$\frac{(FV - PV)(1 - \text{tax rate})}{PV} \times n.$
100	100	
×	55.335	Multiply by 100 to make into whole percentage = rate of return.

Part 1.

What is the annual payment on a loan of $86,000 taken for 10 years if the rate is 8% per year?

Using the formula:

$$PMT = PV \left[\frac{i}{1 - (1 + i)^{-n}} \right]$$

where: PMT = payment,
PV = present value,
i = interest rate per period (in decimal),
n = number of periods.

KEY IN	DISPLAY SHOWS	COMMENTS
1	1	
ENT	1.	
.08	.08	i.
+	1.08	(1 + i).
10	10	n.
CHS	−10	
y^x	.4631941	$(1 + i)^{-n}$.
CHS	−.4631941	
1	1	
+	.5368059	$1 - (1 + i)^{-n}$.
.08	.08	
x-y	.5368059	
÷	.14902965	$\frac{i}{1 - (1 + i)^{-n}}$
86000	86000	PV.
×	12816.549	PMT.

See Appendix B — Part 4 for a stack diagram of this example.

Part 2.

In the above example (part 1), what is the remaining balance after the 6th payment?

Using the formula:

$$BAL_k = PMT \left[\frac{1 - (1 + i)^{k-n}}{i} \right]$$

where: k = number of payments made.

KEY IN	DISPLAY SHOWS	COMMENTS
1	1	
ENT	1.	
.08	.08	i.

KEY IN	DISPLAY SHOWS	COMMENTS
MS	.08	Store for further use.
+	1.08	$1 + i$.
6	6	k.
ENT	6.	
10	10	n.
—	−4.	$k - n$.
y^x	.7350307	$(1 + i)^{k-n}$.
CHS	−.7350307	
1	1	
+	.2649693	$1 - (1 + i)^{k-n}$.
MR	.08	Recall i.
÷	3.3121162	$\frac{1 - (1 + i)^{k-n}}{i}$
12816.55	12816.55	PMT (from part 1).
×	42449.902	Bal_k.

Appendix B — Part 2: Hyperbolic and Inverse Hyperbolic Functions

The hyperbolic and inverse hyperbolic functions can be found by using the Gudermannian function:

$$gd\ x = 2 \text{ arc tan } e^x - \pi/2 \quad (\text{Note: } \pi/2 = 90°).$$

and the inverse Gudermannian function:

$$gd^{-1}\ x = \ln \tan [\pi/4 + x/2] \quad (\text{Note: } \pi/4 = 45°).$$

in conjunction with the following formulas:

$$\sinh x = \frac{e^x - e^{-x}}{2},$$

$$\cosh x = \frac{e^x + e^{-x}}{2},$$

$$\tanh x = \frac{\sinh x}{\cosh x} = \sin gd\ x,$$

$$\coth x = \frac{1}{\tanh x},$$

$$\text{sech } x = \frac{1}{\cosh x},$$

$$\text{csch } x = \frac{1}{\sinh x}.$$

$$\sinh^{-1} x = \ln [x + \sqrt{(x^2 + 1)}] = gd^{-1}(\sin^{-1} x),$$

$$\cosh^{-1} x = \text{sech}^{-1}\ 1/x,$$

$$\tanh^{-1} x = 1/2 \ln [1 + x/1 - x] = gd^{-1}(\sin^{-1} x),$$

$$\coth^{-1} x = \tanh^{-1} 1/x,$$

$$\text{sech}^{-1} x = [\ln 1/x + \sqrt{1/x^2 - 1}] = gd^{-1}(\cos^{-1} x),$$

$$\text{csch}^{-1} x = \sinh^{-1} 1/x.$$

Examples:

Gudermannian function: gd 0.225 = 12.7841.
Key in: .225 e^x F ($\tan^{-1}$) 2 × 90 —
Display shows: 12.7841

Inverse Gudermannian function: gd^{-1} 60° = 1.316958.
Key in: 60 ENT 2 ÷ 45 + tan ln
Display shows: 1.316958.

Hyperbolic sine: sinh 2.5 = 6.0502025.
Key in: 2.5 e^x ENT 1/x — 2 ÷
Display shows: 6.0502025.

Hyperbolic cosine: cosh 2.5 = 6.1322875.
Key in: 2.5 e^x ENT 1/x + 2 ÷
Display shows: 6.1322875.

