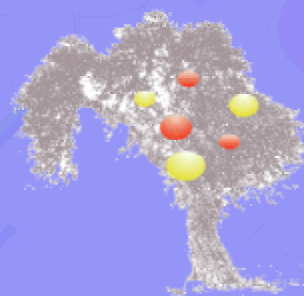


**Portrait**   
*Graphing & Calculating*

*User's Guide*



MathResources Inc.  
5516 Spring Garden Road  
Suite 312  
Halifax, Nova Scotia  
Canada  
B3J 1G6

1-902-429-1323 (Phone)  
1-800-720-1323 (Toll Free in North America)  
1-902-492-7101 (Fax)  
[support@mathresources.com](mailto:support@mathresources.com) (Email)

<http://www.mathresources.com>

***Portrait 4 for Pocket PC - User's Guide***

© Copyright 2005. MathResources Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of the copyright holder.

MathResources Inc. acknowledges all trademarks and registered trademarks in this document.

Printed in Canada

ISBN 1-896977-20-0

06-06-2005

# TABLE OF CONTENTS

The Screen / Interface .....	3
BUTTONS ON THE CALCULATOR .....	3
SYMBOLIC CONSTANTS .....	4
Entering Data .....	4
DATA TYPES .....	4
CONVERTING BETWEEN DATA TYPES (CASTING) .....	5
BOOLEAN OPERATORS .....	5
RELATIONS .....	6
ORDER OF OPERATIONS .....	6
FUNCTIONS MENU .....	7
HISTORY LIST .....	8
ERROR MESSAGES .....	8
Options Menu .....	9
FUNCTION TEMPLATES .....	9
HELP MODE .....	10
ACCESSING HELP FROM START .....	10
SHOW CALCULATION LOG .....	11
FIX - NUMBER OF DECIMAL PLACES .....	12
MODE – NORMAL OR SCIENTIFIC/DECIMALS OR FRACTIONS/DEGREES OR RADIANS .....	12
Editing Menu & Tools .....	13
LIST EDITOR .....	13
MATRIX EDITOR .....	14
SEQUENCE EDITOR .....	15
TABLE EDITOR .....	16
UNIT CONVERTER .....	16
VARIABLE EDITOR .....	17
GENERATING A GRAPH FROM THE LIST, MATRIX, SEQUENCE OR TABLE EDITOR .....	18
Calculus .....	18
GRAPHING CALCULUS FUNCTIONS .....	19
Complex Numbers .....	20
Consumer Applications .....	21
Date .....	22
Equations .....	23
GRAPHING A SINGLE EQUATION .....	23
GRAPHING A SYSTEM OF LINEAR EQUATIONS .....	25
Exponential and Logarithmic Functions .....	25
Finance: Amortization .....	26
Finance: Depreciation Template .....	27
Finance: Time Value of Money (TVM Template) .....	28
Finance: Cash Flow Analysis .....	30
Finance: Interest, Bonds and Net Present/Future Values .....	31
Hyperbolic Functions .....	33
INVERSE HYPERBOLIC FUNCTIONS .....	34
Linear Programming .....	35
GRAPHING LINEAR INEQUALITIES .....	36
Lists .....	36
Lists .....	37
Matrices .....	38
Number Theory Functions .....	42
Real Numbers .....	45
Sequences .....	46
Statistical Functions .....	47
CONFIDENCE INTERVALS .....	47
PROBABILITY DISTRIBUTIONS .....	49
GRAPHING PROBABILITY DISTRIBUTIONS .....	52
STATISTICAL TESTS .....	53
ADDITIONAL STATISTICAL FUNCTIONS .....	55

String.....	57
Test.....	58
Trigonometric Functions.....	58
INVERSE TRIGONOMETRIC FUNCTIONS.....	59
Plots Menu and Graphing.....	61
FILE MENU.....	62
FILE MENU.....	62
EDIT MENU.....	62
TOOLS MENU.....	65
CHANGING COLORS, POINTS AND OTHER PROPERTIES.....	66
OTHER GRAPHING ICONS.....	68
Bar Graphs.....	69
Box and Whisker Plots.....	70
Curves of Best Fit.....	71
Histograms.....	73
Pie Graphs.....	74
Normal Probability Plot.....	75
Graphs of Functions.....	76
Polar Plots.....	77
Parametric Plots.....	78
Graphs of Equations.....	79
Point Plots.....	80
Graphs of Inequalities.....	81
Trouble Shooting.....	82
Contacting MathResources Inc.....	82
Copyrights and Trademarks.....	82

## Using *Portrait 4*

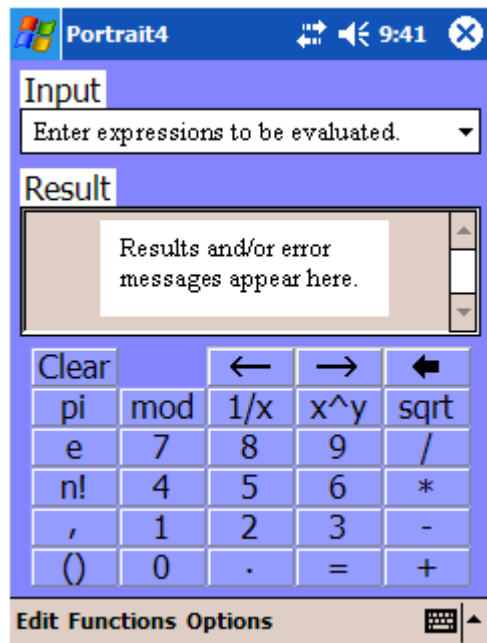
*Portrait 4* is a powerful piece of graphing and calculating software that is also very easy to use. This *User's Guide* is written in an easy-to-understand style and covers basic functionality of the software. **Help** is also available from the main menu of your handheld computer. The software is designed to have you up and running very quickly.

### The Screen / Interface

The basic operating window of *Portrait 4* is very simple. When the program is open there is a calculator screen with three menus at the bottom. From these menus (the **Edit** menu, the **Functions** menu and the **Options** menu) you can access all of the graphing and calculating power of *Portrait 4*. To open a menu, tap on the menu name and then tap on a specific command to open it. Some menus have submenus to choose from.

The screen also contains number and function buttons that are standard on most handheld calculators. To use any of these buttons simply tap on them.

The **Plots** menu is a part of the **Functions** menu. The trigonometric, hyperbolic, exponential and log functions are also found in the **Functions** menu.




### Buttons on the Calculator

The buttons on *Portrait 4* provide standard calculator operations.

- Clear** Clear -- Tapping on Clear removes all information in the Input and Result windows.
- ←** Left Arrow -- Tapping on the left arrow moves the cursor to the left.
- Right Arrow -- Tapping on the right arrow moves the cursor to the right.
- ↶** Backspace -- Tapping on the backspace arrow moves the cursor to the left, deleting input.
- mod** mod --  $m \bmod n$  finds  $m$  modulo  $n$ , that is, the remainder when  $m$  is divided by  $n$ .  
**Example:**  $5 \bmod 2 = 1$ .
- 1/x** Reciprocal ( $1/x$ ) -- read one over  $x$ . **Example:**  $1/3 = 0.3333\dots$
- x^y** Exponentiation (^) -- A caret (^) **must** be used to indicate an exponent.  
**Example:**  $2^5 = 2*2*2*2*2 = 32$ .
- sqrt** Square root -- Finds the square root of a non-negative number. **Example:**  $\text{sqrt}(16) = 4$ .  
Other roots can be calculated using the following format:  $x^{(1/n)}$ . **Example:**  $27^{(1/3)} = 3$ .
- n!** Factorial (n!) -- **Example:**  $5! = 5*4*3*2*1 = 120$ .

## Symbolic Constants

There are two symbolic constants on the calculator. They are:

 pi -- The constant  $\pi$  is denoted by pi. It is approximately equal to 3.14159265359.

 e -- The constant e. It is approximately equal to 2.71828182846.

## Entering Data

There are some basic rules to keep in mind when entering data in *Portrait 4*. These are covered in the following sections on Data Types, Converting between Data Types, Boolean Operators, Relations and Order of Operations.

### Data Types

All values in the calculator must be one of the following types. Any value or expression that exceeds the range of its type generates an error message.

- **Integers** -- numbers like 1, 0, -5, etc. There is a limit to the size of integer you can use in an expression. The largest or smallest integer you can use is +/- 2,147,483,647 or  $2^{31}-1$ . Entering an integer that is too large results in *error: integer overflow*. If the integer you input is too large you may want to put it in decimal form. For example,  $2^{32}$  gives an error message but  $2.0^{32}$  does not.
- **Fractions** -- The numerators and denominators of fractions are stored as integers. Any fraction whose numerator or denominator exceeds the integer range generates an integer overflow error.
- **Decimal numbers** -- numbers with a decimal place like 1.5, 3.0, and -2.1. Scientific notation is also allowed. For example,  $1.1e90$  and  $3.2e-5$ . There is a limit to the size of decimal number you can use in an expression. The largest or smallest decimal allowed is approximately +/-  $1.79e308$ . Any decimal number that is too large results in *error: floating-point overflow*.
- **Boolean** -- true/false values. A Boolean variable, function, or operator takes either of the values *true* or *false*. To manipulate Boolean values in expressions, you must use **Boolean operators**.
- **Lists** -- list of expressions. Any function that takes a single argument can also handle a list of arguments. For example, {1, 2, 3, 4, 5} or {1, 2, {1, 2, 3, 4}}. Lists must be enclosed within **braces** { } with each element separated by commas. Each list element can be a value of any type, but most functions that use lists expect numeric elements only. For example,  $\sin(\{1, 2, 3\}) = \{\sin(1), \sin(2), \sin(3)\} = \{0.84, 0.91, 0.14\}$ . Similarly, arithmetic operations with lists and numbers are also possible. For example,  $\{1, 2, 3, 4\} * 5 = \{5, 10, 15, 20\}$  and  $\{1, 2, 3\}^2 = \{1, 4, 9\}$ . An empty list cannot be entered but can be generated by certain functions.
- **Matrices** -- rectangular arrays of numbers that are arranged in rows and columns. Matrices can be inserted into other expressions using the **matrix( )** function. Only two-dimensional matrices are supported, and only numeric types are allowed in matrices.
- **Complex Numbers** -- Arithmetic operations and operations using real numbers are supported with complex numbers. Complex numbers,  $z$ , are numbers in the form  $a + b*I$  where  $I^2 = -1$ . For example,  $(-2.5)^{(3.5)} = 0 - 24.705*I$ . **Capital I is used for complex numbers.**
- **Dates** -- Dates can be entered as a type of data. The following function can be used to create a date: **makeDate(day, month, year, hour, minute, second)**. For example, **makeDate( 2, 4, 2000, 11, 16, 15)** returns the time and date 11:16:15 am, Sunday, April 2, 2000.
- **Strings** -- A string is a finite sequence of characters, like numerals, letters and symbols. The length of a string is determined by its number of characters. The length of a string can be zero or any positive integer. Strings are always contained in quotation marks, like the string "abcdef". Substrings are any adjacent sequence of characters in a string.

## Converting between Data Types (Casting)

To convert a value to a different data type, insert the desired type, in parentheses, before the value to be converted. Casting takes effect before any other operation.

### Converting to an Integer

To convert either a Boolean or a decimal number to an integer: **(integer)value** – converts the true or false value to an integer. The conversion rules are:

- The Boolean value **true** is converted to 1. **Example:** (integer>true = 1.
- The Boolean value **false** is converted to 0. **Example:** (integer>false = 0.
- Decimal number values are truncated to the largest integer less than the decimal number.  
**Example:** (integer)5.75 = 5.

### Converting to a Decimal

To convert either a Boolean or integer type to a decimal number: **(double)value** – converts the true or false value to a decimal number. The conversion rules are:

- The Boolean value **true** is converted to 1.0. **Example:** (double>true = 1.0. This is displayed on the calculator as 1 but, it is stored as 1.0 in the calculator.
- The Boolean value **false** is converted to 0.0. **Example:** (double>false = 0.0. This is displayed on the calculator as 0 but, it is stored as 0.0 in the calculator.
- Integers are converted to the same value but are stored internally as decimal numbers. This is convenient when using integers in an expression that might overflow the range of integer values.  
**Example:** (double)5 = 5.0.  
**Example:** 5\*1000\*1000\*1000 overflows the integer range but (double)5\*1000\*1000\*1000 does not.

### Converting to Boolean

To convert either an integer or decimal number to a Boolean value: **(boolean)value** converts any value to a Boolean. If the value is non-zero, the result is true and otherwise the result is false.

**Example 1:** (boolean)0 = false

**Example 2:** (boolean)5 = true.

Converting to other data types (fractions, lists, matrices) is not supported by this method.

## Boolean Operators

- **&** or **and** --  $p \& q$  -- tests if both  $p$  and  $q$  are true.  
**Example:** (9>5)&(4<3) tests if both of these expressions are true. One of them is not, so the result is false.
- **|** or **or**--  $p|q$ -- tests if  $p$  or  $q$  is true.  
**Example:** (9>5)|(4<3) tests if one of these expressions is true. One of them is true, so the result is true.
- **~** or **not** --  $\sim p$  -- tests if  $p$  is not true.  
**Example:**  $\sim(9>5)$  tests if the expression is not true. The expression is true, so the result is false.
- **xor** --  $p \text{ xor } q$  tests if  $p$  or  $q$  is true, but not both. The result is true if exactly one of the two is true and false if both are true or if both are false.  
**Example:** (9>5) xor (9>6) tests if exactly one of these expressions is true. They are both true, so the result is false.

## Relations

Relational comparisons between numeric expressions can be evaluated. They return a Boolean value.

- `<` -- less than.  $x < y$  tests whether  $x$  is less than  $y$ . It is true if  $x$  is less than  $y$  and false if  $x$  is greater than or equal to  $y$ .  
**Example:**  $9 < 4 = \text{false}$ .
- `>` -- greater than.  $x > y$  tests whether  $x$  is greater than  $y$ . It is true if  $x$  is greater than  $y$  and false if  $x$  is less than or equal to  $y$ .  
**Example:**  $9 > 4 = \text{true}$ .
- `<=` -- less than or equal to.  $x <= y$  tests whether  $x$  is less than or equal to  $y$ . It is true if  $x$  is less than or equal to  $y$  and false if  $x$  is greater than  $y$ .  
**Example:**  $12 <= 16 = \text{true}$ .
- `>=` -- greater than or equal to.  $x >= y$  tests whether  $x$  is greater than or equal to  $y$ . It is true if  $x$  is greater than or equal to  $y$  and false if  $x$  is less than  $y$ .  
**Example:**  $12 >= 16 = \text{false}$ .
- `=` -- is equal to.  $x = y$  tests whether  $x$  is equal to  $y$ . It is true if  $x$  is equal to  $y$  otherwise it is false.  
**Example:**  $3 + 4 = 7 = \text{true}$ .
- `<>` -- not equal to.  $x <> y$  tests whether  $x$  is not equal to  $y$ . It is true if  $x$  is not equal to  $y$  and false if  $x$  is equal to  $y$ .  
**Example:**  $3 + 4 <> 7 = \text{false}$ .

## Order of Operations

When evaluating expressions, the calculator does so in a particular order. The order from highest precedence to lowest precedence is as follows:

Casting -- ((integer), (boolean), (double))

Factorial -- (!)

Exponentiation -- (^)

Multiplicative -- (\*, /, mod)

Additive -- (+, -)

Relational -- (<, >, <=, >=, =, <>)

Boolean -- (&, |, ~, xor)

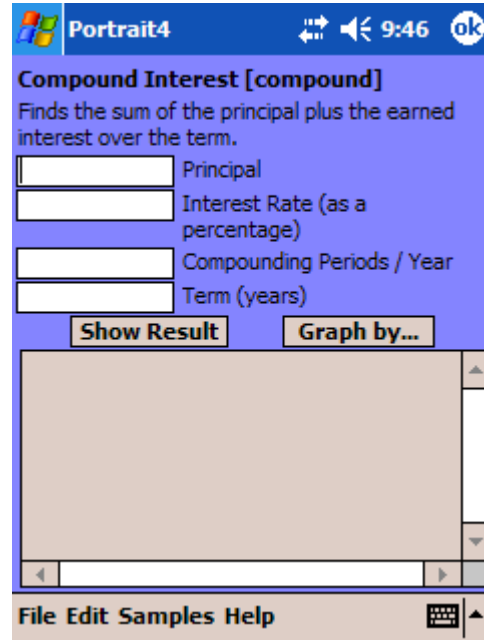
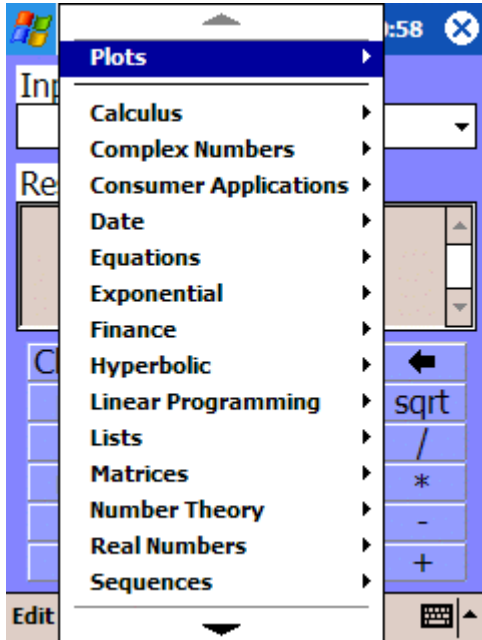
If you are uncertain about the order, simply use brackets.



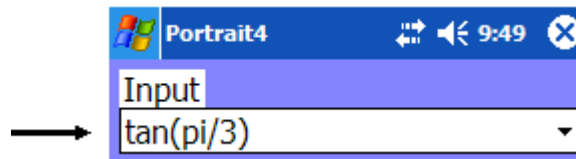
## Functions Menu

Portrait 4 has over 200 functions. The three ways to access the functions are:

1. Use the **Functions** menu on the calculator. The functions listed in the menu are organized into categories. Each of these categories expands to display the available functions. Tap on a function to open a template. Enter data in the appropriate fields.



2. Tap on a **button**. Some of the functions have a button assigned to them on the calculator. Examples of these are  $x^y$ ,  $\sqrt{x}$  and  $1/x$ . When you select one of these functions, it appears in the **Input** window. Enter your data inside the brackets in the input window.
3. **Enter** the function you want directly in the **Input** window as a string of data. Any command, including those assigned to a button and those listed in the **Functions** menu, can be entered as a string of data. Tap in the **Input** window. Enter the expression you want to evaluate as in the example below.



