Keyboard

This is a list of the keys available to the calculator:

Numeric keys (0-9)	
Decimal point	'.'
Parentheses	'(' or ')'
Multiply symbol	'*'
Division symbol	'/'
Addition symbol	'+'
Subtraction symbol	'-'
Power symbol	'^'
Enter key	'=' or <CR>
Alphabetic keys	(all displayable keys, available only in macros).

Fingers N Toes

Help Index

For information on how to use help,
press F1 or choose Help Using Help.

Extras

Conversions

Fingers N Toes has over 30 built in conversions. Any conversion can be selected from the **conversion** dialog box and assigned to the conversion key cnv.

To change the current conversion select the Conversions menu item from the Extras menu. A dialog box appears displaying the conversions available. Select the conversion desired and it is automatically assigned to the cnv key.

Constants

Fingers N Toes has over a dozen built in constants. Select any constant from the **konstant** dialog box for and assignment to the constant key k.

To change the current constant, select the Konstants menu item from the Extras menu. A dialog box appears displaying the available constants. Select the constant desired and it is automatically assigned to the k key.

Macros

Macros allow you to program your calculator to perform repetitious tasks.   You can record up to one hundred macros.   Press Escape to stop a macro in progess.   All macros are saved automatically to disk and are available every time Fingers N Toes is invoked.

Record

Play

View

Delete

Example

Supplied Macros

Record Macro

To record a macro, select Record from the Macro menu. A dialog box appears in which you enter the name of the macro. Entering a duplicate name replaces the existing macro with that name.

Each keystroke or button press is recorded in the macro. The record function beeps each time it records a step. The display does not update during calculations.

To stop recording, select **Stop Recording** from the Macro menu.

Play macro

To execute a macro, select Play from the Macros menu.   A dialog box appears with the available macros.   Select the macro you want to play and press OK.

View macro

To view a macro, select View from the Macros menu.   A dialog box appears with a list of the available macros.   Double click on the macro you want to view.

The view macro dialog box also provides a printing function.   Select a macro using the mouse and the click on the Print button.   The macro sent to the printer is formated for easy reading.

Delete macro

To delete a macro, select Delete from the Macros menu. A dialog box appears with a list of the available macros. Select the macro to delete and press the Delete button. Macros are not deleted until you confirm by pressing the OK button.

Example macro

This example shows how to start a macro, display a message, wait for user input, and perform a conditional jump.

Example: create a macro that selects a whole random number between 1 and 10, inclusive, and asks the user to guess the number, keeping track of the number of guesses.

select Record from the Macro menu.   Enter "GUESS IT" as the name of this macro

1: AC clear the calculator

2: View Scientific select scientific view to assure that rnd is available

3: rnd select a random number between 0 and 1

4: * multiply by 9 and add 1 to get a value

5: 9 between 1 and 10, inclusive

6: +

7: =

8: Mode Decimal switch to decimal view to force an integer conversion

9: sto 0 0 store the value in register 00

10: 0 initialize attempt counter

11: sto 0 1 store the attempt counter in register 01

12: msg **Guess a number between 1 and 10**

13: inp wait for the user to enter a value and press ENTER

14: rcl 0 1 recall register 01

15: + add 1 to the attempt counter

16: 1

17: =

18: sto 0 1 store the attempt counter back in register 01

19: tst **if x <> RCL00 goto013**
if the displayed value does not equal the secret number the macro continues at step 13

20: bep alerts the user to success

21: msg **You guessed it, It took this many tries:**

22: rcl 0 1 display the attempt counter

Supplied macros

Any interest rate that is required by a macro should be in the form of a whole number with a possible fractional value, i.e. 10.2 percent is entered as 10.2.

Future value of an investment

This macro calculates a future value of an investment when interest is a factor.   You must provide the amount of the initial investment, the nominal interest rate, the number of compounding periods per year and the number of years of investment.

Future value of regular deposits

This macro calculates a future value when deposits are made regularly.   All deposits are equal.   You must provide the amount of each deposit, the number of deposits per year, the number of years, and the nominal interest rate.