Hyperbolic tangent: tanh 2.5 = .9866143.
Key in: 2.5 e^x F ($\tan^{-1}$) 2 × 90 — sin
Display shows: .9866143.

Hyperbolic cotangent: coth 2.5 = 1.0135673.
Key in: 2.5 e^x F ($\tan^{-1}$) 2 × 90 — sin 1/x
Display shows: 1.0135673.

Hyperbolic secant: sech 2.5 = .16307128.
Key in: 2.5 e^x ENT 1/x + 2 ÷ 1/x
Display shows: .16307128.

Hyperbolic cosecant: csch 2.5 = .16528372.
Key in: 2.5 e^x ENT 1/x − 2 ÷ 1/x
Display shows: .16528372.

Inverse hyperbolic sine: $\sinh^{-1} 30 = 4.094624$.
Key in: 30 F ($\tan^{-1}$) 2 ÷ 45 + tan ln
Display shows: 4.094624.

Inverse hyperbolic tangent: $\tanh^{-1} .52 = .5763396$.
Key in: .52 F ($\sin^{-1}$) 2 ÷ 45 + tan ln
Display shows: .5763396.

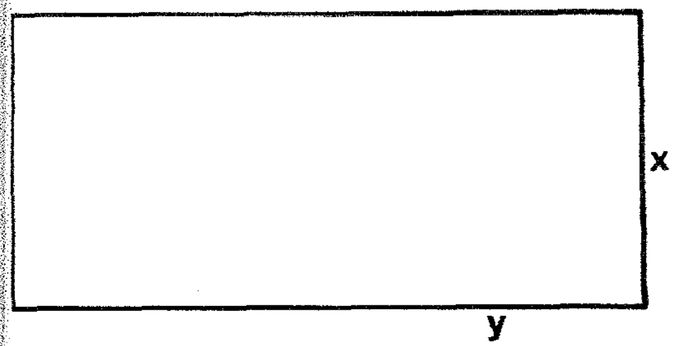
Inverse hyperbolic secant: $\text{sech}^{-1} .52 = 1.271361$.
Key in: .52 F ($\cos^{-1}$) 2 ÷ 45 + tan ln
Display shows: 1.271361.

Inverse hyperbolic cosine: $\cosh^{-1} 30 = 4.094066$.
Key in: 30 1/x F ($\cos^{-1}$) 2 ÷ 45 + tan ln
Display shows: 4.094066.

Inverse hyperbolic cotangent: $\coth^{-1} 30 = 0.0333458$.
Key in: 30 1/x F ($\sin^{-1}$) 2 ÷ 45 + tan ln
Display shows: .0333458.

Inverse hyperbolic cosecant: $\text{csch}^{-1} .52 = 1.408696$.
Key in: .52 1/x F ($\tan^{-1}$) 2 ÷ 45 + tan ln
Display shows: 1.408696.

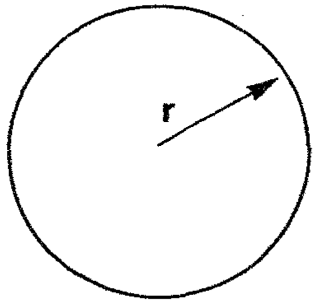
Appendix B — Part 3: Some Common Mathematical Formulae with Examples



Rectangle, area and perimeter

Rectangle of width X and length Y
Area = XY Perimeter = 2X + 2Y

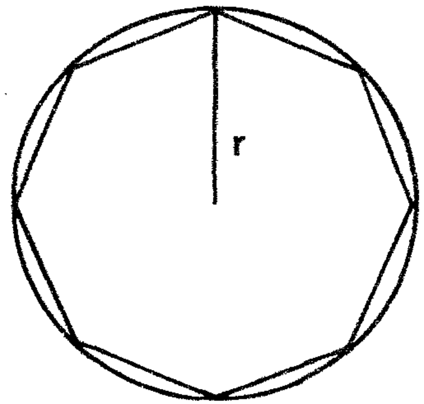
Example: Rectangle of width 4 and length 8:
Area: Key in: 4 ENT 8 ×
Display shows: 32.
Perimeter: Key in: 2 ENT 4 × 2 ENT 8 × +
Display shows: 24.



Circle, area and circumference

Circle of radius r.
Area = πr^2 Circumference = $2\pi r$

Example: Circle of radius 5.
Area: Key in: π ENT 5 F (x^2) ×
Display shows: 78.539815
Circumference: Key in: 2 ENT π × 5 ×
Display shows: 31.415926



Regular polygon circumscribed in a circle, area and perimeter

Regular polygon with n sides inscribed in a circle of radius r.