- An asterisk (\*) **must** be used to indicate multiplication. Implicit multiplication by placing two factors side-by-side is not supported.
- A caret (^) **must** be used to indicate an exponent.

### Tips

- Expressions can be combined in the input window. If you are entering more than one expression, semicolons must separate the expressions. For example, input

$$x:=2*y; y:=3; x.$$

Here, a value is assigned to  $x$  and to  $y$  and then the calculator solves for  $x$ ; the solution is 6. You can combine any number of expressions but only the result of the last expression is displayed in the **Result** window.

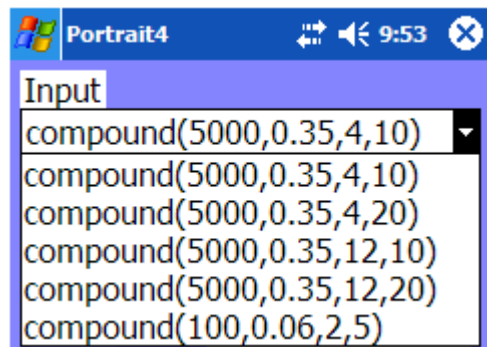
- To wrap a function around information already entered in the input window, highlight the text and then choose the function you want to enclose it in. This inserts the function in the correct position, with its brackets appropriately placed. This can be a helpful, time saving feature.

If you are inputting strings of data in the **Input** window, answers or error messages are displayed in the **Result** window. Once you input your data, tap on the equal sign (=) button on the calculator, or press **Enter** on the keyboard to view a solution. If the solution is longer than the result window, a scroll bar will appear so the complete solution can be viewed.

To remove a result, simply tap on the **Clear** button. This clears the **Input** window as well.

## History List

At the far right of the **Input** window is a down arrow. If you are inputting strings of data in the **Input** window tap on the arrow to view a list of the most recent calculations. It stores up to 25 previous calculations (you can use the scroll bar to view them all). Tap on any entry in the list to copy it into the **Input** window. To remove all the calculations in the history list, go to the **Edit** menu and select **Clear History**.



## Error Messages

Error messages appear in the result window if the expression you input cannot be evaluated. There are many potential error messages and they are self-explanatory. For example:

- **error: syntax error** - You made a typing mistake or input a nonsensical expression. The expression is then highlighted approximately where the error has been made.
- **error: division by zero** - The expression you want to evaluate includes division by zero.
- **error: input is too large** - You entered an integer or decimal number that is too large. See the section on Data Types for the limits on the size of integers and decimal numbers that can be input.
- **error: wrong number of arguments** - You have not entered the proper number of arguments to evaluate the expression.
- **error: unassigned variable** - You have not given a value to a variable you have used.
- **error: floating point overflow, error: interger overflow** - An intermediate or final result is too large for the range of its data type (see the section on DataTypes).
- **error: fsolve requires at least 4 arguments**

To remove an error message, simply tap the **Clear** button. This clears the **Input** window as well.

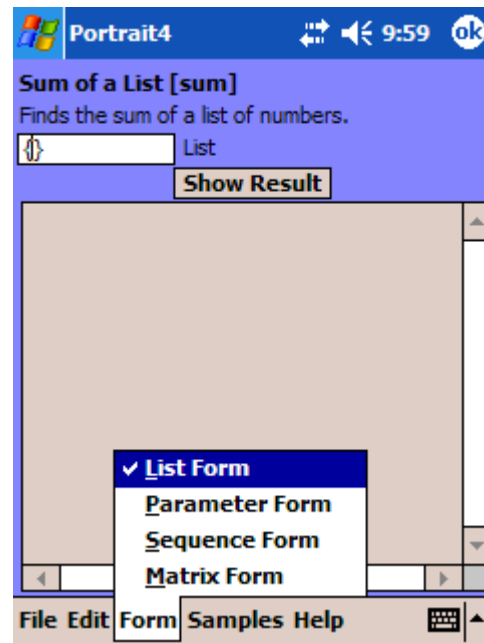
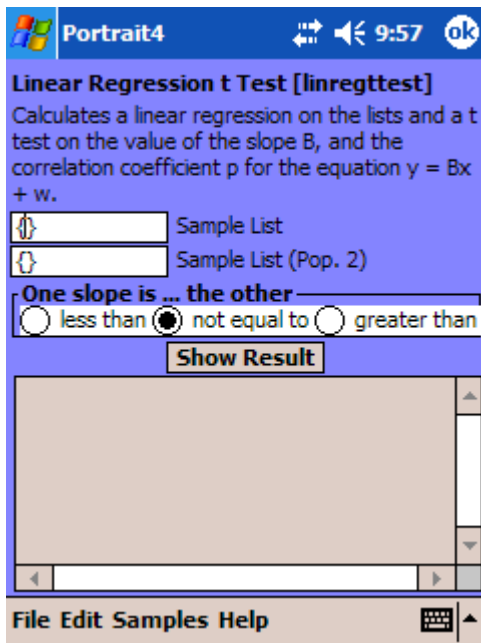
# Options Menu

## Function Templates

The simplest, most efficient way to do calculations with *Portrait 4* is by using the **Function Templates**, especially if you are uncertain about entering strings of data in the **Input** window. When a **Function** is selected from the menu, a template opens for that function. The template indicates the order and nature of the function's arguments that must be entered to obtain a result. Results are obtained by tapping on the **Show Result** button. **Function Templates** "on" is the default setting as indicated by the check mark. If you want to turn this option off, go to the **Options** menu and deselect **Functions Templates**. If you now select **Options** a second time, **Function Templates** no longer has a check mark beside it.

Most function arguments have an entry field followed by a brief description. In most entry fields, you can enter any expression involving values, operators and functions. Some entry fields require a variable or a list; the description indicates if this is the case. If you are required to enter a list, the list must be contained within braces { }. Some arguments are represented by a group of buttons. You can select any of the buttons to indicate your choice for that argument.

The results displayed in the **Show Result** window are followed by the parameters used to obtain that result. If you are experimenting with different values for a specific parameter this will help you compare results.



Some functions have multiple forms, that is, they can take different lists of arguments. These functions have a **FORM** menu. Once you select the form of the function that you want, the template changes to show the list of arguments for that form.

All templates have several useful menus. The **File** menu allows you to **Open** or **Save** the data in a template. The **Edit** menu allows you to perform normal editing operations. The **Samples** menu contains one or more sets of sample data. Selecting an item from the **Samples** menu copies the sample data into the template. Tap on the **Show Result** button to obtain the result of the sample data.

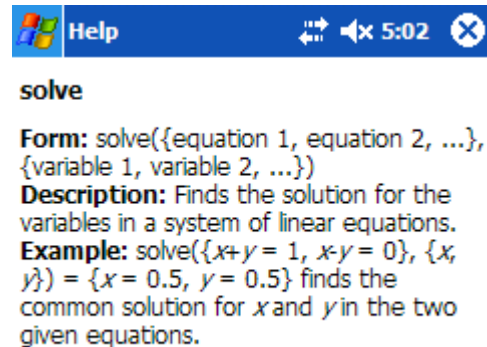
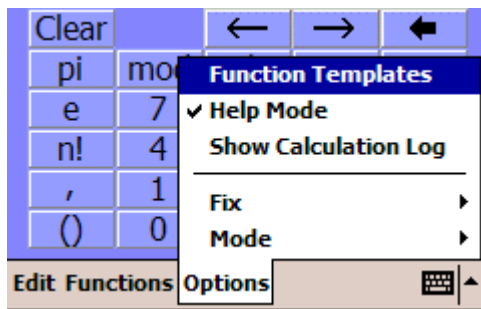
Tap **ok** in the corner of the template to close it.

**It is important to remember when you are entering functions and/or equations in the Function Template:**

- An asterisk (\*) **must** be used to indicate multiplication. Implicit multiplication by placing two factors side-by-side is not supported.
- A caret (^) **must** be used to indicate an exponent.

## Help Mode

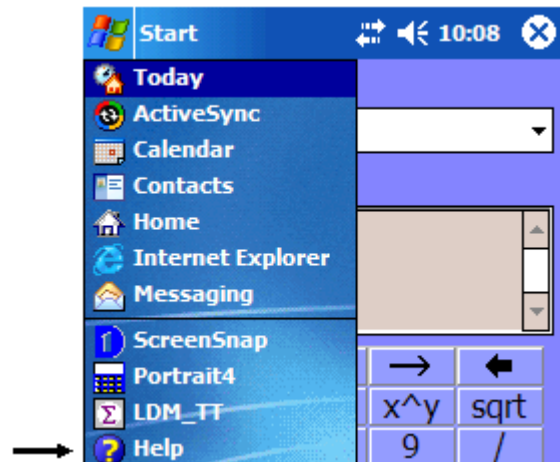
Under the Options menu select Help Mode. If you select Options a second time you see that Help Mode has a check mark beside it. The check mark indicates that the Help Mode has been activated. Now when a Function (with Function Templates deactivated) or Plot is selected from the menu bar the Help file for that command opens. For example, if you open the FUNCTION menu and then select solve from the Equations section of the menu, the help file will open a separate window containing the section on solving a system of linear equations. Each Help window contains information on what is required in the input window and how to input the information for immediate results. A complete example is also included. Each individual function template also has its own Help menu.



As long as Help Mode is selected you are provided with assistance on all computation and plotting functions. **When learning how to use the calculator it may be useful to keep this feature on.**

## Accessing Help from Start

When *Portrait 4* is open on your handheld Help can be accessed by tapping on the Start menu. Tap on Help to see a Table of Contents for Graphing Calculator Help. Each item in the table of contents expands to show all of the items in that list. For example, tap on Plots to see a complete list of all of the plots available in *Portrait 4*. From this list tap on a specific plot to see detailed help for that plot. Each Help window includes information on what is required in the input window, how to input the information for immediate results plus a complete example.





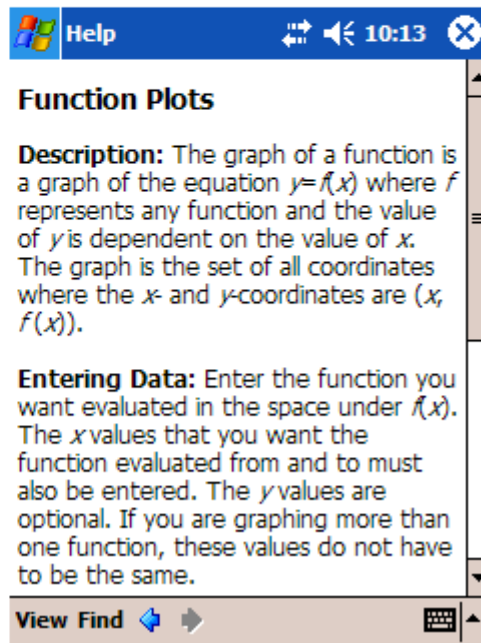
## Graphing Calculator Help

- [Edit Menu](#)
- [Plots](#)
- [Functions](#)
- [Keypad](#)



### Plots

- [Bar Graph](#)
- [Box and Whisker Plot](#)
- [Curves of Best Fit](#)
- [Histogram](#)
- [Pie Graph](#)
- [Normal Probability Plot](#)
- [Plot  \$y=f\(x\)\$](#)
- [Polar Plot  \$r=r\(t\)\$](#)
- [Parametric Plot  \$x=f\(t\), y=g\(t\)\$](#)
- [Plot Equations](#)
- [Point Plot](#)
- [Inequality Plot](#)



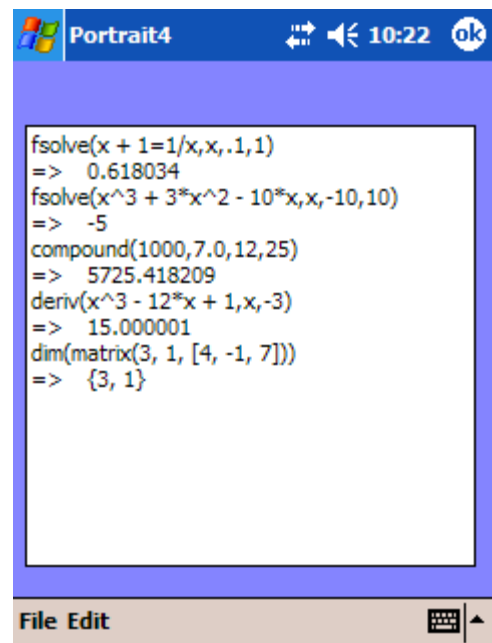
## Show Calculation Log

The Calculation Log provides a record of the calculations you have performed using the function templates.

The data in the Calculation Log can be saved.

It can also be copied and pasted to a new location, like a word file.

The log can be cleared as well.

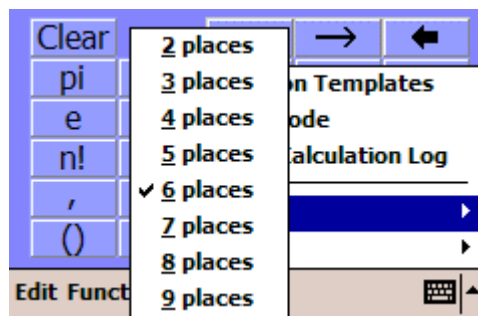


## Fix - Number of Decimal Places

You can select the number of decimal places that you would like results to show. You can have as few as 2 digits after the decimal place and as many as 9.

To change the number of decimal places:

1. Tap on the Options menu.
2. Select Fix to see the numbers of decimal places available.
3. Select the number of decimal places you want the results to have by tapping on your choice. A check mark indicates your selection. The default setting is 6.

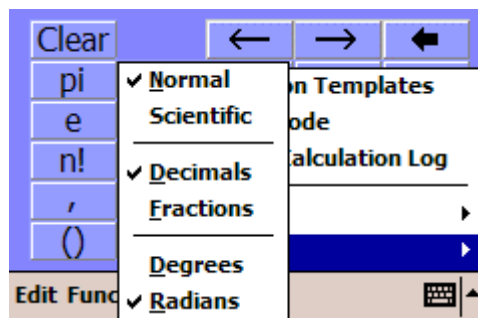


## Mode – Normal or Scientific/Decimals or Fractions/Degrees or Radians

You can choose to see numeric results as Normal or Scientific, and Decimals or Fractions. The Mode option also includes a choice between Degrees and Radians.

To change between normal and scientific mode:

1. Tap on the Options menu.
2. Select Mode to display the choices.
3. Choose either Normal or Scientific from the menu. A check mark indicates your selection. The default setting is Normal.



To change between decimal and fraction form:

1. Tap on the Options menu.
2. Select Mode to display the choices.
3. Choose either Decimals or Fractions from the menu. A check mark indicates your selection. The default setting is Decimals.

To change between degrees and radians:

1. Tap on the Options menu.
2. Select Mode to display the choices.
3. Choose either Degrees or Radians from the menu. A check mark indicates your selection. The default setting is Degrees.

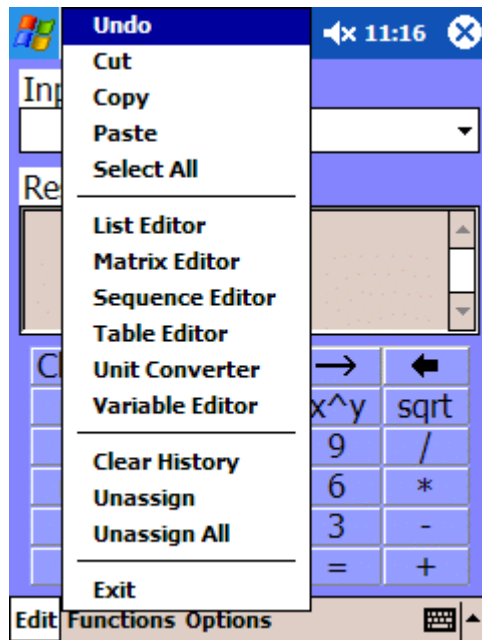
If the input uses fractions and decimals, the result appears in decimal form regardless of the Decimals/Fractions mode. For example, if the input is  $\frac{1}{2} * 5.0$ , the result is 2.5, even if the fraction mode is selected. If the input is  $\frac{1}{2} * 5$ , the result is  $\frac{5}{2}$ , if the fraction mode is selected, otherwise the result will display as 2.5.

### Tip

- Keep in mind that some calculations can be more accurate if done in fraction mode. For example, you get an exact answer calculating determinants of matrices with integer entries in fraction mode. In decimal mode there are always round off errors and you can get misleading answers.

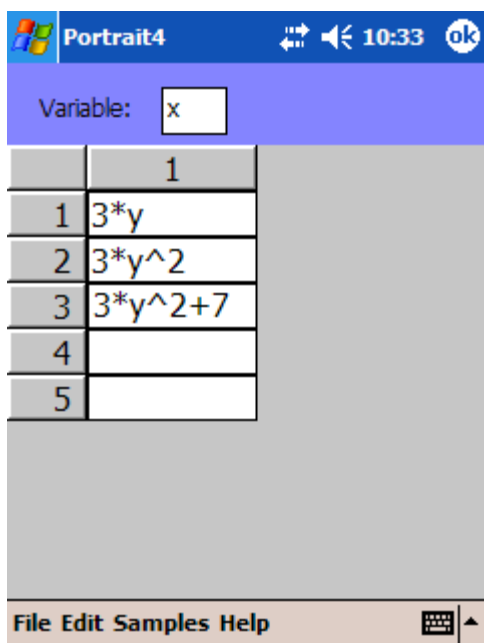
## Editing Menu & Tools

The Edit menu includes the cut, copy, paste, and undo operations. The **Edit** menu also provides access to the **List Editor**, the **Matrix Editor**, the **Sequence Editor**, the **Table Editor**, the **Unit Converter** and the **Variable Editor**. The **Clear History** and **Unassign** commands are also found here.



## List Editor

With the List Editor you can create a list of values or expressions using a spreadsheet like editor. It is also possible to type lists directly into the Input window. You have the option of assigning your list to a variable.



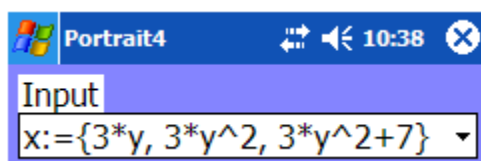
To enter a list:

1. Tap the Edit menu.
2. Open the List Editor.
3. Enter a variable in the space provided. This step is optional. If you do enter a variable name it automatically assigns the list to the variable before the window closes. If there are any errors, a message will appear and the window will not close.
4. Enter the elements of the list in Column 1. To add a row to the list, place the cursor in a cell that takes data and press Enter, or select Insert Row from the Edit menu.
5. When you finish entering the list, tap OK and it will appear in the Input window. Sample data is available.