Initial investment

This macro calculates the investment necessary to provide a stated future value in a specified time period.   You must enter the future value of the investment, the number of years of investment, the number of compounding periods per year and the nominal interest rate.

Investment for withdrawals

This macro calculates the minimum investment required to allow regular withdrawls over a specified time period.   The amount calculated is dependent upon the amount of each withdrawal, the number of withdrawals per year, the number of years, and the nominal interest rate on the investment.   All withdrawals are equal.

Only the least amount necessary for your investment is calculated; the macro assumes a balance of $0.00 to be left at the end of the time period.   Any investment larger than the amount calculated will also enable you to withdraw the desired amount, but leave a remaining balance.

Linear regression

This macro fits a straight line to a given set of coordinates using the method of least squares.   The coefficient of determination, coefficient of correlation and standard error of estimate are displayed.   Once the line has been fitted, you may predict values of y for given values of x.   Press ESC to quit the macro.

Regular payment on a loan

This macro calculates the amount required as a regular payment in order to repay a loan over a specified time period.

Regular withdrawals

This macro calculates the maximum amount which may be withdrawn regularly from an investment over a specified time period.   All withdrawals are assumed to be equal.   You must provide the amount of the initial investment, the nominal interest rate, the number of withdrawals per year and the number of years.

Required regular deposits

This macro calculates the amount required as a regular deposit to provide a stated future value in a specified time period.   All deposits are equal.   It is necessary for you to supply the future value, the nominal interest rate, the number of deposits per year and the number of years.

Functions

^	raise to the power of
1/x	reciprocal
°'"	degrees, minutes, seconds
±	sign change
<<	shift bits left
>>	shift bits right
AC	clear current computation
and	binary and
bep	beep
C	clear the display
cnv	conversion
cos	cosine
EE	enter exponent
fix	fix number of decimal points
inp	wait for user input
inv	inverse next function
k	konstant
lnx	natural logarithm
log	base 10 logarithm
m+	add value to memory register
m-	subtract value from memory
m<>	

	swap display and memory
mc	clear memory
mr	recall memory
msg	display a message to the user
not	binary not
or	binary or
pi	3.1415926
rcl	recall value from memory
rnd	random number
sin	sine
sqr	square root
sto	store value in memory
tan	tangent
tst	test a value
xor	binary exclusive or
x^2	square

^

raise to the power of

This function raises the value in the display to a given power.

Example: raise 5 to the power of 3.4

Operation	Display
5	5.
^	5.0000
3 . 4	3.4
=	237.9567

Using the inv key before raising the power computes the root of a value.

Example: compute the cube root of 7, (7^1/3)

Operation	Display
7	7.
inv ^	7.
3	3
=	1.9129

These examples assume that the number of decimal places is fixed to 4.

1/x

reciprocal

This function divides 1 by the value in the display.

Example: compute the reciprocal of 12.

Operation	Display
1 2	12.
1/x	0.0833

This example assumes that the number of decimal places is fixed to 4.

°'"

degrees, minutes, seconds

This function allows you to enter a number expressed in sexagesimal notation.

Example: enter   5 degrees, 6 minutes, and 23 seconds.

Operation	Display
5	5.
°'"	5.0000
6	6.
°'"	5.1000
2 3	23.
°'"	5.1064

5.1064 is equivalent to 5 degrees, 6 minutes, and 23 seconds.

This function also displays values in sexagesimal notation.

Example: display 5.1064 in sexagesimal notation

Operation	Display
5 . 1 0 6 4	5.1064
inv	5.1064
°'"	5°6' 23"

These examples assume that the number of decimal places is fixed to 4.

±

sign change

This function reverses the sign of the value in the display. It works on either the mantissa or exponent as they are entered.

Example: change the sign of the mantissa being entered.

Operation	Display
1 2	12.
±	-12.0000

Example: change the sign of the exponent being entered.