Area = $\frac{1}{2}nr^2 \sin 360/n$ Perimeter = $2nr \sin 180/n$

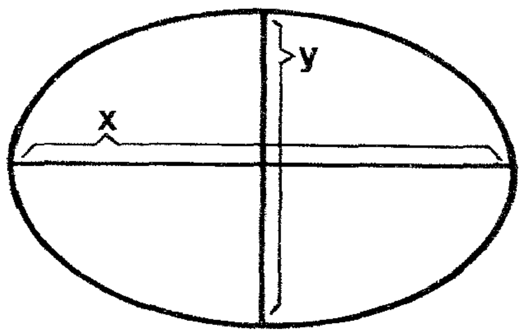
Example: Polygon with eight sides inscribed in a circle of radius 5.

Area: Key in: 1 ENT 2 ÷ 8 × 5 F (x^2) × 360 ENT 8 ÷ sin ×

Display shows: 70.71068

Perimeter: Key in: 2 ENT 8 × 5 × 180 ENT 8 ÷ sin ×

Display shows: 30.614672



Ellipse, area and circumference

Ellipse of major axis X and minor axis Y.

Area = $\frac{1}{4}\pi XY$ Circumference = $2\pi\sqrt{1/8(X^2+Y^2)}$

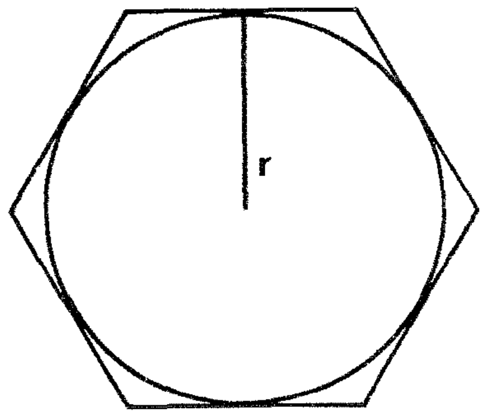
Example: Ellipse of major axis 8 and minor axis 4.

Area: Key in: 1 ENT 4 ÷ π × 8 × 4 ×

Display shows: 25.13274

Circumference: Key in: 8 F (x^2) 4 F (x^2) + 8 ÷ √ 2 × π ×

Display shows: 19.869175



Regular polygon circumscribing a circle, area and perimeter

Regular polygon with n sides circumscribing a circle of radius 5.

Area = $nr^2 \tan 180/n$ Perimeter = $2nr \tan 180/n$

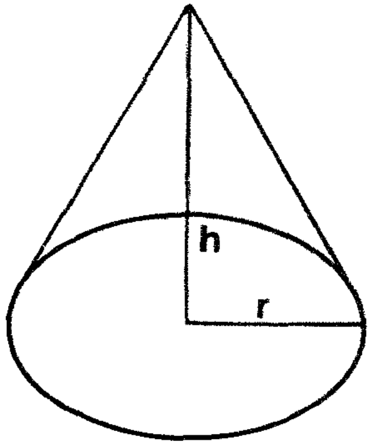
Example: Polygon with eight sides circumscribing a circle of radius 5.

Area: Key in: 8 ENT 5 F (x^2) × 180 ENT 8 ÷ sin ×

Display shows: 76.53668

Perimeter: Key in: 2 ENT 8 × 5 × 180 ENT 8 ÷ tan ×

Display shows: 33.13708



Cone, area and volume

Cone of radius r and height h.

Volume = $\frac{1}{3}\pi r^2 h$ Area = $\pi r\sqrt{r^2 + h^2}$

Example: Cone of radius 5 and height 10.

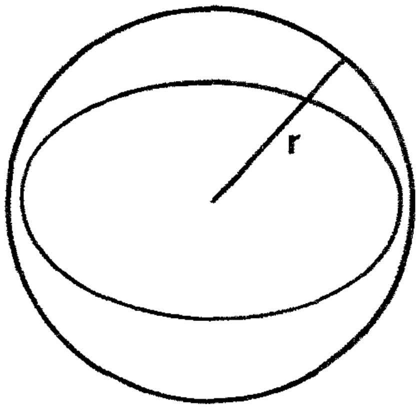
Volume: Key in: π 5 F (x^2) × 10 × 3 ÷

Display shows: 261.79938

Area: Key in: 5 MS F (x^2) 10 F (x^2) +

√ MR × π ×

Display shows: 175.62035



Sphere, area and volume

Sphere of radius r.