## Note

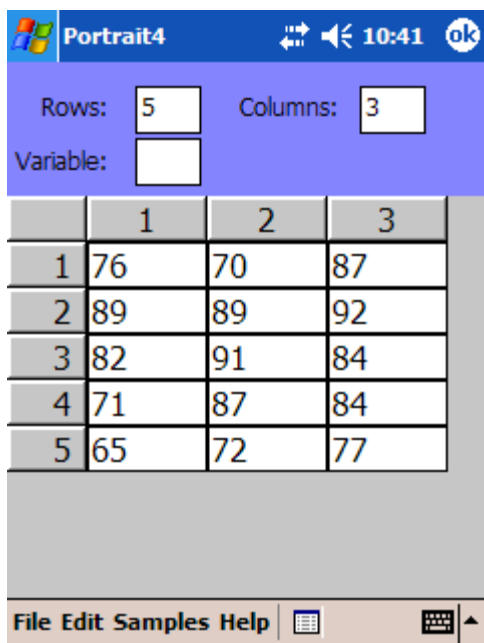
Lists can be saved and retrieved using the File menu in the List Editor. Use the Edit menu in the List Editor to cut, copy and paste the elements of a list. You can also insert and delete rows using this menu as well as send the data to one of four graphing templates listed.

When you enter a list in the List Editor and tap ok, the list appears in the Input window. The input window below shows the list entered from the List Editor above. You can also enter lists directly into the Input window by typing in a similar expression.



## Matrix Editor

With the Matrix Editor you can build a two-dimensional matrix of any size. The data input can be saved for use at a later time, which may be useful for very large matrices. The information can also be copied and pasted using the appropriate menus.



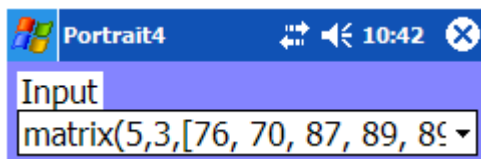
To enter a matrix:

1. Tap the Edit menu.
2. Open the Matrix Editor.
3. Enter the number of rows and columns you want the matrix to have. Assigning the matrix to a variable is optional. Enter the variable in the cell provided.
4. To build a matrix tap on the Build matrix button to create an empty grid of the desired size.
5. Enter the matrix elements in the grid cells and tap ok. The matrix appears in the Input window. Sample data is available.

### Important Note:

If you do enter a variable name it automatically assigns the matrix to the variable before the window closes. If there are any errors, a message will appear and the window will not close.

When you have entered a matrix in the Matrix Editor and tapped ok, the matrix appears in the Input window. The input window below shows the matrix generated from the Matrix Editor in the figure above. Matrices can also be entered directly into the Input window by typing in a similar expression.





## Note

If you do enter a variable name it automatically assigns the matrix to the variable before the window closes. If there are any errors, a message will appear and the window will not close.

If you make an error entering the elements of a matrix or if you simply want to make changes to the matrix you have input, you can. Highlight the matrix expression in the input window and select **Matrix Editor** again from the **Edit** menu to re-open the **Editor**. When the **Matrix Editor** is re-opened make your changes. Re-enter the matrix by tapping **ok** again.

Matrices can be saved using the **File** menu in the **Matrix Editor**. Use the **Edit** menu in the **Matrix Editor** to cut, copy and paste cells. You can also insert and delete columns using this menu as well as send the data to one of the graphing templates listed.

## Results in the Matrix Editor

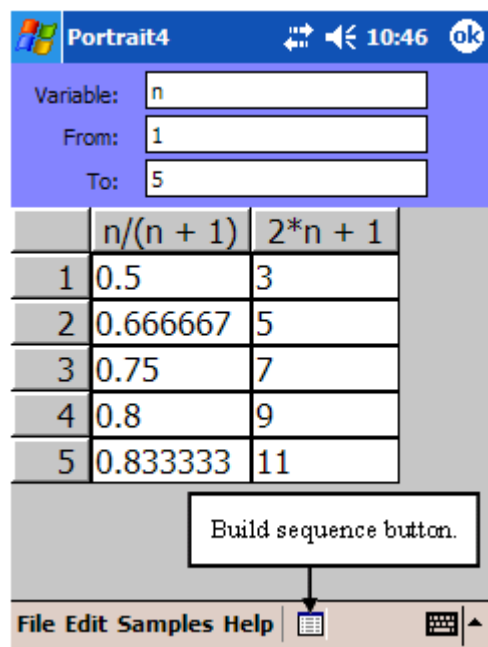
Any expression whose result is a matrix automatically opens the **Matrix Editor**, displaying the result of the calculation as an array.

## Sequence Editor

A sequence is a list of elements or numbers that succeed, or follow, each other in a specific order or pattern. With the **Sequence Editor** you can build a sequence of any size. Sequences can be saved for use at a later time, which may be useful for very large sequences. The data can also be copied and pasted using the appropriate menus.

To build a sequence:

1. Tap the **Edit** menu.
2. Open the **Sequence Editor**.
3. Enter a variable in the cell provided. This step is **not** optional.
4. Enter the range you want the variable evaluated **From** and **To** in the appropriate cells.
5. Below the cell where the range is entered there are two gray cells. Tap on the 2<sup>nd</sup> cell and enter the expression that you want the variable evaluated under.
6. When you have input the expression, tap the **Build Sequence** button to build the sequence.
7. To add a column, tap on the **Edit** menu and select **Insert Column**. Enter a new expression at the top of the new column. Tap the **Build Sequence** button again to fill in the new column.



## Note

Sequences can be saved and retrieved using the **File** menu in the **Sequence Editor**. Use the **Edit** menu in the **Sequence Editor** to copy and clear cells. You can also insert and delete columns using this menu as well as send the data to the graphing template listed.

## Table Editor

With the Table Editor you can build a table of values.

The screenshot shows the Table Editor interface. At the top, there are input fields for "Variable: x", "From: -2", "To: 2", and "Increment: 1". Below these is a table with 5 rows and 3 columns. The first column contains row numbers 1-5. The second column contains values from -2 to 2. The third column contains values from 0 to 4. Below the table is a "Build table button." and a menu bar with "File Edit Samples Help".

	x	x+2
1	-2	0
2	-1	1
3	0	2
4	1	3
5	2	4

To build a table of values:

1. Tap the Edit menu.
2. Open the Table Editor.
3. Enter a variable in the cell provided. This step is **not** optional.
4. Enter the range you want the variable to be evaluated From and To in the appropriate cells.
5. Enter the Increment you want the variable to increase by for each row.
6. Below the cell where the increment is entered there are two gray cells. Tap on the 2<sup>nd</sup> cell and enter the expression that you want the variable evaluated under.
7. When you have input the expression, tap the Build Table button to build a table of values.
8. To add a column to the table, tap on the Edit menu and select Insert Column. Enter a new expression at the top of the new column. Tap the Build Table button again to fill in the new column.

### Note

Tables can be saved and retrieved using the File menu in the Table Editor. Use the Edit menu in the Table Editor to copy and clear cells. You can also insert and delete columns using this menu as well as send the data to one of the graphing templates listed.

## Unit Converter

The Unit Converter is a measurement unit converter for area, length and distance, temperature, volume and capacity, and weight and mass. Choose a conversion category and units of measurement to convert from and to. Enter a number in the Convert From cell. The result appears in the Convert To cell. If you are working with very large or very small numbers you can enter numbers and see the results in Scientific notation. If using scientific notation you must use the format  $3.25e+6$ . For example, to convert  $458990 \text{ m}^2$  to  $\text{ft}^2$  using Scientific notation gives a result of  $4.940527e+6$ , which is equal to  $4.940527 \times 10^6$ . Tap on the Normal or Scientific button to switch between the two views.

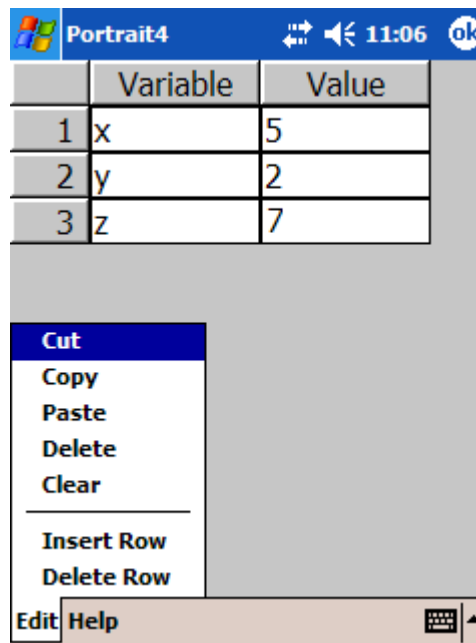
The screenshot shows the Unit Converter interface in Normal view. It has two columns: "Convert From:" and "Convert To:". Under "Convert From:" is a dropdown menu with "m<sup>2</sup>" selected and a text input field containing "458990". Under "Convert To:" is a dropdown menu with "ft<sup>2</sup>" selected and a text input field containing "4940527.2". Below these is a dropdown menu with "Area" selected. At the bottom is a "Normal" button.

The screenshot shows the Unit Converter interface in Scientific view. It has two columns: "Convert From:" and "Convert To:". Under "Convert From:" is a dropdown menu with "m<sup>2</sup>" selected and a text input field containing "458990". Under "Convert To:" is a dropdown menu with "ft<sup>2</sup>" selected and a text input field containing "4.940527e+6". Below these is a dropdown menu with "Area" selected. At the bottom is a "Scientific" button.

## Variable Editor

A variable represents a value or other mathematical concept. The Variable Editor lists all of the assigned variables. Assigned variables can be added or deleted using the Edit menu. Deleting a row will unassign a variable. When entering variables certain keywords and symbols **cannot** be used. Symbols indicating operations, relations, or symbolic constants cannot be used as variables. The names of functions or certain abbreviations cannot be used. Nor can specific words used in the graphing package to describe the graphs. Using one of these reserved words as a variable name generates a syntax error. If an error is generated the window will not close and the row containing the error is highlighted.

A variable can hold a value of any type, including a list or matrix or even an expression to be evaluated later. The name of a variable must start with a letter and contain only letters, numbers, or underscores ( \_ ).



### To Assign a Value to a Variable in the Input Window

- $x:= 5$  assigns  $x$  the value of 5. You can also assign any expression to  $x$  and evaluate it later.  
**Example 1:**  $x:= 5 + 6$ . The expression remains unevaluated. If you then use  $x$  in an expression, the calculator looks up the expression, evaluates it and returns the answer. You can assign values or expressions to more than one variable.  
**Example 2:**  $x:=2*y; y:=2$ . Evaluating  $x$  now returns the answer 4. If you now set  $y:=3$  and evaluate  $x$ , the result is 6.  
Remember you can also assign matrices or lists to a variable.  
**Example 3:**  $x:=\{1, 2, 3, 4\}; \text{sum}(x) = 10$ .
- If you are using the List, Matrix or Sequence Editor there is a field available to enter a variable that you want to assign to the list, sequence or matrix.

### To Unassign a Variable

The Edit menu includes two operations that allow you to unassign a variable or unassign all variables, if more than one is assigned. Tap the Edit menu open and select the operation you want.

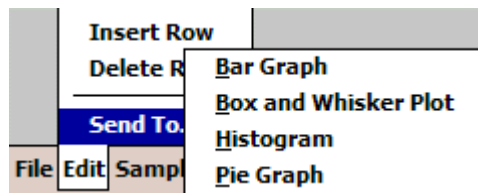
- **Unassign** – If you select this, `unassign( )` appears in the Input window. Enter the variable you want unassigned. Tap the equal sign to remove the assigned variable from the memory.
- **Unassign All** – If you select this, `unassign(all)` appears in the Input window. Tap the equal sign to remove all of the assigned variables from the memory.

The result of assigning a value to a variable, or unassigning one or more variables, is the Boolean value **true**.

## Generating a Graph from the List, Matrix, Sequence or Table Editor

The data in each of the List, Matrix, Sequence and Table Editors can be used to create graphs. The data in these editors can be sent to one of the graphing templates listed to make a visual representation of the data. To generate a graph from an Editor:

1. Enter the data in the List, Matrix, Sequence or Table Editor.
2. Tap the Edit menu open.
3. Open Send To ... in the Edit menu. Send To ... provides a list of graphing templates available for the Editor you are using. Each Editor has different graphing templates available. The List Editor has the choices shown here.
4. Select the type of graph you want by tapping on your choice. This opens the chosen graphing template with the data from the Editor already entered.
5. Tap the pencil icon to draw the graph.



### Tip

- If you only want a portion of the data graphed you may highlight the information to be sent to the graphing template otherwise the entire set of data will be sent. You can select any rectangular region of cells by tapping in one corner of the region and dragging the stylus to the other corner.

## Calculus

Calculus is a branch of mathematics that was developed by Gottfried Leibniz and Isaac Newton during the 17<sup>th</sup> century as a way of solving measurement problems in physics and geometry. Today calculus has far reaching applications in many fields that study the dynamics of change. Using calculus you can measure and describe rates of change and their impact. The word calculus is derived from the Latin for pebble. The ancient Romans and Greeks, making the first “calculators” used pebbles as counters on counting tables.

There are two branches of calculus: differential calculus and integral calculus. Differential calculus deals with measuring the instantaneous rate of change of one quantity relative to the change in another quantity and finding the slope of the tangent to a curve at a certain point. Integral calculus involves finding the area under a curve. These two branches of calculus are inverses of one another.

To get to the calculus functions tap the following: **Functions → Calculus**

### ▪ Derivative of a Function

**Input:** **deriv(function, variable, value, delta)** (Delta is optional.)

**Description:** Finds the derivative of the function with respect to the variable. You must provide the value at which to find the derivative. The derivative of a function is a function whose value at  $x$  is the slope of the tangent to the graph of the original function. The derivative can also be thought of as a function whose value at  $x$  is the instantaneous rate of change of the original function with respect to  $x$ , at point  $x$ .

**Example:**  $\text{deriv}(2x^3+1, x, -1) = 6.0000$ . The derivative of  $2x^3 + 1$  with respect to  $x$  at  $x = 1$  is 6.

- **Integral of a Function**

**Input:** **fint(function, variable, lower limit, upper limit)**

**Description:** Finds the numerical integral of the function with respect to the variable. A lower and an upper limit must be given. The integral of a function equals the area of the region between the graph of the function and the  $x$ -axis, with sections below the  $x$ -axis counting as negative areas.

**Example:**  $\text{fint}(2*x+3, x, 1, 4) = 24$ . The integral of the expression, from 1 to 4 is 24. The signed area of the region between the graph of the function within the specified limits and the  $x$ -axis is equal to 24.

- **Local Minimum of a Function**

**Input:** **fmin(function, variable, lower limit, upper limit)**

**Description:** Finds the value at which a local minimum of the function occurs, with respect to the variable. The lower and upper limits must be given. A local minimum is any vertex of a graph of a function that is lower than its neighboring points. It is not necessarily the absolute minimum.

**Example:**  $\text{fmin}(x^3-12*x+1, x, -3, 3) = 2$ . A local minimum of this function in the interval  $(-3, 3]$  is 2. Answers generated are approximate.

- **Local Maximum of a Function**

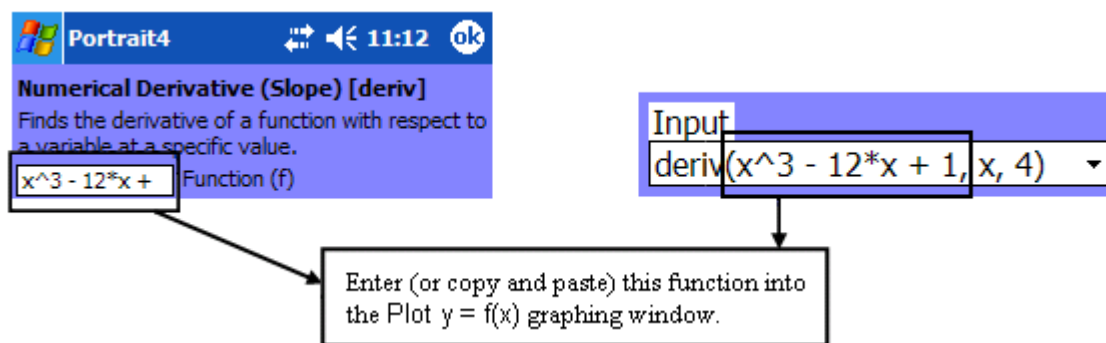
**Input:** **fmax(function, variable, lower limit, upper limit)**

**Description:** Finds the value at which a local maximum of the function occurs, with respect to the variable and the lower and upper limits given. A local maximum is any vertex of a function that is higher than its neighboring points. It is not necessarily the absolute maximum.

**Example:**  $\text{fmax}(x^3-12*x+1, x, -3, 1) = -2$ . A local maximum of this function in the interval  $(-3, 1]$  is -2. Answers generated are approximate.

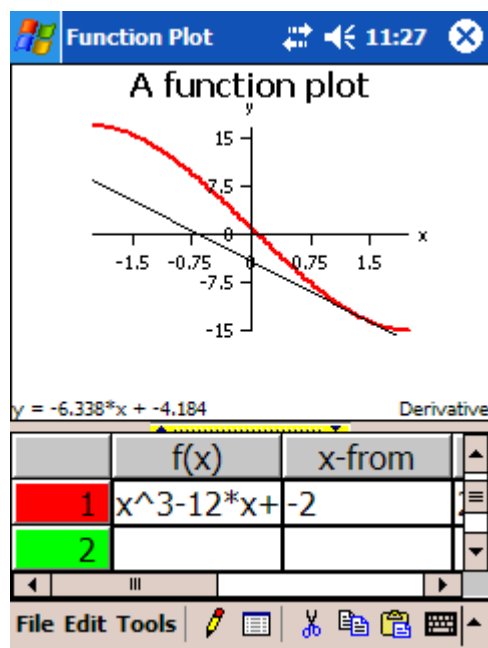
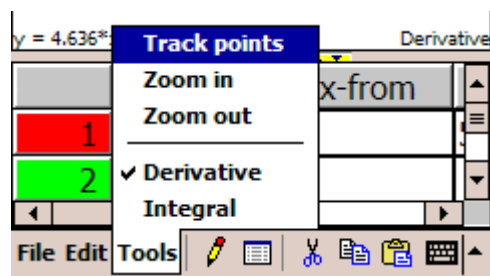
## Graphing Calculus Functions

Graphing functions will, in most cases, provide a deeper understanding of the functions being explored. To graph a function select the Plot menu on the calculator and choose the Plot  $y = f(x)$  option. Enter (or copy and paste) the function you want graphed under the  $f(x)$  header. You must enter the domain of the function under the  $x$ -from and  $x$ -to headers. The range is optional.