Operation	Display
1 2	12.
EE	-12. 00
5	12. 05
±	12. -05

<<

shift bits left

This function shifts the bits representing the displayed value to the left. The function is only available in Binary, Octal, Decimal, or Hexadecimal modes. Each shift to the left is equivalent to multiplication by 2.

Example: shift the bits left 3 times in the value 324.

Operation	Display
View Decimal	
3 2 4	324
<<	648
<<	1296
<<	2592

>>

shift bits right

This function shifts the bits representing the displayed value to the right. The function is only available in Binary, Octal, Decimal, or Hexadecimal modes. Each shift to the right is equivalent to division by 2.

Example: shift the bits right 3 time in the value 2592.

Operation	Display
View	
Decimal	
2 5 9 2	2592
>>	1296
>>	648
>>	324

AC

clear current computation

This function clears the numeric display, the message display, and the computation stack, i.e. parentheses, exponent, and functions still in progress.

and

binary and

This function performs a binary **and** operation on two operands. This function is only available in Binary, Octal, Decimal, or Hexadecimal modes. A binary **and** is true only if both bits are true.

Example: **and** the binary values 1010 and 0110.

Operation	Display
View	
Binary	
1 0 1 0	1010
and	1010
1 0 0 1	1001
=	1000

To perform a **nand** (not **and**) operation press the not after pressing the equals sign.

Example: perform a **nand** on the binary values of 1010 and 1001. This operation uses leading zeros that are not displayed. In the example below, although it is a 16 bit operaton, only 4 bits are displayed. Twelve bits of leading zeros are not displayed.

Operation	Display
View	
Binary	
1 0 1 0	1010
and	1010
1 0 0 1	1001
= not	1111111111110111

bep

beep

This function produces a beep through the computer's speaker.   Within macros, it is a useful prompt for user input.

C

clear the display

This function clears the displayed value.   It does not affect the computation stack, i.e. parentheses, exponent, and functions still in progress.

cnv
conversion

This function transfers the currently selected conversion to the display. Use the Conversions command in the Extras menu to select a conversion formula.

Example: convert 5 miles to kilometers

Operation	Display
Extras	
Conversions	
select miles st. -> kilometers ok	
5	5.
*	5.
cnv	1.6090
=	8.0450

inv cnv <- conversion

To reverse the conversion use the inv button.

Example: convert 8.0450 kilometers to miles

Operation	Display
Extras	
Conversions	
select miles st. -> kilometers ok	
8 . 0 4 5 0	8.0450
*	8.0450
inv cnv	0.6215
=	5.0000

cos

cosine

This function produces the cosine of the displayed value.

Example: calculate the cosine of 67 degrees.

Operation	Display
6 7	67.
cos	0.3907

inv cos arc cosine

To perform an arc cosine, precede the cosine function with the inv key.

Example: calculate the arc cosine of 0.3907.

Operation	Display
. 3 9 0 7	0.3907
inv cos	67.0019

EE

enter exponent

This function allows you to enter a value in exponential format. This is necessary when the value exceeds the limits of the display. For example, convert the value 237650000000000000000.0 to 2.3765x10^20 for entry, this is read as 2.3765 times 10 to the 20th. The exponent of 20 indicates the number of places the decimal point has been moved. If the exponent is positive the the decimal point has been moved to the left, if it is negative the decimal point has been moved to the left.

Example: enter the value 237650000000000000000 in exponential format.

Operation	Display
2 . 3 7 6 5	2.3765
EE	2.3765 ^{00}
2 0	2.3765 ^{20}

fix

fix number of decimal points

This function allows you to select the number of decimal places that appear in the display after a calculation. The precision of the value is maintained internally even though the display is truncated.

Example: enter the number 3.14159 and fix the number of decimal places to 2, then fix the number of decimal places to five.

Operation	Display
3 . 1 4 1 5 9	3.14159
fix	3.14159
2	3.14
fix	3.14
5	3.14159

inp

wait for user input

This function is available when recording a macro. Place this function in a macro at the location where user input is required. When a macro is playing it suspends execution at the points where the inp button was pressed during recording. Once the user has entered the requested information and pressed the = button or enter key, the macro continues.