Volume = $4/3\pi r^3$ Area = $4\pi r^2$

Example: Sphere of radius 5.

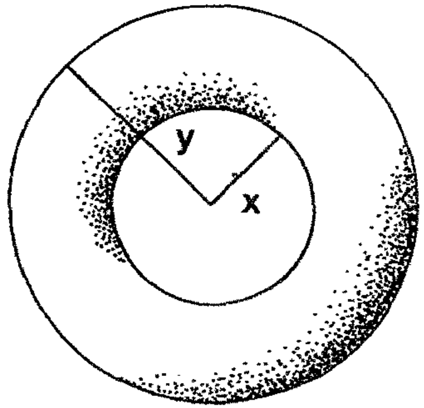
Volume: Key in: 5 ENT 3 y^x π × 4 ENT

3 ÷ ×

Display shows: 523.59833

Area: Key in: 4 ENT π × 5 F (x^2) ×

Display shows: 314.15925



Torus, area and volume

Torus of inner radius x and outer radius y.

Volume = $\frac{1}{4}\pi^2(x+y\ (y-x)^2$ Area = $\pi^2(y^2-x^2)$

Example: Torus with inner radius 2 and outer radius 4.

Volume: Key in: π F (x^2) 2 ENT 4 + ×

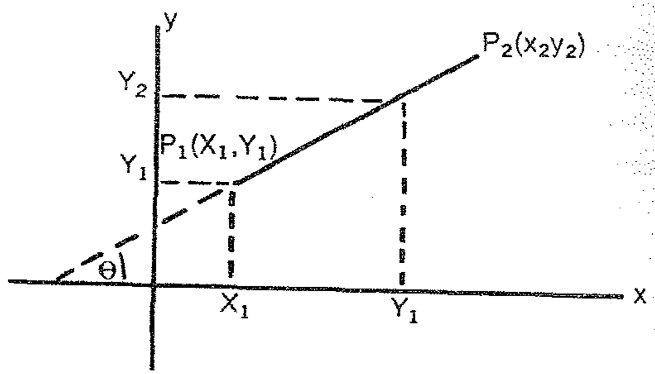
4 ENT 2 − F (x^2) × 4 ÷

Display shows: 59.217622

Area: Key in: π F (x^2) 4 F (x^2) 2

F (x^2) − ×

Display shows: 118.43524



Distance between two points, P_1 and P_2

Distance d between two points $P_1(x_1y_1)$ and $P_2(x_2y_2)$.

$$d = \sqrt{(x_2-y_1{}^2 + (y_2-y_1)^2}$$

Example: Distance between points $P_1(3,4)$ and $P_2(5,8)$.

Key in: 5 ENT 3 − F (x^2) 8 ENT 4 − F (x^2) + √

Display shows: 4.4721359

Slope and angle of line between points

Slope and angle of line between points P_1 and P_2.

Slope $= m = y_2-y_1/x_2-x_1 = \tan\theta$

Example:

Slope: Key in: 8 ENT 4 − 5 ENT 3 − ÷

Display shows: 2.

Angle: Key in: F $(\tan^{-1})$

Display shows: 63.43495

Appendix B — Part 4: Stack Diagrams for Some Examples

Stack diagram for: $(2 \times 3) + (4 \times 5) = 26$

	REGISTER CONTENTS			
KEY IN	X	Y	Z	M
2	2			
ENT	2	2		
3	3	2		
×	6			
4	4	6		
ENT	4	4	6	
5	5	4	6	
×	20	6		
+	26			

Stack diagram for: $(2+3) \times (4+5) = 45$

KEY IN	REGISTER CONTENTS X	Y	Z	M
2	2			
ENT	2	2		
3	3	2		
+	5			
4	4	5		
ENT	4	4	5	
5	5	4	5	
+	9	5		
×	45			

Stack diagram for: $X = R \cos \Theta$, $Y = R \sin \Theta$

KEY IN	REGISTER CONTENTS X	Y	Z	M
Θ	Θ			
ENT	Θ	Θ		
R	R	Θ		
MS	R	Θ		R
x–y	Θ	R		R
ENT	Θ	Θ	R	R
cos	cos Θ	Θ	R	R
MR	R	cos Θ	Θ	R
×	X=R cos Θ	Θ	R	R
x–y	Θ	R cos Θ	R	R
sin	sin Θ	R cos Θ	R	R
MR	R	sin Θ	R cos Θ	R
×	Y=R sin Θ	R cos Θ	R	R