The boxed portion of the template and the input window, above, represents the function that can be graphed. The “ $y =$ ” or “ $f(x) =$ ” part of the function is assumed. Copy this portion of the input and then open a Plot  $y = f(x)$  graphing window. Paste the function under the  $f(x)$  header. Enter the domain you want the function graphed over. The range is optional. To draw the graph, tap on the pencil icon

The Tools menu provides the option of graphing the derivative or the integral of the function. In this example, we selected Derivative and then tapped on the point on the graph that we wanted to see the derivative of. The graph is drawn and the equation provided in the bottom left corner.



## Complex Numbers

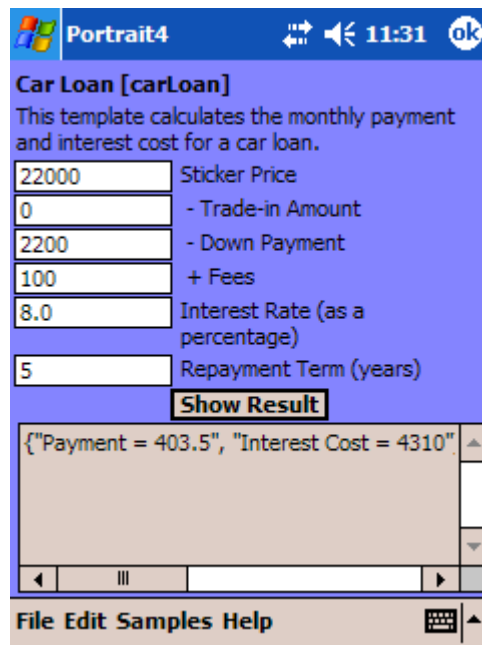
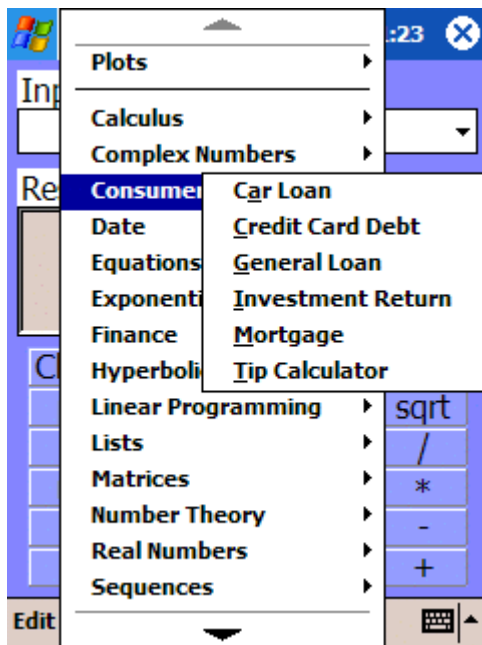
Complex numbers,  $z$ , are numbers in the form  $a + b*I$  where  $I^2 = -1$  and  $a$  and  $b$  are real numbers. To get to the complex number functions tap the following: **Functions → Complex Numbers**

- **Argument**  
**Input:**  $\text{arg}(z)$   
**Description:** Finds the argument of a complex number.  
**Example:**  $\text{arg}(I) = 1.570796$ . The argument of  $I$  is equal to 1.570796.
- **Conjugate**  
**Input:**  $\text{conj}(z)$   
**Description:** Finds the conjugate of a complex number.  
**Example:**  $\text{conj}(I) = -I$ . The conjugate of  $I$  is equal to  $-I$ .
- **Imaginary Part**  
**Input:**  $\text{imag}(z)$   
**Description:** Finds the imaginary part of a complex number.  
**Example:**  $\text{imag}(I) = 1$ . The imaginary part of  $I$  is 1.

- **Modulus**  
**Input:** **modulus(z)**  
**Description:** Finds the modulus of a complex number.  
**Example:** modulus(I) = 1. The modulus of I is 1.
  
- **Real Part**  
**Input:** **real(z)**  
**Description:** Finds the real part of a complex number.  
**Example:** real(I) = 0. The real part of I is 0.
  
- **Convert from Rectangular Form to Polar Form**  
**Input:** **toPolar(z)**  
**Description:** Converts a complex number in rectangular form to polar form.  
**Example:** toPolar(3 + I) = {3.162278, 0.321751} = (r,  $\theta$ )
  
- **Covert from Polar Form to Rectangular Form**  
**Input:** **toRect({list})** The list is equal to (r,  $\theta$ ).  
**Description:** Converts a complex in polar form to rectangular form.  
**Example:** toRect({2, pi}) = -2 + 0\*I

## Consumer Applications

As well as many plotting and calculation functions, *Portrait 4* includes helpful templates for everyday calculations. These are found in the Functions menu, under Consumer Applications. When you select an application, a template opens, similar to the Function templates. Enter any value or expression appropriate to the application in the entry cells. Tap on the **Show Result** button to display the result. Close the window by tapping OK in the top-right corner.



## Date

The date functions use a 24 hour clock. One am is entered as 1 while 1 pm is entered as 13. Dates are entered as day, month, and year. To get to the date functions tap the following: **Functions → Date**

- **Adjust a Date**

**Input:** **adjDate(date, day, months, years)**

**Description:** Adds or subtracts a number of days, months and years from a given date.

**Example:** `adjDate(makeDate(5, 2, 2005, 10, 30, 0), 3, 2, -10)` = 10:30:00 am, Saturday, April 8, 1995. Three days and two months were added to the original date and 10 years were subtracted.

- **Adjust a Time**

**Input:** **adjTime(date, hours, minutes, seconds)**

**Description:** Adds or subtracts a number of hours, minutes and seconds from a given date.

**Example:** `adjTime(makeDate(5, 2, 2005, 10, 30, 0), 4, 10, 45)` = 02:40:45 pm, Saturday, February 5, 2005. Four hours 10 minutes and 45 seconds were added to the original time.

- **Difference Between Two Dates**

**Input:** **diffDates(date 1, date 2)**

**Description:** Finds the difference in days between two dates.

**Example:** `diffDates(makeDate(5, 2, 2005, 0, 0, 0), makeDate(9, 6, 2014, 0, 0, 0))` = -3411. The difference between the two dates is 3411 days.

- **Find the Hours**

**Input:** **hours(date)**

**Description:** Finds the hours in a given date/time.

**Example:** `hours(makeDate(5, 2, 2005, 10, 30, 0))` = 10. In the given date and time there are 10 hours.

- **Create a Date Object**

**Input:** **makeDate(day, month, years, hours, minutes, seconds)**

**Description:** Creates a date/time object.

**Example:** `makeDate(4, 3, 1975, 4, 16, 32)` = 4:16:32 am, Tuesday, March 4, 1975

- **Find the Minutes**

**Input:** **minutes(date)**

**Description:** Finds the minutes in a given date/time.

**Example:** `minutes(makeDate(5, 2, 2005, 10, 30, 0))` = 30. In the given date and time there are 30 minutes.

- **Find the Month**

**Input:** **month(date)**

**Description:** Finds the month in a given date.

**Example:** `month(makeDate(5, 2, 2005, 10, 30, 0))` = 2. In the given date the month is 2, February.

- **Day of the Month**

**Input:** **monthDay(date)**

**Description:** Finds the day of the month a given date lands on.

**Example:** `monthDay(makeDate(5, 2, 2005, 10, 30, 0))` = 5. In the given date the day of the month is the 5<sup>th</sup>.



- **Current Date and Time**  
**Input:** `now( )`  
**Description:** Returns the current local date and time.  
**Example:** Click on the Show Result button to find the current date and time.
  
- **Find the Seconds**  
**Input:** `seconds(date)`  
**Description:** Finds the seconds in a given date/time.  
**Example:** `seconds(makeDate(5, 2, 2005, 10, 30, 12)) = 12`. In the given date and time there are 12 seconds.
  
- **Find the Time**  
**Input:** `time(date)`  
**Description:** Finds the time in a given date/time.  
**Example:** `time(makeDate(5, 2, 2005, 10, 30, 12)) = {10, 30, 12}`. In the given date the time is 10:30:12 am.
  
- **Day of the Week**  
**Input:** `weekDay(date)`  
**Description:** Finds the day of the week that a given date lands on.  
**Example:** `weekDay(makeDate(5, 2, 2015, 10, 30, 12)) = Thursday`. The date February 5, 2015 falls on a Thursday.
  
- **Find the Year**  
**Input:** `year(date)`  
**Description:** Finds the year in a given date.  
**Example:** `year(makeDate(5, 2, 2015, 10, 30, 12)) = 2015`. The year is 2015.

## Equations

To solve equations tap the following: Functions → Equations

The equation solving functions available are as follows:

- **Solving a Single Equation**  
**Input:** `fsolve(equation, variable, lower bound, upper bound)`  
**Description:** Finds a value for the variable (between the lower and the upper bounds given), making the function equal to zero. When solving for a variable in an equation it is assumed that the equation is equal to 0. You can, however, also enter equations in the form  $x^2 = x + 1$ .  
**Example:** `fsolve(x^3+3*x^2-10*x, x, -10, 10) = -5`. The solution of  $x$  for this equation between the upper and lower bounds given is -5.

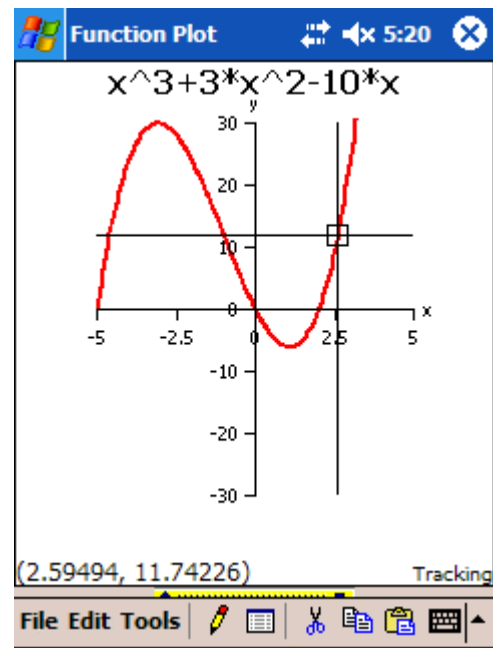
## Graphing a Single Equation

Graphing an equation can help you in determining the **lower** and **upper bounds** that you want to evaluate the equation between. A graph shows you how many roots an equation has. Remember that each time a graph crosses the  $x$ -axis there is a different root of the equation. For example, the graph of the equation  $x^3 + 3x^2 - 10x$  crosses the  $x$ -axis three times, indicating three roots. To graph the equation open a Plot  $y = f(x)$  window. Type (or copy and paste) the equation you want graphed under the  $f(x)$  header. You must enter the domain of the function under the  $x$ -from and  $x$ -to headers. The range is optional but may be helpful in getting a clearer picture of the graph.

To find all three roots of this equation evaluate `fsolve` three times, using different lower and upper bounds each time. To approximate the appropriate lower and upper bounds for each root look at the graph and estimate the value where the graph crosses the  $x$ -axis. Choose a lower bound less than this value and an upper bound greater than this value making sure that the graph crosses the  $x$ -axis only once between these two bounds.

In the graph of  $x^3 + 3x^2 - 10x$  the graph crosses the  $x$ -axis between  $-6$  and  $-4$ . These are appropriate choices for lower and upper bounds to determine the value of  $x$  at this point. The graph also crosses the  $x$ -axis between  $-1$  and  $1$  and again between  $1$  and  $3$ . Calculating `fsolve` with these three sets of lower and upper bounds gives you the three correct roots of the function.

The tracking tool, which is available on the graph, is helpful in determining suitable bounds and the approximate root locations.



### ▪ Solving Systems of Linear Equations

**Input:** `solve({equation 1, equation 2}, {list of variables})`

**Example:** `solve({x + y = 1, x - y = 0}, {x, y})` finds the common solution for  $x$  and  $y$  in the two given equations. In this example,  $x = 0.5$  and  $y = 0.5$ .

### Tips

- When solving systems of linear equations you can enter as many equations as you want. Remember to use braces `{ }` around the list of equations. Equations must also be separated by commas or you get an error.
- The system of equations must be followed by a complete list of the variables in the equations. A comma must separate the list of equations from the list of variables. The list of variables must be in braces `{ }` and the variables must be separated by commas. If not, a syntax error occurs.
- **The equations entered must be linear equations.** If you enter a non-linear equation you get an error message. This function does not solve systems of inequalities.

If the system of equations you enter is **inconsistent** or has no common solution you get a message in the **Result** window indicating this. For example, `solve({x + y = 1, x + y = 2}, {x, y})` produces the following result, *error: inconsistent system of equations*.

You can assign a list of equations to a variable.

**Example:** `L:={x + y = 1, x - y = 0}` so, `solve(L, {x, y})` = “ $x = 0.5, y = 0.5$ ”

If you enter only one equation instead of a system of equations, the calculator solves for one variable in terms of the other.

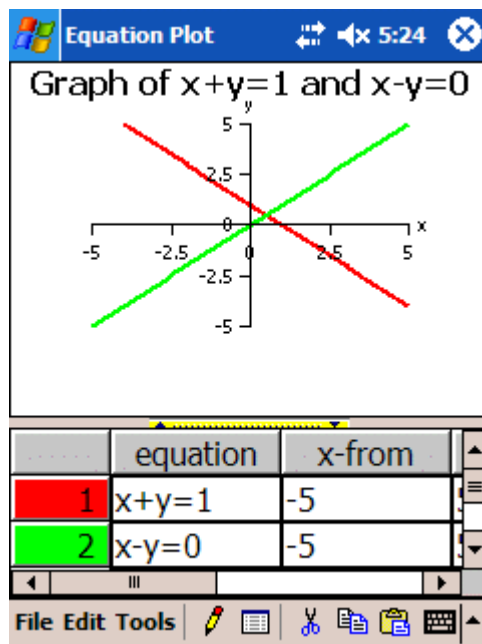
**Example:** `solve({2*x + y = 2}, {x, y})` solves for  $x$  in terms of  $y$ , producing the result  $x = 1 - 0.5*y$ .

To solve this same equation for  $y$  in terms of  $x$ , enter the equation the same way but reverse the order of the variables.

**Example:** `solve({2*x + y = 2}, {y, x})` produces the result  $y = 2 - 2*x$ .

## Graphing a System of Linear Equations

When solving systems of linear equations you may find it helpful to graph the equations. To graph equations go to the Plot menu and open the Plot Equations window. Simply enter the equations and the appropriate domain and range. Tap on the pencil icon to see the graph.



### Tip

- You can use the tracking tool on the graph to locate the approximate solution.

## Exponential and Logarithmic Functions

Exponential and logarithmic functions are examples of transcendental functions, that is, functions that are not algebraic. Exponential and logarithmic functions are inverses of one another. They have a wide variety of applications in mathematics and science. All of the exponential and logarithmic functions are defined for complex numbers.

John Napier (1550 – 1617) originally developed the natural logarithm as an aid for multiplication (and division). Henry Briggs (1561 – 1630), who proposed that base-10 logarithms would be more useful, later developed the common logarithm.

To get to the exponential/logarithmic functions tap the following: **Functions → Exponential**

### Exponential Function

**Input:**  $\text{exp}(x)$

**Description:** Denotes the exponential function  $e^x$ . This is equivalent to the calculator expression  $e^x$ . Exponential functions have a constant base and a variable exponent. They are continuous.

**Example:**  $\text{exp}(2) = 7.389$

- Logarithm**  
**Input:**  $\ln(x)$  or  $\log(x)$   
**Description:** Denotes the natural logarithm function. The natural logarithm has an irrational base, the constant  $e$ , which is named for Leonard Euler.  
**Example 1:**  $\ln(e^2) = 2$   
**Example 2:**  $\log(-1) = 3.1415\pi$
- Base 10 Logarithm**  
**Input:**  $\log_{10}(x)$   
**Description:** Denotes the base 10 logarithm. It is also called the common logarithm.  
**Example:**  $\log_{10}(100) = 2$
- Base  $a$  Logarithm of  $x$**   
**Input:**  $\log[a](x)$   
**Description:** Finds the logarithm to base  $a$  of  $x$ .  
 This function must be entered in the Input window. Place square brackets around the first number and curved brackets around the second number.  
**Example:**  $\log[2](10) = 3.32$

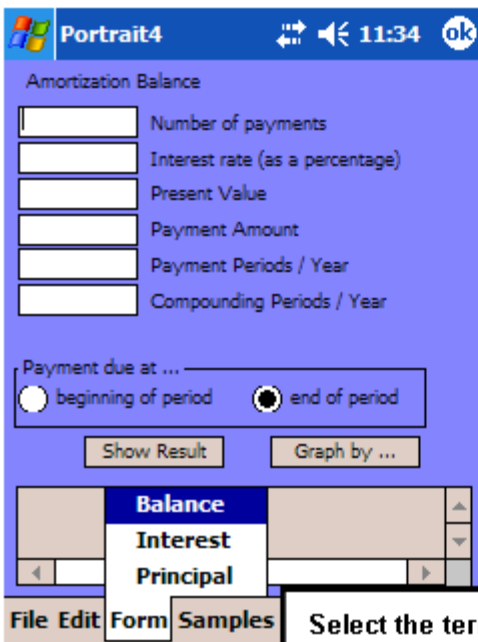
## Finance: Amortization

Amortization is a gradual reduction of an amount over time. The following amortization functions calculate the outstanding balance, the sum of interest and the sum of the principal for an amortization schedule. Generally speaking, an amortization schedule shows the retiring of debt over time including the portion of interest and principal that each payment represents and the outstanding balance on the debt after each payment is made.

To get to the amortization functions tap the following:

Functions → Finance → Amortization Template

- Cash to be received** must be entered as a **positive** number.
- Cash to be paid** must be entered as a **negative** number.



To use the Amortization Template, determine the term you want to solve for and select that term from the Form menu. For example, if you want to calculate the interest paid, select **Interest** from the Form menu at the bottom of the template. This calls up the proper version of the template. Fill in the fields with the appropriate information and tap **Show Result** for the answer.

To now do another calculation, simply tap on the Form menu again and make your selection from the available list. Again, this will call up the proper version of the Amortization Template. Make sure that each of the fields contains data and tap **Show Result** for the answer.

There is sample data available for each of the terms that the Amortization Template can solve for in the **Samples** menu.

Select the term you want to solve for from the Form Menu.

- **Amortization Schedule Balance**

**Input:** **amort\_bal** (number of payments, interest, present value, payment amount, payment periods, compounding periods)

**Description:** Finds the balance for an amortization schedule, that is, the outstanding balance still to be paid.

**Example:**  $\text{amort\_bal}(24, 7.5, 65000, -500, 12, 4) = 62516.90$ . With a present value of \$65,000, an interest rate of 7.5% compounded quarterly, making monthly payments of \$500, at the end of 24 payments the balance remaining is \$62,516.90.