Example: The example for this function is included in the macros section of the help.

inv

inverse the next function

This function causes the next function entered to perform an inverse of its operation..

Examples: these functions have inverse capability.

inv cnv
<- conversion

inv cos
arc cosine

inv sin
arc sine

inv tan
arc tangent

inv lnx
e^x

inv log
10^x

k

constant

This function transfers the currently selected constant to the display. Select Constants with the Konstants command in the Extras menu.

Example: display the speed of light

Operation	Display
Extras Konstants	
select speed of light ok	
k	186281.34

lnx

natural logarithm

This function computes the natural (Napierian) logarithm of a number.

The natural logarithm of a number is defined as the power to which the base **e** must be raised to produce the number.   The constant **e** has the approximate value of 2.7182818285.

Example: calculate the natural logarithm of 2384.

Operation	Display
2 3 8 4	2384
lnx	7.7765

inv lnx inverse of natural logarithm

The inverse function for natural logarithms raises the constant e to the given power.

Example: calculate **e** to the power of 7.7765.

Operation	Display
7 . 7 7 6 5	7.7765
inv lnx	2383.9165

note: this returns the value used in example one.

[log]

base 10 logarithm

This function computes the base 10 (Briggs) logarithm of a number.

The base 10 logarithm of a number is defined as the power to which the base **10** must be raised to produce the number.

Example: calculate the base 10 logarithm of 2384.

Operation	Display
[2] [3] [8] [4]	2384
[log]	3.3773

[inv] [log] inverse of base 10 logarithm

The inverse function for base 10 logarithms raises the constant 10 to the given power.

Example: calculate 10 to the power of 3.3773.

Operation	Display
[3] [.] [3] [7] [7] [3]	3.3773
[inv] [log]	2383.9656

note: this returns the value used in example one.

m+

add value to memory

This function adds the value in the display to a memory accumulator. Memory can be used to store temporary values. A lower case **m** appears in the display when the memory accumulator has a stored value.

Example: Add the values 512 and 284 to the memory accumulator.

Operation		Display
AC		0.0000
mc		0.0000
5 1 2		512.
m+	m	512.
2 8 4	m	284.
m+	m	284.
C	m	0.0000
mr	m	796.0000

m-

subtract value from memory

This function subtracts the value in the display from the memory accumulator. A lower case **m** appears in the display when the memory accumulator has a stored value.

Example: subtract the values 512 and 284 from memory.

Operation		Display
AC		0.0000
mc		0.0000
5 1 2		512.
m-	m	512.
2 8 4	m	284.
m-	m	284.
C	m	0.0000
mr	m	-796.0000

m<>

swap display and memory

This function swaps the value in the display with the value stored in the memory accumulator.

Example: place the value 24 in memory, 36 in the display, and swap them.

Operation	Display
mc	0.0000
2 4	24.
m+	24.
3 6	36.
m<>	24.0000
m<>	36.0000

mc

clear memory

This function clears the value stored in the memory accumulator.

mr

recall memory

This function copies the value stored in the memory accumulator to the display.

msg

display a message

This function displays a message during macro playback.   The message appears just above the numerical display area.   Any characters can be used to create the message.

After running a macro, a message may remain in the display. Press msg to clear the message area.

Example: The example for this function is included in the macros section of the help.

not

binary not

This function reverses the result of a logic operation at a bit level, effectively flipping the bits in the displayed value. The function is only available in Binary, Octal, Decimal, or Hexadecimal modes.

Example: Perform a **not** on the binary value 100101.

Operation	Display
View Binary	
1 0 0 1 0 1	100101
not	11010
not	100101

or

binary or

This function performs a binary **or** using two operands. The function is only available in Binary, Octal, Decimal, or Hexadecimal modes.

Example: perform an **or** on the binary values 1010 and 0110.

Operation	Display
Modes	
Binary	
1 0 1 0	1010
or	1010
1 0 0 1	1001
=	1011

To perform a **nor** (not **or**) operation press the not after pressing the equal sign.