Stack diagram for: $R = \sqrt{6^2 + 8^2}$ and $\Theta = \tan(8/6)$

KEY IN	REGISTER CONTENTS X	Y	Z	M
6	6			
ENT	6	6		
ENT	6	6	6	
×	36	6		
8	8	36	6	
MS	8	36	6	8
F	8	36	6	8
x^2	64	36	6	8
+	100	6		8
√	10	6		8
x–y	6	10		8
MR	8	6	10	
x–y	6	8	10	
÷	1.33	10		
F	1.33	10		
$\tan^{-1}$	53.13			

Stack diagram for: $PMT = PV \left[\frac{i}{1-(1+i)^{-n}}\right]$

KEY IN	REGISTER CONTENTS X	Y	Z	M
1	1			
ENT	1	1		
i	i	1		
+	1+i			
n	n	1+i		
CHS	−n	1+i		
y^x	$(1+i)^{-n}$			
CHS	$-(1+i)^{-n}$			
1	1	$-(1+i)^{-n}$		
+	$1-(1+i)^{-n}$			
i	i	$1-(1+i)^{-n}$		
x–y	$1-(1+i)^{-n}$	i		
÷	$\frac{i}{1-(1+i)^{-n}}$			
PV	PV	$\frac{i}{1-(1+i)^{-n}}$		
×	$PV \left[\frac{i}{1-(1+i)^{-n}}\right]$			

Appendix C — Conditions for Error Indication

FUNCTION	CONDITION (X=contents of register X)
+, −, ×, ÷	$X > 99999999$
÷, 1/x	$\|X\| \leqslant 0.00000001$
$\sqrt{x}$	$X < 0$
Y^x	$Y \leqslant 0$; $18.42060 < X \ln Y < -28$
LOG X, Ln x	$X \leqslant 0$
e^x	$18.42068 < X < -28$
SIN, COS	$X \geqslant 7$ radians, $X \geqslant 401°$
TAN	$\|X\| \geqslant 90°$, $X \geqslant 7$ radians
SIN^{-1}, COS^{-1}	$X > 1$
TAN^{-1}	$X > 99999999$

WARRANTY INFORMATION

The Consumer Products Division of National Semiconductor Corporation is proud to give you the following Warranty on your calculator.

Full 90-Day Warranty

National Semiconductor gives you a full ninety-day Warranty from the date of original purchase on all defects in material and workmanship.
Should the calculator prove to have such defects within ninety days of original purchase, National Semiconductor will repair, or, at its discretion, replace it with a new calculator, all without charge, within a reasonable period of time after receipt by National Semiconductor, postage prepaid. This FULL WARRANTY does not apply to defects or malfunctions caused by abuse, accident, modifications, negligence, or any other damage not resulting from defects in materials or workmanship or beyond the control of National Semiconductor. THIS FULL WARRANTY IS IN LIEU OF ANY CLAIM BY THE CONSUMER FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES and shall apply to the purchaser or the purchaser's transferee, so long as the mailing instructions below are followed.

Limited 275-Day Warranty

National Semiconductor also offers a LIMITED WARRANTY in addition to the above FULL WARRANTY which shall apply to any defects in material or workmanship which occur during the 275-day period after the 90-day period covered by the FULL WARRANTY, excluding the battery, which is not a component part of your calculator. Under this LIMITED WARRANTY, National Semiconductor, for a service charge of U.S. $3.50, and for no additional charge will repair such defects or, at its discretion, replace your calculator with one that is identical or reasonably equivalent, all within a reasonable period of time after receipt by National Semiconductor, postage prepaid. This LIMITED WARRANTY does not apply to defects or malfunctions caused by abuse, accident, modifications, negligence, or any other damage not resulting from defects in materials or workmanship or beyond the control of National Semiconductor. THIS LIMITED WARRANTY IS IN LIEU OF ANY CLAIM BY THE CONSUMER FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES and shall apply to the purchaser or the purchaser's transferee, so long as the mailing instructions below are followed. THE DURATION OF ALL IMPLIED WARRANTIES ON THIS CALCULATOR WILL BE THE SAME AS, AND NO LONGER THAN, THE 365-DAY PERIOD OF THE FULL AND LIMITED WARRANTIES COMBINED.