- **Amortization Schedule Interest Sum**

**Input:** **amort\_int**(range of payments, interest, present value, payment, payment periods per year, compounding periods per year)

**Description:** Finds the sum of the interest for a given range of payment periods, that is, the amount of interest that has been paid over a specified range of payments.

**Example:**  $\text{amort\_int}(1, 24, 7.5, 65000, -500, 12, 4) = -9516.90$ . Starting with a present value of \$65,000, an interest rate of 7.5% compounded quarterly, and making monthly payments of \$500; at the end of 24 payments \$9,516.90 in interest will have been paid.

- **Amortization Schedule Principal Sum**

**Input:** **amort\_prn** (range of payments, interest, present value, payment, payment periods per year, compounding periods per year)

**Description:** Finds the sum of the principal for a given range of payment periods, that is, the amount of the principal that has been paid back over a specified range of payments.

**Example:**  $\text{amort\_prn}(1, 24, 7.5, 65000, -500, 12, 4) = -2483.10$ . Starting with a present value of \$65,000, an interest rate of 7.5% compounded quarterly, and making monthly payments of \$500; at the end of 24 payments \$2,483.10 of the principal will have been paid back.

## Finance: Depreciation Template

Depreciation is a way of measuring the loss in value of an asset due to general wear and tear over a given period of time. To get to the depreciation template tap the following:

Functions → Finance → Depreciation Template

The Depreciation Template provides the choice of four different methods of depreciation. They are:

- Declining Balance – This method assumes that the asset declines in value, or depreciates, more at the beginning of the life of the asset.
- Declining Balance Crossover – This method begins with the declining balance method and then crosses over to the straight line method when the annual depreciation using the declining balance method becomes less than the annual depreciation using the straight line method.
- Straight Line - This method assumes that the asset depreciates by an equal amount over the life of the asset.
- Sum of Years Digits - This is an accelerated method of depreciation. It is expressed as a fraction where the denominator is the sum of digits from 1 to the number of useful years of life and the numerator is the number of years in reverse order.

**Depreciation Method**

Declining Balance

Declining Balance Crossover

Straight Line

Sum of Years Digits

Method	Options	Schedule	
--------	---------	----------	--

Once you have selected the method of depreciation you want to use, open the **Options** menu. Here you will enter information about the asset you are depreciating like the original cost, the salvage value, and the life of the asset. When all of the appropriate information is entered open the **Schedule** menu to see the schedule of depreciation. This information can be copied and pasted in to an Excel or word document.

Cost	<input type="text" value="1000"/>				
Salvage	<input type="text" value="200"/>				
Life	<input type="text" value="5"/>				
Month	<input type="text" value="1"/>				
Rate	<input type="text" value="8.0"/>				
<table border="1"> <tr> <td>Method</td> <td>Options</td> <td>Schedule</td> <td></td> </tr> </table>		Method	Options	Schedule	
Method	Options	Schedule			

	Value	Book Value	Amount				
1	784	984	16				
2	768.25	968.256	15.744				
3	752.76	952.76390	15.49209				
4	737.51	937.51968	15.24422				
5	722.51	922.51936	15.00031				
<table border="1"> <tr> <td>Method</td> <td>Options</td> <td>Schedule</td> <td></td> </tr> </table>				Method	Options	Schedule	
Method	Options	Schedule					

## Finance: Time Value of Money (TVM Template)

Money has different values at different points in time and calculating the time value of money can be an effective way to analyze a variety of financial instruments like mortgages, loans, leases and annuities. The difference in the value of money at different times is a result of the accumulation of interest. Use the TVM Template to calculate the number of payments, interest rate, present value, payment amount and future value. To get to the TVM functions tap the following:

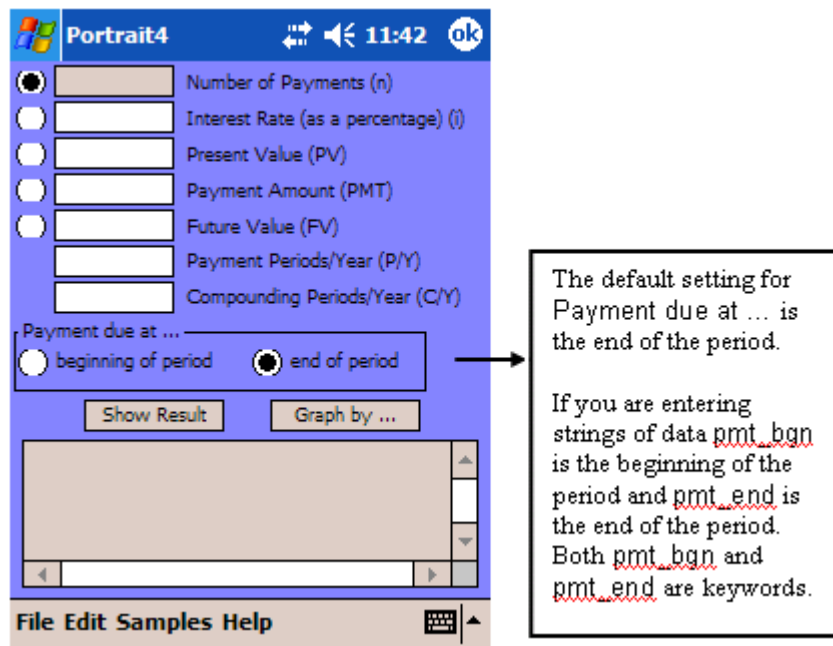
Functions → Finance → TVM Template

- **Cash to be received** must be entered as a **positive** number.
- **Cash to be paid** must be entered as a **negative** number.

To use the TVM Template, select the term that you want to solve for. For example, if you want to determine the number of payments, select Number of Payments ( $n$ ) in the template by tapping on the toggle button in front of number of payments. This action grays out the field, preventing you from entering data in it. Fill in the other fields with the information you have and tap **Show Result** for the answer.

To now do another calculation, simply tap on the toggle button in front of the new term you want to solve for. Again, this action grays out that specific field, preventing you from entering data. Make sure that each of the other fields contains data and tap **Show Result** for the answer.

There is sample data available for each of the five terms that the TVM Template solves for in the **Sample** menu.



- **Number of Payment Periods in an Annuity ( $n$ )**  
**Input:** `tvm_n(interest, present value, payment, future value, payment periods, compounding periods, pmt_due)`  
**Description:** Finds the number of payment periods.  
**Example:** `tvm_n(9.5, 10000, -500, 0, 4, 4, pmt_end) = 27.45`. An annuity with a present value of \$10,000 with an interest rate of 9.5% compounded 4 times a year provides equal payments of \$500 four times a year. There are 27.45 payment periods until the future value is equal to \$0.
- **Interest Rate of an Annuity**  
**Input:** `tvm_int(# of payments, present value, payment, future value, payment periods, compounding periods, pmt_due)`  
**Description:** Finds the annual interest rate.  
**Example:** `tvm_int(60, 12000, -300, 0, 12, 12, pmt_end) = 17.2734`. An annuity with a present value of \$12,000 provides 60 \$300 payments twelve times a year. The interest rate the annuity earns is 17.2734%.
- **Present Value of an Annuity ( $PV$ )**  
**Input:** `tvm_pv(# of payments, interest, payment, future value, payment periods, compounding periods, pmt_due)`  
**Description:** Finds the present value of an annuity.  
**Example:** `tvm_pv(24, 7.5, -100, 0, 12, 4, pmt_end) = 2223.28`. To receive 24 equal payments of \$100 twelve times a year, the present value of the annuity, earning 7.5% interest compounded four times a year, is \$2,223.28.
- **Amount of Each Payment in an Annuity**  
**Input:** `tvm_pmt(# of payments, interest, present value, future value, payment periods, compounding periods, pmt_due)`  
**Description:** Finds the amount of each payment.  
**Example:** `tvm_pmt(36, 18, 5000, 0, 12, 12, pmt_end) = -180.76`. Given an annuity with a present value of \$5,000 that earns 18% interest compounded 12 times a year, you receive 36 equal payments 12 times a year until the future value is equal to \$0. Each payment is \$180.76.

- **Future Value of an Annuity (FV)**  
**Input:** **tvm\_fv(# of payments, interest, present value, payment, payment periods, compounding periods, pmt\_due)**  
**Description:** Finds the future value. The future value is the value an investment will have at a future time if it earns compound interest at a specified rate.  
**Example:**  $\text{tvm\_fv}(36, 18, 0, -145, 12, 12, \text{pmt\_end}) = 6855.02$ . The future value of the annuity is \$6,855.02.

## Finance: Cash Flow Analysis

To get to the Cash Flow Analysis functions tap the following:

Functions → Finance → Cash Flow Analysis

- **Number of Cash Flow Periods**  
**Input:** **cfCount({list})**  
**Description:** Returns the number of periods in the given cash flow list excluding the initial cash flow.  
**Example:**  $\text{cfCount}(\{1000, 1500, 2000, 2500, 3000\}) = 4$ . Excluding the initial cash flow period, there are 4 cash flow periods in this list.
- **Internal Rate of return**  
**Input:** **cfIRR({list})**  
**Description:** Returns the internal rate of return of the list of cash flows.  
**Example:**  $\text{cfIRR}(\{-3000, 1500, 2000, 2500, 3000\}) = 54.8$ . The internal rate of return of this list of cash flow is 54.8%.
- **Modified Internal Rate of Return**  
**Input:** **cfMIRR(interest rate, {list})**  
**Description:** Returns the modified internal rate of return of the list of cash flows.  
**Example:**  $\text{cfMIRR}(10, \{-5000, 4000, 3000, 2000\}) = 26.5773$ . The modified internal rate of return is 26.5773%.
- **Net Future Value**  
**Input:** **cfNFV(interest rate, {list})**  
**Description:** Returns the net future value of the list of cash flows.  
**Example:**  $\text{cfNFV}(10, \{-5000, 4000, 3000, 2000\}) = 3485.00$ . The net future value is \$3,485.
- **Net Present Value**  
**Input:** **cfNPV(interest rate, {list})**  
**Description:** Returns the net present value of the list of cash flows.  
**Example:**  $\text{cfNPV}(10, \{-5000, 4000, 3000, 2000\}) = 2618.33$ . The net present value is \$2,618.33.
- **Net Uniform Series**  
**Input:** **cfNUS(interest rate, {list})**  
**Description:** Returns the net uniform series of the list of cash flows.  
**Example:**  $\text{cfNUS}(10, \{-5000, 4000, 3000, 2000\}) = 1052.87$



- **Payback Period**  
**Input:** **cfPbk({list})**  
**Description:** Returns the period when the initial investment of a given cash flow will be paid back.  
**Example:** cfPbk({-5000, 3000, 3000, 5000}) = 1.666667. The initial investment will be paid back in 1.67 periods.
  
- **Profitability Index**  
**Input:** **cfProf(interest rate, {list})**  
**Description:** Returns the profitability index of the given list of cash flows.  
**Example:** cfProf(10, {-5000, 4000, 3000, 2000}) = 1.523666. The profitability index of this list is 1.52.
  
- **Cash Flow Total**  
**Input:** **cfTotal({list})**  
**Description:** Returns the sum of the given cash flow.  
**Example:** cfTotal({1000, 1500, 2000, 2500, 3000}) = 10000. The total of the cash flows in this list is \$10,000.

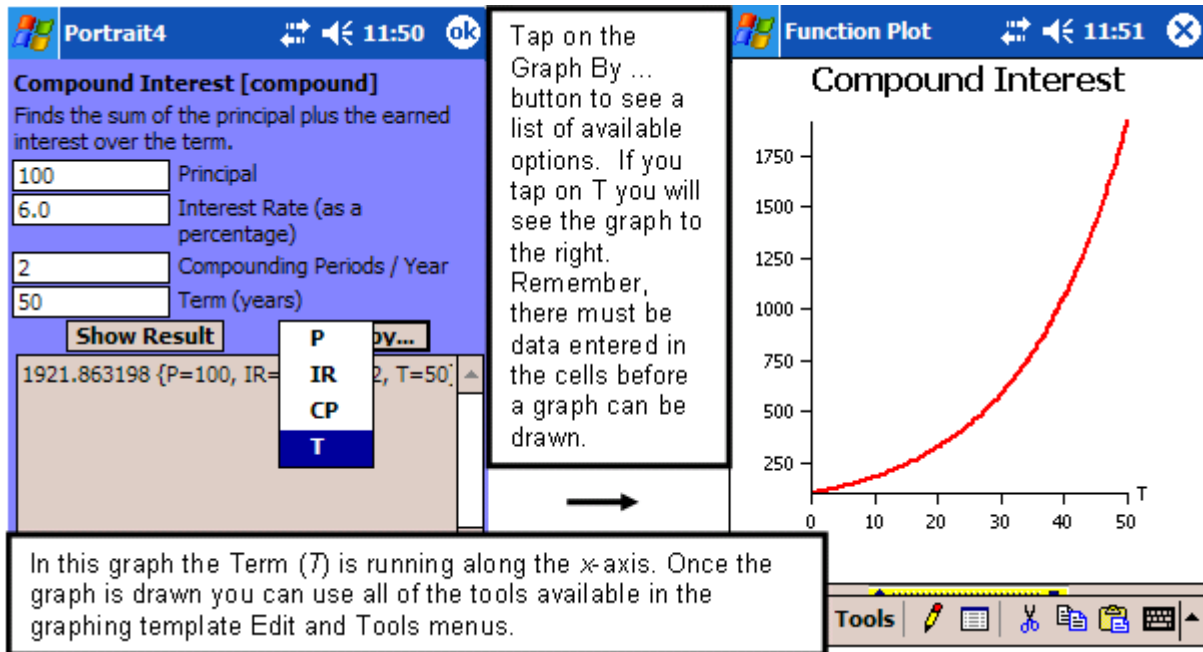
## Finance: Interest, Bonds and Net Present/Future Values

- **Compound Interest**  
**Input:** **compound(principal, interest, compounding periods, term)**  
**Description:** Finds the accumulated amount, that is, the sum of the principal plus the earned interest compounded  $c/y$  times a year over a given period of time, the term.  
**Example:** compound(100, 6, 2, 5) = 134.39. Starting with a principal of \$100 and an interest rate of 6% compounded twice a year, over a term of 5 years the accumulated amount is \$134.39.
  
- **Bond Accrued Interest**  
**Input:** **bondA(settlement date, maturity date, compounding periods, coupon rate, par value)**  
**Description:** Returns the accrued interest of a bond.  
**Example:** bondA(makeDate(1, 1, 2003), makeDate(15, 12, 2015), 2, 8, 1000) = 36.67. The interest that has accrued on the bond since the last interest payment is \$36.67.
  
- **Bond Price**  
**Input:** **bondP(settlement date, maturity date, compounding periods, coupon rate, residual value, annual yield)**  
**Description:** Returns the price of a bond.  
**Example:** bondP(makeDate(1, 1, 2003), makeDate(1, 1, 2015), 2, 6.5, 100, 5.524) = 108.48. The price of the bond is \$108.48.
  
- **Bond Yield**  
**Input:** **bondY(settlement date, maturity date, compounding periods, coupon rate, residual value, price)**  
**Description:** Returns the yield of a bond.  
**Example:** bondY(makeDate(1, 1, 2003), makeDate(1, 1, 2015), 2, 6.5, 100, 108.48) = 5.524053. The bond yield is 5.524%.

- **Effective Rate of Interest**  
**Input:** **eff(nominal rate, compounding periods)**  
**Description:** Finds the effective rate of interest, that is, the simple interest rate that would produce the same accumulated amount in one year as the nominal or stated rate compounded  $c/y$  times a year. The effective rate of interest is often used to compare compound rates of interest.  
**Example:**  $\text{eff}(6, 2) = 6.09$ . Given a nominal rate of interest of 6% compounded twice a year, the simple interest rate that would produce the same accumulated amount in one year is 6.09%.
- **Net Present Value**  
**Input:** **net\_pv(interest, initial cash flow, {list of cash flows})**  
**Description:** Finds the net present value for a series of cash inflows and outflows. The list of cash flow items must be in braces { }. **Cash inflows** must be entered as **positive** numbers while **cash outflows** must be entered as **negative** numbers.  
**Example:**  $\text{net\_pv}(8, 1000, \{-500, 2000, 1500\}) = 3442.46$ . Given an interest rate of 8%, an initial cash flow of \$1000, and the three cash flow amounts, the net present value is \$3442.46.
- **Nominal Rate of Interest**  
**Input:** **nom(effective rate, compounding periods)**  
**Description:** Finds the nominal or stated rate of interest.  
**Example:**  $\text{nom}(16.25, 8) = 15.19988$ . The nominal rate of interest is 15.19988%.
- **Simple Interest**  
**Input:** **simple(principal, interest, term)**  
**Description:** Finds the accumulated amount, that is, the sum of the principal and the simple interest earned on the principal over a given period of time, the term.  
**Example:**  $\text{simple}(100, 6, 5) = 130.00$ . Given a principal of \$100 and an interest rate of 6%, the accumulated amount over a term of 5 years is \$130.00.
- **Single Payment Future Value**  
**Input:** **spfv(interest rate, number of payments)**  
**Description:** Returns the future value of a single \$1.00 payment.  
**Example:**  $\text{spfv}(15, 10) = 4.05$
- **Single Payment Present Value**  
**Input:** **sppv(interest rate, number of payments)**  
**Description:** Returns the present value of a single \$1.00 payment.  
**Example:**  $\text{sppv}(15, 10) = 0.247$
- **Uniform Series Future Value**  
**Input:** **usfv(interest rate, number of payments)**  
**Description:** Returns the future value of a series of \$1.00 payments.  
**Example:**  $\text{usfv}(15, 10) = 20.3037$
- **Uniform Series Present Value**  
**Input:** **uspv(interest rate, number of payments)**  
**Description:** Returns the present value of a series of \$1.00 payments.  
**Example:**  $\text{uspv}(15, 10) = 5.0188$

## Graphing Finance Functions

Each of the templates in the Finance menu include a **Graph by...** button. Tap on this button to see a list of the variables available for graphing that particular function. For example, if you tap on the **Graph by ...** button in the Compound Interest template you are presented with four options: *P* (principal), *IR* (interest rate), *CP* (compounding periods per year) and *T* (term). If you tap on *T* (term) a graph of the data is drawn with the term along the *x*-axis and the compound interest along the *y*-axis.