Example: perform a **nor** on the binary values 1010 and 1001. This operation uses leading zeros that are not displayed. In the example below, although it is a 16 bit operaton, only 4 bits are displayed. Twelve bits of leading zeros are not displayed.

Operation	Display
View	
Binary	
1 0 1 0	1010
or	1010
1 0 0 1	1001
= not	1111111111110100

pi

3.1415926

This function copies the constant **pi** to the display.

Example: What is the circumference of a circle with a radius of 2.5.

Operation	Display
2 . 5	2.5
*	2.5
2	2.
*	5.0000
pi	3.1416
=	15.7080

rcl

recall value from memory

This function copies the contents of one of the 100 memory registers to the display. Memory registers are numbered 00 through 99.

Example: Store 127 in register 23 and perform an operation on it.

Operation	Display
1 2 7	127.
sto 2 3	127.
5	5.
*	5.
rcl 2 3	127.0000
=	635.0000

rnd

random number

This function produces a random number between 0.0 and 1.0 inclusive.

Example: Produce a random number between 1 and 100.

Operation	Display
rnd	0.6823
*	0.6823
9 9	99.
+	67.5477
1	1.
=	68.5477

sin

sine

This function produces the sine of the displayed value.

Example: Calculate the sine of 67 degrees.

Operation	Display
6 7	67.0
sin	0.9205

inv sin arc sine

To perform an arc sine, precede the sine function with the inv key.

Example: calculate the arc sine of 0.9205.

Operation	Display
. 9 2 0 5	0.9205
inv sin	66.9992

sqr

square root

This function calculates the square root of a number.   The square root of a number is a number which when multiplied with itself produces the original number. Squareroot operates only on positive values.

Example: calculate the square root of 324.

3 2 4	324.0
sqr	18.0000

sto

store value in register

This function copies the contents of the display to one of the 100 memory registers. Memory registers are numbered 00 through 99.

Example: Store 127 in register 23 and perform an operation on it by recalling the value..

Operation	Display
1 2 7	127.
sto 2 3	127.
5	5.
*	5.
rcl 2 3	127.0000
=	635.0000

tan

tangent

This function produces the tangent of the displayed value.

Example: Calculate the tangent of 67 degrees.

Operation	Display
6 7	67.0
tan	2.3558

inv tan arc tangent

To perform an arc tangent function precede the tangent function with the inv key.

Example: calculate the arc tangent of 2.3558.

Operation	Display
2 . 3 5 5 8	2.3558
inv tan	66.9995

tst

test a value

This function is available only when recording a macro.   Use it to place conditional jumps in a macro.

The **build test** dialog box allows you to create a conditional of the form:

if X conditional RCLmm GOTOnnn

where x is the value in the display

conditional is one of the following:

< less than
<= less than or equal to
> greater than
>= greater than or equal to
= equal to
<> not equal to

mm is the number of a memory register

nnn is the step number to go to.

Example: The example for this function is included in the macros section of the help.

xor

binary exclusive or

This function performs a binary exclusive **or** using two operands. The function is only available in Binary, Octal, Decimal, or Hexadecimal modes.

Example: perform an **exclusive or** on the values of 1010 and 1001. This operation uses leading zeros that are not displayed. In the example below, although it is a 16 bit operaton, only 4 bits are displayed. Twelve bits of leading zeros are not displayed.

Operation	Display
View	
Binary	
1 0 1 0	1010
xor	1010
1 0 0 1	1001
=	11

To perform an **xnor** (exclusive **not or**) operation press the not after pressing the equals sign.

Example: perform a binary **xnor** on the binary values of 1010 and 1001.

Operation	Display
View	
Binary	
1 0 1 0	1010
xor	1010
1 0 0 1	1001
= not	1111111111111100

x^2

square

This function squares the value in the display. Squaring a value is the same as multiplying it by itself.

Example: square the value 34.2.

Operation	Display
3 4 . 2	34.2
x^2	1169.6400

Modes