Post Warranty Repairs

Even though your Warranty may have expired, if your calculator becomes defective after the 365-day warranty period, National Semiconductor will make repairs for a nominal charge of $15.50, providing the mailing instructions below are followed. Repair prices during the post-warranty period are subject to change without notice.

Product Service Locations

United States
N.C.P.S.—East Coast
Commerce Park
Danbury, CT 06810

Canada
N.C.P.S.
286 Wildcat Road
Downsview
Ontario M3J-2N5
Canada

Asia
NS Electronics SDN BHD
National Semiconductor
Product Service
Bayan Lepas
Free Trade Zone
Penang, Malaysia

Scotland
NS—UK Ltd.
National Semiconductor
Product Service
Larkfield Industrial Estate
Greenock PA16 OEQ,
Scotland

Germany
National Semiconductor GmbH
Product Service
D808 Furstenfeldbruck
Industriestrasse 10
Bundesrepublik
Deutschland

Mailing Instructions

Should your calculator need servicing, pack it carefully in a sturdy box for shipping. Proof of original purchase date must be enclosed. Be sure to include your name and return address. The package should be mailed postpaid to the nearest National Semiconductor Service Center. If your calculator is returned for warranty repairs more than ninety days after the original purchase date, you must enclose the appropriate service charge (if the service charge during the POST WARRANTY period has been changed, National Semiconductor will request you to supply the additional amount, if any is needed, or make the appropriate refund, if there is any difference, by check or money order payable to National Semiconductor).

Consumer Warranty Registration Certificate

Please put your warranty into effect by completing this form and mailing it within 10 days from date of purchase to the NOVUS service center in your area.

Novus Model 4615

Serial Number____________________

Purchase Date____________________
(month/day/year)

Purchased from____________________

Address____________________

City, State, Zip____________________

Your Name____________________

Your Address____________________

City, State, Zip____________________

Was this calculator purchased for:
- ☐ Gift ☐ Personal use

What is your occupation?
- ☐ Student or Teacher ☐ Professional
- ☐ Executive ☐ Financial or Commercial
- ☐ Engineering or Scientific ☐ Statistical fields
- ☐ Other occupation________________

What is your age group?
- ☐ Under 18 ☐ 18-34 ☐ 35-49 ☐ 50-over

Where will you most use your Novus calculator?
- ☐ At home ☐ At school ☐ At work
- ☐ During travel

Where did you learn about the Novus calculators?
- ☐ Magazine ☐ Newspaper ☐ Television
- ☐ Radio ☐ Mail ☐ Store salesman
- ☐ Friend
- ☐ Other________________

What most attracted you to your Novus calculator?
- ☐ Appearance ☐ Size ☐ Reputation
- ☐ Price ☐ Features and capabilities

NOVUS Warranty Certificate
Please retain for your records. See insert for trouble-shooting tips and product service locations.

Model Number________________

Serial Number________________

Purchased from________________

Date purchased________________

Other "professional" calculators from National Semiconductor...

National Semiconductor 4615 Mathematician PR

The Programmable Electronic Slide Rule

- Trig and inverse trig functions
- Common and natural logs and anti-logs
- Fully addressable, accumulating memory
- 100-step programming capability

National Semiconductor 4520 Scientist

The Scientist's Electronic Slide Rule

- Scientific notation
- Trig and inverse trig functions
- Common and natural logs and anti-logs

National Semiconductor 4525 Scientist PR

Scientist's Programmable Electronic Slide Rule

- Same features as National Semiconductor 4520
- 100-step programming capability

National Semiconductor 6025 Financier PR

Programmable Electronic Financial Calculator

- Dedicated to solving financial calculations
- Pre-programmed financial equations
- Fully addressable, accumulating memory
- 100-step programming capability

National Semiconductor Statistician PR

Programmable Electronic Statistical Calculator

- Dedicated to solving statistical calculations
- Pre-programmed statistical equations
- Fully addressable, accumulating memory
- 100-step programming capability

National Semiconductor AC adaptors and chargers are also available.

For further information see your dealer or write:

CUSTOMER RELATIONS DEPT.
1177 Kern Avenue
Sunnyvale, CA 94086
(408) 732-5000

NATIONAL SEMICONDUCTOR

101814-A