## Hyperbolic Functions

Hyperbolic functions and their inverses can be used to solve a variety of problems in engineering and the physical sciences. They are a group of functions of an angle originally defined in terms of **trigonometric** or **exponential functions**. All of the hyperbolic functions are defined for complex numbers.

To get to the hyperbolic functions tap the following: **Functions** → **Hyperbolic**

- **Hyperbolic Sine**

**Input:**  $\sinh(x)$

**Description:** Denotes the hyperbolic sine function. The hyperbolic sine function is defined as  $(e^x - e^{-x})/2$  for every real number  $x$ .

**Example:**  $\sinh(e) = 7.544$

- **Hyperbolic Cosine**

**Input:**  $\cosh(x)$

**Description:** Denotes the hyperbolic cosine function. The hyperbolic cosine function is defined as  $(e^x + e^{-x})/2$ , for every real number  $x$ .

**Example:**  $\cosh(e) = 7.610$

- **Hyperbolic Tangent**  
**Input:**  $\tanh(x)$   
**Description:** Denotes the hyperbolic tangent function. The hyperbolic tangent function is defined as  $\sinh x / \cosh x$ .  
**Example:**  $\tanh(e) = 0.991$
  
- **Hyperbolic Cosecant**  
**Input:**  $\operatorname{csch}(x)$   
**Description:** Denotes the hyperbolic cosecant function. The hyperbolic cosecant function is defined as  $1/\sinh x$ , where  $x \neq 0$ .  
**Example:**  $\operatorname{csch}(e) = 0.133$
  
- **Hyperbolic Secant**  
**Input:**  $\operatorname{sech}(x)$   
**Description:** Denotes the hyperbolic secant function. The hyperbolic secant function is defined as  $1/\cosh x$ .  
**Example:**  $\operatorname{sech}(e) = 0.131$
  
- **Hyperbolic Cotangent**  
**Input:**  $\operatorname{coth}(x)$   
**Description:** Denotes the hyperbolic cotangent function. The hyperbolic cotangent function is defined as  $\cosh x / \sinh x$ , where  $x \neq 0$ .  
**Example:**  $\operatorname{coth}(e) = 1.009$

## Inverse Hyperbolic Functions

To get to the inverse hyperbolic functions tap the following: [Functions](#) → [Hyperbolic](#)

- **Inverse Hyperbolic of Sine**  
**Input:**  $\operatorname{asinh}(x)$  or  $\operatorname{arcsinh}(x)$   
**Description:** Denotes the inverse of the hyperbolic sine function. The inverse of the hyperbolic sine function is defined as  $y = \ln(x + \sqrt{x^2 + 1})$ , for all real numbers  $x$ .  
**Example:**  $\operatorname{arcsinh}(e) = 1.725$
  
- **Inverse Hyperbolic of Cosine**  
**Input:**  $\operatorname{acosh}(x)$  or  $\operatorname{arccosh}(x)$   
**Description:** Denotes the inverse of the hyperbolic cosine function. The inverse of the hyperbolic cosine function is defined as  $y = \ln(x + \sqrt{x^2 - 1})$ , for  $x \geq 1$ .  
**Example:**  $\operatorname{arccosh}(e) = 1.657$
  
- **Inverse Hyperbolic of Tangent**  
**Input:**  $\operatorname{atanh}(x)$  or  $\operatorname{arctanh}(x)$   
**Description:** Denotes the inverse of the hyperbolic tangent function. The inverse of the hyperbolic tangent function is defined as  $y = 1/2 \ln((1 + x)/(1 - x))$ , for  $|x| < 1$ .  
**Example:**  $\operatorname{arctanh}(0.5) = 0.549$

- **Inverse Hyperbolic of Cosecant**

**Input:** **acsch(x)** or **arccsch(x)**

**Description:** Denotes the inverse of the hyperbolic cosecant function. The inverse of the hyperbolic cosecant function is defined as  $y = \ln(1/x + (\sqrt{1 + x^2}/|x|))$ , for  $x \neq 0$ . This function can also be typed in or you can use the mathematical equivalent **arcsinh(1/x)**.

**Example:** **arccsch(0.5)** = 1.444

- **Inverse Hyperbolic of Secant**

**Input:** **asech(x)** or **arcsech(x)**

**Description:** Denotes the inverse of the hyperbolic secant function. The inverse of the hyperbolic secant function is defined as  $y = \ln((1 + (\sqrt{1 - x^2})/x))$ , where  $0 \leq x \leq 1$ . This function can be typed in or you can use the equivalent **arccosh(1/x)**.

**Example:** **asech(0.5)** = 1.317

- **Inverse Hyperbolic of Cotangent**

**Input:** **acoth(x)** or **arccoth(x)**

**Description:** Denotes the inverse of the hyperbolic cotangent function. The inverse of the hyperbolic cotangent function is defined as  $y = 1/2 \ln((x + 1)/(x - 1))$ , for  $x \leq -1$  or  $x \geq 1$ . This function can also be typed in or you can use the mathematical equivalent **arccoth(1/x)**.

**Example:** **acoth(e)** = 0.386

## Linear Programming

*Portrait 4* provides functions for solving linear programming problems. Generally speaking, linear programming problems deal with the optimization, either **minimization** or **maximization**, of linear functions subject to certain constraints imposed by a system of linear inequalities that are placed on that function. The system of inequalities defines a feasible region for the function to be optimized. In the graphing calculator, the constraints must be written as inequalities but the inequalities must not be strict. The implicit constraints, that the variables be non-negative, do not need to be included. The variables to be solved for do have to be indicated.

To get to the linear programming functions tap: **Functions** → **Linear Programming**.

- **Minimization Problems**

**Input**

**minimize(function, {constraints}, {variables})** ▾

**minimize(2\*x + y, {3\*x + y >= 10, 5\*x + 2\*y >= 6}, {x, y})**

**Description:** Finds the minimum value of a function under the given constraints and the  $x$  and  $y$  values at this minimum. Remember to use braces  $\{ \}$  around the constraints and the variables. The constraints must be written as inequalities. Use commas appropriately.

**Example:** **minimize(2\*x + y, {3\*x + y >= 10, 5\*x + 2\*y >= 6}, {x, y})** = 6.67;  $x = 3.33$ ,  $y = 0$ . The minimum value of the function under the given constraints is 6.67. This value occurs when  $x$  is equal to 3.33 and  $y$  is equal to 0.

- **Maximization Problems**

**Input:** **maximize(function, {constraints}, {variables})**

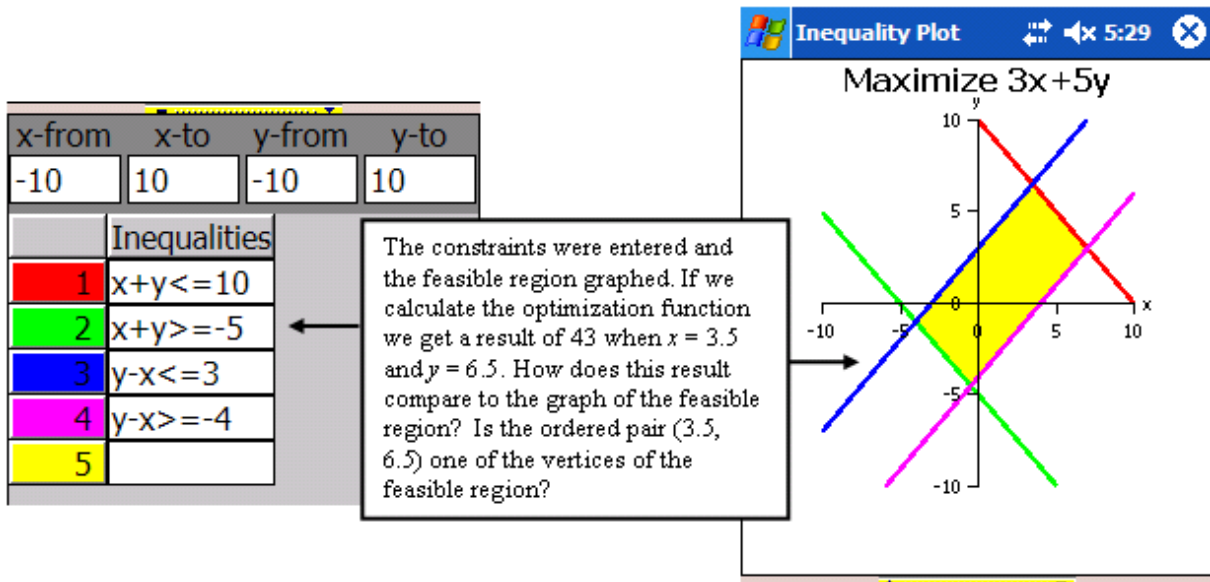
**Description:** Finds the maximum value of the function under the given constraints and the  $x$  and  $y$  values at this maximum. Remember to use braces  $\{ \}$  around the constraints and the variables. The constraints must be written as inequalities. Use commas appropriately.

**Example:**  $\text{maximize}(2*x + y, \{3*x + y \leq 10, 5*x + 2*y \leq 6\}, \{x, y\}) = 3; x = 0, y = 3$ . The maximum value of the function under the given constraints is 3. This value occurs when  $x$  is equal to 0 and  $y$  is equal to 3.

## Graphing Linear Inequalities

Linear functions are subject to restrictions imposed by a system of linear inequalities and equations. This system of inequalities defines a feasible region for the function to be optimized. Problems in two variables can be easily visualized and solved by considering only the vertices of the feasible region. One of the theorems of linear programming states that if a linear function has a maximum or minimum value over a feasible region then it must occur at one of the vertices of the feasible region. The inequalities underlined in the entry below define the feasible region. To graph these, select **Inequality Plot** from the **Plots** menu on the calculator. Enter these constraints/inequalities in the cells in the column labeled **Inequalities**.

**maximize(3\*x+5\*y,  $\{x + y \leq 10, x + y \geq -5, y - x \leq 3, y - x \geq -4\}$ , {x, y})**



## Lists

To get to the list functions tap the following: Functions → Lists

- **Augment**  
**Input:** `augment({list A}, {list B})`  
**Description:** This operation joins two lists, or joins matrix B to matrix A as new columns; the two matrices must have the same number of rows.  
**Example:** `augment({1, 2, 3, 4}, {4, 3, 2, 1}) = {1, 2, 3, 4, 4, 3, 2, 1}`
- **Length of a List**  
**Input:** `card({list A})`  
**Description:** Finds the cardinal number of elements in a list.  
**Example:** `card({1, 4, 0, 7, 7}) = 5`. This list contains 5 elements.
- **Cumulative Sum**  
**Input:** `cumSum({list A})`  
**Description:** Returns the cumulative sums of the elements in a list.  
**Example:** `cumSum({1, 2, 3, 4, 5}) = {1, 3, 6, 10, 15}`. In this example, 1 is the first element in the list but is not added to another element; the second element, 2, is added to the first to get 3; the third element is added to 3 to get 6 and so on until all elements have been added together.
- **List Differences**  
**Input:** `diffList({list A})`  
**Description:** Returns a list containing the differences between consecutive elements in the original list.  
**Example:** `diffList({1, 2, 3, 4, 5}) = {1, 1, 1, 1}`. In the example,  $2 - 1 = 1$ ;  $3 - 2 = 1$ ;  $4 - 3 = 1$ ; and  $5 - 4 = 1$ .
- **Fill a List**  
**Input:** `fill(element, length of list)`  
**Description:** Returns a list or matrix containing copies of a value.  
**Example:** `fill(5, 4) = {5, 5, 5, 5}`. In the example, the result is 4 copies of the value 5.
- **Maximum Value**  
**Input:** `max({list A})`  
**Description:** Finds the largest entry of a list or matrix.  
**Example 1:** `max({1, 2, -4, 5, 0}) = 5`. Five is the largest value in this set of data.  
**Example 2:** `max(primeSet(12)) = 3`. The largest prime number dividing 12 is 3.
- **Minimum Value**  
**Input:** `min({list A})`  
**Description:** Finds the smallest entry of a list or matrix.  
**Example 1:** `min({1, 2, -4, 5, 0}) = -4`. Negative 4 is the smallest value in this list of data.  
**Example 2:** `min(primeSet(12)) = 2`. The smallest prime number dividing 12 is 2.
- **Ascending Sort**  
**Input:** `sortA({list A})`  
**Description:** Sorts a list of numbers into ascending (increasing) order.  
**Example:** `sortA({-1, 2, 3, 4, -5}) = {-5, -1, 2, 3, 4}`

- **Descending Sort**  
**Input:** `sortD({list A})`  
**Description:** Sorts a list of numbers into descending (decreasing) order.  
**Example:** `sortD({1, 2, 3, 4, 5}) = {5, 4, 3, 2, 1}`
- **Sort Indices**  
**Input:** `sortIndices({list A},true=ascending/false=descending)`  
**Description:** Sorts a list and returns the indices (locations) of the sorted elements within the original list.  
**Example:** `sortIndices({-1, 8, 6, 0, 4}, true) = {1, 4, 5, 3, 2}`

## Matrices

Matrices are rectangular arrays of numbers. The numbers that comprise the array are called elements or entries. The elements are arranged in rows and columns and the size of the matrix is then stated as the number of rows by the number of columns. See the section on the **Matrix Editor** on page 14 for information on inputting matrices.

To get to the matrix functions tap the following: **Functions** → **Matrices**

### Arithmetic with Matrices

Simple arithmetic operations on matrices are not part of the **Matrices** function menu. These calculations are very easy to perform. Simply enter the matrices you want to add, subtract, or multiply, making sure the proper operation symbol separates them. An asterisk (\*) **must** be used to indicate multiplication. A caret (^) **must** be used to indicate an exponent. Tap on the equal sign or press **Enter** for the result.

- **Adding Matrices**  
The sum of matrices is obtained by adding corresponding elements in two or more matrices. Matrices must be the same size to be added. Adding matrices of different sizes gives an error.  
**Example:** `matrix(2, 2, [1, 2, 3, 4]) + matrix(2, 2, [4, 3, 2, 1]) = matrix(2, 2, [5, 5, 5, 5]).`
- **Subtracting Matrices**  
The difference between two matrices is obtained by subtracting the corresponding elements of two matrices. Matrices have to be the same size to be subtracted. Subtracting matrices of different sizes gives an error message.  
**Example:** `matrix(2, 2, [1, 2, 3, 4]) - matrix(2, 2, [4, 3, 2, 1]) = matrix(2, 2, [-3,-1, 1, 3]).`
- **Scalar Product**  
The scalar product is obtained by multiplying each element in a matrix by a given real number.  
**Example:** `3*matrix(2, 2, [1, 2, 3, 4]) = matrix(2, 2, [3, 6, 9, 12]).`
- **Multiplying Matrices**  
To multiply matrices, the number of columns in the first matrix must equal the number of rows in the second matrix.  
**Example:** `matrix(2, 3, [1, 2, 3, 4, 5, 6]) * matrix(3, 2, [6, 5, 4, 3, 2, 1]) = matrix(2, 2, [20, 14, 56, 41]).`



- **Matrix Exponentiation**  
Exponentiation can only be done on square matrices. The power must be a non-negative integer.  
**Example:** `matrix(2, 2, [1, 2, 3, 4])^2 = matrix(2, 2, [7, 10, 15, 22])`.
- **Dividing Matrices**  
Dividing one matrix by another matrix is not supported.
- **Adding and Multiplying Rows**  
**Input:** `addrow(matrix A, destination row, source row, factor)`  
**Description:** This operation replaces row  $i_1$  of a matrix with  $c \cdot \text{row } i_2 + \text{row } i_1$ .  
**Example:** `addrow(matrix(2, 2, [1, 2, 3, 4]), 1, 2, 3) = matrix(2, 2, [10, 14, 3, 4])`. In this example, multiply row 2 by 3 and add the result to row 1. Multiplying a row by **1** simply adds the source row to the destination row.
- **Adding and Multiplying Columns**  
**Input:** `addcol(matrix A, destination column, source column, factor)`  
**Description:** This operation replaces column  $j_1$  of a matrix with  $c \cdot \text{column } j_2 + \text{column } j_1$ .  
**Example:** `addcol(matrix(2, 2, [1, 2, 3, 4]), 2, 1, 2) = matrix(2, 2, [1, 4, 3, 10])`. In this example, multiply column 1 by 2 and add the result to column 2. Multiplying a column by **1** simply adds the source column to the destination column.
- **Augment**  
**Input:** `augment(matrix A, matrix B)`  
**Description:** This operation joins matrix  $B$  to matrix  $A$  as new columns. You can only augment a matrix with another matrix that has the same number of rows.  
**Example:**  
`augment(matrix(2, 2, [1, 2, 3, 4]), matrix(2, 2, [1, 0, 0, 1])) = matrix(2, 4, [1, 2, 1, 0, 3, 4, 0, 1])`.
- **Column Norm**  
**Input:** `cNorm(matrix A)`  
**Description:** Finds the column norm of a matrix.  
**Example:** `cNorm(matrix(3, 3, [1, 2, 3, 4, 5, 6, 7, 8, 9])) = 18`
- **Cross Product**  
**Input:** `crossProd(matrix A, matrix B)`  
**Description:** Finds the cross product of two 3-vectors (1 x 3 matrices).  
**Example:** `crossProd(matrix(1, 3, [1, 2, 3]), (matrix(1, 3, [4, 5, 6]))) = matrix(1, 3, [-3, 6, -3])`
- **Determinant**  
**Input:** `det(matrix A)`  
**Description:** Finds the determinant of a square matrix. A determinant is a scalar quantity that represents a defined alternating sum of products of elements of a square matrix, one from each row and column.  
**Example:** `det(matrix(2, 2, [1, 2, 3, 4])) = -2`. The determinant of this 2 x 2 matrix is -2.
- **Dimension**  
**Input:** `dim(matrix A)`  
**Description:** Returns the dimensions of a matrix.  
**Example:** `dim(matrix(2, 3, [1, 2, 3, 4, 5, 6])) = {2, 3}`

- **Dot Product**  
**Input:** **dotProd(matrix A, matrix B)**  
**Description:** Finds the dot product of two  $n$ -vectors ( $1 \times n$  matrices).  
**Example:** `dotProd(matrix(1, 3, [1, 2, 3]), matrix(1, 3, [1, 2, 3])) = 14`
- **Fill a List or Matrix**  
**Input:** **fill(matrix A, n)**  
**Description:** Returns a list or matrix containing copies of a value.  
**Example:** `fill(matrix(2, 2, [1, 2, 3, 4]), 6) = matrix(2, 2, [6, 6, 6, 6])`
- **Identity Matrix**  
**Input:** **ident(n)**  
**Description:** Returns the Identity matrix of a given size. The Identity matrix is a square matrix of size  $n$ , having 1s along the main diagonal and zeros everywhere else.  
**Example:** `ident(2) = matrix(2, 2, [1, 0, 0, 1])`
- **Inverse of a Matrix**  
**Input:** **inverse(matrix A)**  
**Description:** Finds the inverse of a square matrix. The inverse of a matrix is that matrix that when multiplied by the given matrix, returns the **identity matrix**. A matrix multiplied by its inverse will give you the identity matrix. Not all matrices have inverses.  
**Example:** `inverse(matrix(2, 2, [1, 2, 3, 4])) = matrix(2, 2, [-2, 1, 1.5, -0.5])`
- **Singular**  
**Input:** **isSingular(matrix A)**  
**Description:** Finds whether or not a matrix has an inverse. A matrix that does not have an inverse is **singular**. This function returns a Boolean (true or false) value. If the matrix is singular, that is, it does not have an inverse, the answer is true; otherwise, it is false.  
**Example:** `isSingular(matrix(2, 2, [1, 2, 3, 4])) = false`. This matrix has an inverse.
- **Maximum Value**  
**Input:** **max(matrix A)**  
**Description:** Finds the largest entry of a list or matrix.  
**Example:** `max(matrix(2, 2, [1, 2, 3, 4])) = 4`
- **Minimum Value**  
**Input:** **min(matrix A)**  
**Description:** Finds the smallest entry of a list or matrix.  
**Example:** `min(matrix(2, 2, [1, 2, 3, 4])) = 1`
- **Minor**  
**Input:** **minor(matrix A, row i, column j)**  
**Description:** Finds the  $(i, j)$ -th minor of the matrix  $A$ , that is, the determinant of the matrix obtained by removing row  $i$  and column  $j$  from  $A$ .  
**Example:** `minor(matrix(3, 3, [1, 2, 3, 4, 5, 6, 7, 8, 9]), 1, 3) = -3`. The determinant is -3.
- **Multiplying Rows by a Scalar**  
**Input:** **mulrow(matrix A, row i, factor)**  
**Description:** This operation multiples row  $i$  of a given matrix by a scalar value,  $c$  and stores the result in the row  $i$ .  
**Example:** `mulrow(matrix(2, 2, [1, 2, 3, 4]), 2, 3) = matrix(2, 2, [1, 2, 9, 12])`. The elements of row 2 of the matrix were each multiplied by 3, changing the elements in row 2 to 9 and 12.

- **Multiplying Columns by a Scalar**  
**Input:** **mulcol(matrix B, column j, factor)**  
**Description:** This operation multiplies column  $j$  of a given matrix by a scalar value,  $c$  and stores the result in the column  $j$ .  
**Example:** `mulcol(matrix(2, 2, [1, 2, 3, 4]), 1, 5) = matrix(2, 2, [5, 2, 15, 4])`. The elements of column 1 of the matrix were each multiplied by 5, changing the elements in column 1 to 5 and 15.
- **Frobenius Norm**  
**Input:** **norm(matrix A)**  
**Description:** Finds the Frobenius norm of a matrix. The Frobenius norm of a matrix is the square root of the sum of the absolute squares of its elements.  
**Example:** `norm(matrix(1, 3, [1, 2, 3])) = 3.741657`
- **Random Integer Matrix**  
**Input:** **randIntMatrix(m, n, A, B)**  
**Description:** Generates an  $m \times n$  matrix whose entries are random integers between  $A$  and  $B$ .  
**Example:** `randIntMatrix(2, 2, 1, 10) = matrix(2, 2, [1, 3, 6, 1])`
- **Random Matrix**  
**Input:** **randMatrix(m, n)**  
**Description:** Generates an  $m \times n$  matrix whose entries are random integers between 0 and 1.  
**Example:** `randMatrix(2, 2) = matrix(2, 2, [0.27-241, 0.360027, 0.255287, 0.98529])`
- **Rank of a Matrix**  
**Input:** **rank(matrix A)**  
**Description:** Finds the rank of a square matrix. The rank is the maximum number of linearly independent rows.  
**Example:** `rank(matrix(2, 2, [1, 2, 3, 4])) = 2`. The number of linearly independent rows is 2.
- **Resize a Matrix**  
**Input:** **redim(matrix A, m, n)**  
**Description:** Resizes a matrix to the specified dimensions.  
**Example:** `redim(matrix(1, 3, [1, 2, 3]), 2, 4) = matrix(2, 4, [1, 2, 3, 0, 0, 0, 0, 0])`
- **Row Norm**  
**Input:** **rNorm(matrix A)**  
**Description:** Finds the row norm of a matrix.  
**Example:** `rNorm(matrix(3, 3, [1, 2, 3, 4, 5, 6, 7, 8, 9])) = 24`
- **Row Reduction**  
**Input:** **rowReduce(matrix A)**  
**Description:** Finds the row-echelon form of a given matrix. A matrix is said to be in reduced row echelon form if the following conditions exist: The first nonzero entry in each row is 1. Each successive row has its first nonzero entry in a later column. All entries above and below the first nonzero entry of each row are 0. All full rows of zeros are the final rows of the matrix.  
**Example:** `rowReduce(matrix(2, 3, [1, 2, 3, 4, 5, 6])) = matrix(2, 3, [1, 0, -1, 0, 1, 2])`
- **Sum of a Matrix**  
**Input:** **sum(matrix A)**  
**Description:** Finds the sum of a list of numbers.  
**Example:** `sum(matrix(2, 2, [1, 2, 3, 4])) = 10`

- **Swap Two Rows of a Matrix**

**Input:** `swaprows(matrix A, source row, destination row)`

**Description:** This operation swaps the position of row  $i_1$  and row  $i_2$  in matrix  $A$ .

**Example:** `swaprows(matrix(3, 2, [1, 2, 3, 4, 5, 6]), 1, 3) = matrix(3, 2, [5, 6, 3, 4, 1, 2])`. Row 1 in the given matrix was swapped or changed positions with row 3 in the matrix.

- **Swap Two Columns of a Matrix**

**Input:** `swapcols(matrix A, source column, destination column)`

**Description:** This operation swaps the position of column  $j_1$  and column  $j_2$  in matrix  $A$ .

**Example:** `swapcols(matrix(2, 3, [1, 2, 3, 4, 5, 6]), 2, 3) = matrix(2, 3, [1, 3, 2, 4, 6, 5])`. Column 2 in the given matrix changed positions or was swapped with column 3.

- **Trace of a Matrix**

**Input:** `trace(matrix A)`

**Description:** Finds the trace of a matrix. The trace is the sum of the entries along the main diagonal in a square matrix of any size.

**Example:** `trace(matrix(2, 2, [1, 2, 3, 4])) = 5`. The entries along the main diagonal of this matrix are 1 and 4;  $1 + 4 = 5$ .

- **Transpose of a Matrix**

**Input:** `transpose(matrix A)`

**Description:** Finds the transpose of a given matrix. Interchanging the rows and columns of a given matrix so that Matrix  $m \times n$  is altered to Matrix  $n \times m$  transposes a matrix

**Example:** `transpose(matrix(2, 3, [1, 2, 3, 4, 5, 6])) = matrix(3, 2, [1, 4, 2, 5, 3, 6])`.

### Tip

- To perform an additional function on a matrix, highlight the matrix in the Input window and select the function from the **Matrices** menu. This wraps the selected function around the existing matrix. Tap the **Show Result** button in the template or tap = or **Enter** and the matrix operation is performed.

## Number Theory Functions

Number Theory is the field of mathematics that studies the properties and relations of integers. Some of the topics studied include divisibility, primality and factorization. In the descriptions below, the term “proper factors” indicates all factors of a number except the number itself.

To get to the number theory functions tap the following:

Functions → Number Theory

- **Combinations**

**Input:** `choose( $m$ ,  $n$ )`

**Description:** Finds the number of combinations of  $m$  objects chosen  $n$  at a time. The order of selection is not important. The integers  $m$  and  $n$  must be nonnegative with  $n$  less than or equal to  $m$ .

**Example:** `choose(6, 4) = 15`

- **Factor Set**  
**Input:** **factorSet( $n$ )**  
**Description:** Finds a list of all the factors of  $n$ . Factors of a number are any numbers that divide exactly into that given number.  
**Example:**  $\text{factorSet}(12) = \{1, 2, 3, 4, 6, 12\}$ . The factors of 12 are 1, 2, 3, 4, 6, and 12.
- **Fibonacci Number**  
**Input:** **fibonacci( $n$ )**  
**Description:** Finds the  $n$ -th Fibonacci number. Fibonacci numbers are numbers in the sequence 1, 1, 2, 3, 5, 8, 13, 21, 34, .... Each number in the sequence after the second number is the sum of the previous two numbers.  
**Example 1:**  $\text{fibonacci}(2) = 1$   
**Example 2:**  $\text{fibonacci}(15) = 610$
- **Greatest Common Divisor**  
**Input:** **gcd( $m, n$ )** or **gcf( $m, n$ )** or **hcf( $m, n$ )**  
**Description:** Finds the greatest common divisor of  $m$  and  $n$ . This is also referred to as the greatest common factor (gcf) or highest common factor (hcf). The greatest common divisor is the number that exactly divides into both of the numbers given.  
**Example:**  $\text{gcd}(56, 46) = 2$ . The largest number that exactly divides into both 56 and 46 is 2.
- **Abundant Number**  
**Input:** **isAbundant( $n$ )**  
**Description:** Finds whether  $n$  is an abundant number. An abundant number is a number for which the sum of its proper factors is greater than the number itself. This function returns a Boolean (true or false) value. If  $n$  is abundant, **isAbundant** returns true; otherwise, it returns false.  
**Example:**  $\text{isAbundant}(28) = \text{false}$ . The number twenty-eight is not an abundant number.
- **Amicable Numbers**  
**Input:** **isAmicable( $m, n$ )**  
**Description:** Finds whether  $m$  and  $n$  are amicable numbers. Two numbers are amicable if the sum of the proper factors of one number is equal to the other number. This function returns a Boolean (true or false) value. If  $m$  and  $n$  are amicable, **isAmicable** returns true; otherwise, it returns false.  
**Example:**  $\text{isAmicable}(6, 28) = \text{false}$ . The numbers 6 and 28 are not amicable.
- **Composite Number**  
**Input:** **isComposite( $n$ )**  
**Description:** Finds whether  $n$  is a composite number. A composite number is a number that has more than two factors. This function returns a Boolean (true or false) value. If  $n$  is a composite number, **isComposite** returns true; otherwise, it returns false.  
**Example:**  $\text{isComposite}(7) = \text{false}$ . Seven is not a composite number.
- **Deficient Number**  
**Input:** **isDeficient( $n$ )**  
**Description:** Finds whether  $n$  is a deficient number. A deficient number is a number where the sum of its proper factors is less than the number itself. This function returns a Boolean (true or false) value. If  $n$  is deficient, **isDeficient** returns true; otherwise, it returns false.  
**Example:**  $\text{isDeficient}(28) = \text{false}$ . The number twenty-eight is not a deficient number.

- **Perfect Number**  
**Input:** **isPerfect( $n$ )**  
**Description:** Finds whether  $n$  is a perfect number. A perfect number is a number for which the sum of its proper factors is equal to the number itself. This function returns a Boolean (true or false) value. If  $n$  is perfect, **isPerfect** returns true; otherwise, it returns false.  
**Example:** **isPerfect**(28) = true. Twenty-eight is a perfect number because  $1 + 2 + 4 + 7 + 14 = 28$ .
- **Prime Number**  
**Input:** **isPrime( $n$ )**  
**Description:** Finds whether  $n$  is a prime number. A prime number is a number that has exactly two factors, 1 and the number itself. This function returns a Boolean (true or false) value. If  $n$  is a prime number, **isPrime** returns true; otherwise, it returns false.  
**Example:** **isPrime**(7) = true. Seven is a prime number.
- **Square Number**  
**Input:** **isSquare( $n$ )**  
**Description:** Finds whether  $n$  is a square number; that is, the square of some integer. This function returns a Boolean (true or false) value. If  $n$  is a square number, **isSquare** returns true; otherwise, it returns false.  
**Example 1:** **isSquare**(7) = false.  
**Example 2:** **isSquare**(4) = true because 4 equals  $2^2$ .
- **Square Free**  
**Input:** **isSquareFree( $n$ )**  
**Description:** Finds whether  $n$  is divisible by the square of a number. This function returns a Boolean (true or false) value. If  $n$  is divisible by the square of a number, **isSquareFree** returns false; otherwise, it returns true.  
**Example 1:** **isSquareFree**(8) = false because 8 is divisible by 4 and 4 is a square of 2.  
**Example 2:** **isSquareFree**(70) = true because 70 is not divisible by the square of a number.
- **Least Common Multiple**  
**Input:** **lcm( $m, n$ )**  
**Description:** Finds the least or lowest common multiple of  $m$  and  $n$ . The least common multiple is the least number that is a multiple of both numbers.  
**Example:** **lcm**(56, 46) = 1288. The least common multiple of 56 and 46 is 1288.
- **Nines Excess**  
**Input:** **ninesExcess( $n$ )**  
**Description:** Finds the nines excess of  $n$ . To find the nines excess add the digits of a number (and repeat with the sum if necessary) until you arrive at a single digit. This single digit is the nines excess. To check an answer in basic arithmetic operations find the nines excess of each number and the nines excess of the answer. Take addition for example. If the nines excess of the answer does not equal the nines excess of the sum of the nines excesses of the addends, then the answer is wrong, however, if they are the same they may or may not be correct.  
**Example 1:** **ninesExcess**(56) = 2. Add  $5 + 6$  to get 11. Add  $1 + 1$  to get 2. The nines excess is 2.  
**Example 2:** Assume you want to check whether  $2 * 3 = 33$ . Since the product of the nines excesses of 2 and 3 is 6, and the nines excess of the answer is also 6, you cannot assume that the answer is correct. If the nines excesses had been different, you would know that the answer is wrong.

- **Permutations**  
**Input:** **perm( $m, n$ )**  
**Description:** Finds the number of permutations of  $m$  objects taken  $n$  at a time. The order of selection is important. The integers  $m$  and  $n$  must be nonnegative with  $n$  less than or equal to  $m$ .  
**Example:**  $\text{perm}(6, 4) = 360$ . There are 360 different permutations of 6 objects taken 4 at a time.
  
- **Prime Factors**  
**Input:** **primeSet( $n$ )**  
**Description:** Finds a list of the prime factors of  $n$ . A prime factor of a number is a factor that is a prime number. **Note:** The prime set function does not count multiplicities. It only provides the distinct primes, and does not indicate if they occur more than once.  
**Example:**  $\text{primeSet}(24) = \{2, 3\}$ . The number 24 has two prime factors, 2 and 3.
  
- **Shift Left**  
**Input:** **shiftl( $m, n$ )**  
**Description:** Shifts the integer  $m$  left by  $n$  bit positions.  
**Example:**  $\text{shiftl}(15, 4) = 240$ . The integer 15 is converted to a binary number and is shifted 4 bits to the left. The binary number after the shift is then converted back to an integer, 240.
  
- **Shift Right**  
**Input:** **shiftr( $m, n$ )**  
**Description:** Shifts the integer  $m$  right by  $n$  bit positions.  
**Example:**  $\text{shiftr}(64, 2) = 16$ . The integer 64 is converted to a binary number and is shifted 2 bits to the right. The binary number after the shift is then converted back to an integer, 16.
  
- **Sum of the Factors**  
**Input:** **sigma( $n$ )**  
**Description:** Finds the sum of all the factors of  $n$ . A factor of a number is a number that divides exactly into it.  
**Example:**  $\text{sigma}(12) = 28$ . The sum of the factors of 12 is equal to  $1 + 2 + 3 + 4 + 6 + 12 = 28$ .

## Real Numbers

To get to the real number functions tap the following:

Functions → Real Numbers

- **Absolute Value**  
**Input:** **abs( $x$ )**  
**Description:** Finds the absolute value of  $x$ . The absolute value of a number is its distance from zero on a number line.  
**Example:**  $\text{abs}(-12) = 12$ . The absolute value of -12 is 12.
  
- **Smallest Integer**  
**Input:** **ceil( $x$ )**  
**Description:** Finds the smallest integer that is greater than or equal to  $x$ . Integers are positive and negative whole numbers and 0.  
**Example:**  $\text{ceil}(2.56) = 3$ . The smallest integer that is greater than or equal to 2.56 is 3.

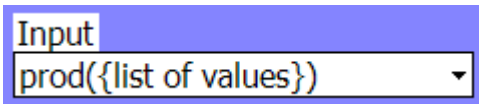
- **Largest Integer**  
**Input:** **floor(x)**  
**Description:** Finds the largest integer that is less than or equal to  $x$ . Integers are positive and negative whole numbers and 0.  
**Example:**  $\text{floor}(2.56) = 2$ . The largest integer that is less than or equal to 2.56 is 2.
- **Fractional Part**  
**Input:** **fracPart(x)**  
**Description:** Finds the fractional part of a real number.  
**Example:**  $\text{fracPart}(37.75) = 0.75$ . The fractional part of 37.75 is 0.75.
- **Integer Part**  
**Input:** **intPart(x)**  
**Description:** Finds the integer part of a real number.  
**Example:**  $\text{intPart}(37.75) = 37$ . The integer part of 37.75 is 37.

## Sequences

A sequence is a list of elements or numbers that succeed, or follow, each other in a specific order or pattern.

To get to the sequence functions tap the following: **Functions → Sequences**

- **Product of a List**



The image shows a screenshot of a calculator's input field. The field is highlighted with a blue border and contains the text "prod({list of values})". Above the input field, the word "Input" is written in a small font.

**Input 1 – List Form:** **prod({1, 2, 3, 4})**

**Description:** Finds the product of a list of values. **Note:** The list of values to be multiplied must be contained within braces { } within the brackets. This will differentiate this product from the product of the results obtained by evaluating an expression for a given range of values.

**Example:**  $\text{prod}(\{1, 2, 3, 4\}) = 24$ . Multiplying  $(1 \times 2 \times 3 \times 4) = 24$ .

**Input 2 – Parameter Form:** **prod(equation, variable:= first in range, ..., last in range)**

**Description:** Finds the product of a list of results for a variable evaluated under a given expression for a given range. The first value in the range must be less than or equal to the last value in the range and they must be integers. Also, between the first value in the range and the final value in the range there are only two dots (..) between the commas.

**Example:**  $\text{prod}(x + 2, x := 1, \dots, 4) = 360$ . Multiplying  $(1 + 2) \times (2 + 2) \times (3 + 2) \times (4 + 2) = 360$ .

**Input 3 – Sequence Form:** **prod(function, variable, {list})**

**Description:** Finds the product of a list of results for a variable evaluated under a given function for a given sequence.

**Example:**  $\text{prod}(x^2, x, \{1, 2, 5, -1\}) = 100$



### ▪ **Generate a Sequence**

**Description:** Finds a list of results for a variable evaluated under a given expression for a given range of values. Note: The first value in the range must be less than or equal to the last value in the range and they must be integers. Also, between the first value in the range and the final value in the range there are only two dots (..) between the commas.

**Input: seq(equation, variable:= first in range, ..., last in range)**

**Example:**  $\text{seq}(3*x + 2, x:= 5, \dots, 10) = \{17, 20, 23, 26, 29, 32\}$ . In this example,  $3*x + 2$  is evaluated for  $x$  ranging from 5 to 10 giving us the sequence  $\{17, 20, 23, 26, 29, 32\}$ .

### ▪ **Sum of a List**

**Input 1 – List Form: sum({list of values})**

**Description:** Finds the sum of a list of values given. Note: The list of values to be summed must be contained within braces { } within the brackets. This will differentiate this sum from the sum of the results obtained by evaluating an expression for a given range of values.

**Example:**  $\text{sum}(\{17, 20, 23, 26, 29, 32\}) = 147$ . Adding  $(17 + 20 + 23 + 26 + 29 + 32) = 147$ .

**Input 2 – Parameter Form: sum(equation, variable:= first in range, ..., last in range)**

**Description:** Finds the sum of a list of results for a variable evaluated under a given equation for a given range of values. The first value in the range must be less than or equal to the last value in the range and they must be integers. Also, between the first value in the range and the final value in the range there are only two dots (..) between the commas.

**Example:**  $\text{sum}(3*x + 2, x:= 5, \dots, 10) = 147$ . The result is the sum of the results of  $3*x + 2$  evaluated for  $x$  ranging from 5 to 10, that is,  $17 + 20 + 23 + 26 + 29 + 32 = 147$ .

**Input 3 – Sequence Form: sum(function, variable, {list})**

**Description:** Finds the sum of a list of results for a variable evaluated under a given function for a given sequence.

**Example:**  $\text{sum}(x^2, x, \{1, 2, 5, -1\}) = 31$

## Statistical Functions

Statistics is the area of mathematics that deals with collecting, analyzing, interpreting and presenting numerical data. Statistics can summarize a set of known data in a clear, concise manner such as in terms of its mean, median or standard deviation.

### Tip

- When entering data to calculate statistical functions, be sure to separate the data by commas.

To get to the statistical functions tap the following: **Functions → Statistics**

## Confidence Intervals

The following abbreviations are used to indicate the information to enter to calculate Confidence Intervals:

$x$	= number of successes in a sample
$x_1$	= number of successes in sample 1
$x_2$	= number of successes in sample 2
$n$	= total number of observations in a sample
$n_1$	= total number of observations in sample 1
$n_2$	= total number of observations in sample 2
mean	= mean of the data

Sx or std dev = standard deviation

Conf Lev or CL = confidence level

list = list of data (in braces, separated by commas)

Pooled = indicates if variances are to be pooled. Input **true** if the variances are to be pooled. Input **false** if the variances are not to be pooled. The **default** setting is **false**.

#### ▪ Confidence Interval for One Proportion

Input  
prop1zint(x, n, ConfLev) ▾

**prop1zint(3000, 5000, 0.95)**

**Description:** Finds a confidence level for an unknown proportion of successes in a binomial experiment.

**Example:** prop1zint(3000, 5000, 0.95) = {0.5864, 0.6136}. Given 3000 successes out of 5000 observations and a confidence level of 95%, the confidence interval is  $0.5864 < p < 0.6136$ .

#### ▪ Confidence Interval for the Difference of Two Proportions

**Input:** prop2zint(x<sub>1</sub>, n<sub>1</sub>, x<sub>2</sub>, n<sub>2</sub>, Conf Lev)

**Description:** Finds a confidence level for the difference between the proportions of successes in two populations.

**Example:** prop2zint(75, 100, 50, 100, 0.99) = {0.0796, 0.4204}. Given 75 successes out of 100 observations in the first population and 50 successes out of 100 observations in the second population and a 99% confidence level, the confidence interval is  $0.0796 < p < 0.4204$ .

#### ▪ Two Sample t Interval

**Input 1:** samp2tint({list 1}, {list 2}, Conf Lev)

**Description:** Finds the confidence interval for the difference between two population means when standard deviations of the two populations are unknown.

**Example:** samp2tint({20, 29, 30, 26, 21}, {28, 21, 27, 21, 20}, 0.99) = {-7.1645, 10.7645}

**Input 2:** samp2tint(mean 1, Sx<sub>1</sub>, n<sub>1</sub>, mean 2, Sx<sub>2</sub>, n<sub>2</sub>, Conf Lev)

**Description:** Finds the confidence interval for the difference between two population means when standard deviations of the two populations are unknown.

**Example:** samp2tint(25.2, 4.5497, 5, 23.4, 3.7815, 5, 0.99) = {-7.1644, 10.7644}

#### ▪ Two Sample Z Interval

**Input 1:** samp2zint(std dev 1, std dev 2, {list 1}, {list 2}, Conf Lev)

**Description:** Finds a confidence interval for the difference between two population means when the standard deviations of the two populations are known.

**Example:**

samp2zint(4.5497, 3.7815, {20, 29, 30, 26, 21}, {28, 21, 27, 21, 20}, 0.99) = {-5.0150, 8.6150}

**Input 2:** samp2zint(std dev 1, std dev 2, mean 1, n<sub>1</sub>, mean 2, n<sub>2</sub>, Conf Lev)

**Description:** Finds a confidence interval for the difference between two population means when both standard deviations are known.

**Example:** samp2zint(4.5497, 3.7815, 25.2, 5, 23.4, 5, 0.99) = {-5.0150, 8.6150}

#### ▪ t Interval

**Input 1:** tinterval({list}, Conf Lev)

**Description:** Finds a confidence interval for an unknown population mean,  $\mu$ , when the population standard deviation,  $\sigma$ , is also unknown.

**Example:** tinterval({20, 29, 30, 26, 21}, 0.99) = {15.8321, 34.5679}

**Input 2: `tinterval(x, Sx, n, Conf Lev)`**

**Description:** Finds a confidence interval for an unknown population mean,  $\mu$ , when the population standard deviation,  $s$ , is also unknown.

**Example:** `tinterval(25.2, 4.5497, 5, 0.99) = {15.8321, 34.5679}`

▪ **Z Interval**

**Input 1: `zinterval(std dev, {list}, Conf Lev)`**

**Description:** Finds a confidence level for an unknown population mean,  $\mu$ , when the population standard deviation,  $s$ , is known.

**Example:** `zinterval(3.7815, {28, 21, 27, 21, 20}, 0.99) = {19.0439, 27.7561}`

**Input 2: `zinterval(std dev, x, n, Conf Lev)`**

**Description:** Finds a confidence level for an unknown population mean,  $\mu$ , when the population standard deviation,  $s$ , is known.

**Example:** `zinterval(3.7815, 23.4, 5, 0.99) = {19.0439, 27.7561}`

**Tip**

- The calculated confidence interval will vary according to the confidence level that you input.

## Probability Distributions

*Portrait 4* calculates both probability density functions (pdf) and cumulative density functions (cdf) for several distributions. The probability density function is a function of a discrete or continuous random variable that determines the probability, or the chance that a specified value,  $x$ , will occur. The cumulative density function calculates the probability that the value of the random variable will fall within a range, specified by the upper and lower bounds.

To get to the probability functions tap the following: **Functions → Statistics → Probability**

▪ **Discrete Binomial Distribution**

**Input: `binompdf(# of trials, probability of success, x)`**

**Description:** Finds the probability at a given  $x$  for a binomial distribution with a specified number of trials and a specified probability of success on each trial. It counts the number of successes in a fixed number of independent events where each event has the same probability of success. For example, the number of defective items in a sample of  $n$  items can be represented by a binomial distribution.

**Example:** `binompdf(10, 0.5, 3) = 0.1172`

▪ **Cumulative Discrete Binomial Distribution**

**Input: `binomcdf(# of trials, probability of success, x)`**

**Description:** Finds a cumulative probability at a given  $x$  for a binomial distribution with a specified number of trials and a specified probability of success on each trial.

**Example:** `binomcdf(10, 0.5, 3) = 0.1719`

- **Chi-Square Distribution**  
**Input:** **chi2pdf(*x*, degrees of freedom)**  
**Description:** Finds the value of the chi-square probability distribution of a specific value of  $x$ . The chi-square distribution is used to measure goodness of fit, and to test hypotheses and obtain confidence intervals for the variance of a normally distributed random variable. The chi-square distribution is the distribution of the sum of squares of the independent, standard normal variables. The chi-square distribution is always positive, always skewed to the right and unimodal. The shape of the distribution depends on the degrees of freedom.  
**Example:**  $\text{chi2pdf}(5, 3) = 0.0732$
- **Cumulative Chi-Square Distribution**  
**Input:** **chi2cdf(lower bound, upper bound, degrees of freedom)**  
**Description:** Finds the area under the chi-square probability distribution between a lower bound and an upper bound where the degree of freedom is specified. The degree of freedom must be greater than 0.  
**Example:**  $\text{chi2cdf}(0.5, 1, 3) = 0.1176$
- **F Distribution**  
**Input:** **fpdf(*x*, numerator degrees of freedom, denominator degrees of freedom)**  
**Description:** Finds the value of the probability density function for the  $F$  distribution at a specified value of  $x$ . The  $F$  distribution is used when independent random samples are drawn from two normal populations with equal variances. It is a continuous distribution that is obtained from the ratio of two chi-square distributions, each of which is divided by the number of degrees of freedom. It is used to test the equality of variances of two normally distributed variables and the significance of possible causal variables in a regression.  
**Example:**  $\text{fpdf}(5, 10, 15) = 0.0027$
- **Cumulative F Distribution**  
**Input:** **fcdf(low bound, up bound, numerator deg of freedom, denominator deg of freedom)**  
**Description:** Finds the area under the  $F$  distribution probability between a lower bound and an upper bound for specified numerator degrees of freedom and denominator degrees of freedom.  
**Example:**  $\text{fcdf}(0.5, 1, 10, 15) = 0.3801$
- **Discrete Geometric Distribution**  
**Input:** **geompdf(probability of success, *x*)**  
**Description:** Finds a probability at a given  $x$ , which is the number of the trial on which the first success occurs, for a geometric distribution with a specified probability of success. In a series of independent trials, the geometric distribution determines the number of the trial on which the first success will occur. For example, a geometric distribution could determine the number of customers that must be contacted before the first sale (success) is made or the number of cars that must go through a radar check before the first speeder is detected.  
**Example:**  $\text{geompdf}(0.3, 10) = 0.0121$
- **Cumulative Discrete Geometric Distribution**  
**Input:** **geomcdf(probability of success, *x*)**  
**Description:** Finds a cumulative probability at the number of the trial on which the first success occurs,  $x$ , for a geometric distribution with a specified probability of success.  
**Example:**  $\text{geomcdf}(0.3, 10) = 0.9718$

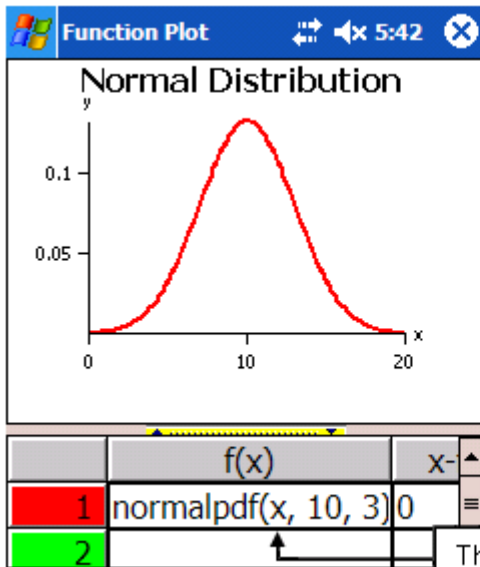
- **Inverse Cumulative Normal Distribution**  
**Input:** **invnorm(area, mean, standard deviation)**  
**Description:** Calculates the inverse cumulative normal distribution function for a given area under a normal distribution curve for a specified mean and standard deviation.  
**Example:**  $\text{invnorm}(0.5, 10, 3) = 10$
- **Normal Distribution**  
**Input:** **normalpdf(x, mean, std dev)**  
**Description:** Finds the value of the probability density function for the normal distribution at a specified value of  $x$ . The normal distribution is a distribution that is continuous and symmetric about the mean.  
**Example:**  $\text{normalpdf}(2, 10, 3) = 0.0038$
- **Cumulative Normal Distribution**  
**Input:** **normalcdf(lower bound, upper bound, mean, standard deviation)**  
**Description:** Calculates the inverse cumulative normal distribution function for a given area under a normal distribution curve for a specified mean and standard deviation.  
**Example:**  $\text{normalcdf}(0.5, 1, 10, 3) = 0.0006$
- **Discrete Poisson Distribution**  
**Input:** **poissonpdf(mean, x)**  
**Description:** Finds a probability for a specified  $x$ , for a Poisson distribution with a specific mean. The Poisson probability distribution provides a good approximation to binomial probabilities when the sample size is large and the population mean is small. In a Poisson distribution limiting arguments or parameters are applied to other distributions. The Poisson distribution could be used to determine distribution if you were restricted or limited by time, volume or area. For example, the Poisson distribution would be useful in modeling the number of flaws in a square meter of fabric or the number of bacteria in a limited sample of water.  
**Example:**  $\text{poissonpdf}(5, 10) = 0.0181$
- **Cumulative Discrete Poisson Distribution**  
**Input:** **poissoncdf(mean, x)**  
**Description:** Finds the cumulative probability at  $x$  for a discrete Poisson distribution with the specified mean.  
**Example:**  $\text{poissoncdf}(5, 10) = 0.9863$
- **Student  $t$ -Distribution**  
**Input:** **tpdf(x, degrees of freedom)**  
**Description:** Finds the value of the probability density function for the Student  $t$ -distribution at a specified value of  $x$ . The degrees of freedom must be greater than 0. The Student  $t$ -distribution is defined as the ratio of a normal distribution to the square root of the quotient of a chi-squared distribution, by its degrees of freedom. The  $t$ -distribution is a bell-shaped, symmetrical distribution with a mean of zero. It is flatter in shape than the normal distribution and the area under the tails is greater. It is named after William Sealy Gosset (1876 - 1937), the Englishman who first published the result under the pseudonym "Student".  
**Example:**  $\text{tpdf}(5, 3) = 0.0042$
- **Cumulative Student  $t$ -Distribution**  
**Input:** **tcdf(lower bound, upper bound, degrees of freedom)**  
**Description:** Finds the cumulative Student  $t$ -distribution probability between a lower bound and an upper bound for the specified degrees of freedom. The degrees of freedom must be greater than 0.  
**Example:**  $\text{tcdf}(0.5, 1, 3) = 0.1302$

## Graphing Probability Distributions

To graph any of the probability distributions open the Plot menu and select the Plot  $y = f(x)$  window. The type of information to be entered in the column under the header  $f(x)$  is probably best explained with the use of an example. Let's say that you want to graph a normal distribution. To calculate a probability value from the normal distribution you need the following information in the Input window:

`normalpdf(x, mean, standard deviation).`

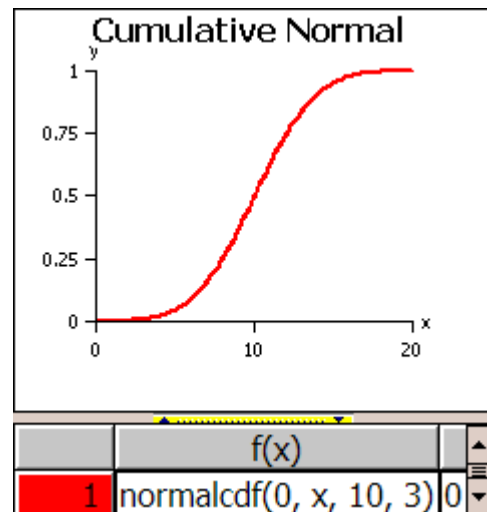
To calculate a probability value from the normal distribution, enter a value of 2 for  $x$ , a mean of 10 and a standard deviation of 3. For example, `normalpdf(2, 10, 3) = 0.003799`.



This is the same information that must be entered in the column under the header  $f(x)$  in the Plot  $y = f(x)$  window. Enter the function description; in this case it is `normalpdf( )`, and within the brackets enter the variable,  $x$ , followed by the values for the mean and the standard deviation. It is important that a domain,  $x$ -from and  $x$ -to, is also entered; otherwise a syntax error occurs. The graph to the left shows the normal distribution of  $x$  with a mean of 10 and a standard deviation of 3.

For all the probability distributions that contain **pdf**, the  $x$  variable is a part of the function. Some of the **cdf** distributions do not include a variable. To graph these distributions you must decide which value to make the  $x$  variable. Typically the upper bound is designated as the  $x$  variable. For example, for cumulative normal distributions make the upper bound the  $x$  variable and enter `normalcdf(0, x, 10, 3)`. See the graph to the right!

You can also plot many probability distributions, or pdf and cdf versions of distributions, on the same graph for comparison purposes. This can be useful to learn about the properties of the different probability distribution functions.



## Statistical Tests

Statistical tests are used to test the validity of hypotheses. Statistical hypotheses are statements regarding the value of population parameters. The hypothesis to be tested is called the null hypothesis and it represents the hypothesis or statement that you hope to reject. The null hypothesis is tested against an alternative hypothesis, the hypothesis or statement that you hope to support. The null hypothesis is a contradiction of the alternative hypothesis. The alternative hypothesis is defined as the statement we hope to show to be the case. It contains a  $<$ ,  $>$  or  $\neq$ . The three possible alternatives that can be entered are:

**Alternative 1:** If the alternative hypothesis is less than ( $<$ ), input -1.

**Alternative 2:** If the alternative hypothesis is greater than ( $>$ ), input 1.

**Alternative 3:** If the alternative hypothesis is not equal to ( $\neq$ ), input 0.

The statistical tests assume that the variances are to be pooled. If the variances are not pooled, input **false** after the alternative hypothesis parameter.

- **Chi-Square Test**

**Input:** `chi2test(matrix A)`

**Description:** Calculates a chi-square test for association on the row and column variables in the specified matrix. It tests the null hypothesis that no association exists between the row and column variables against the alternative that there is an association between the variables.

**Example:** `chi2test(matrix(2, 2, [1, 2, 3, 4])) = {chi2 = 0.079365, p = 0.77816}`

- **Linear Regression  $t$  Test**

**Input:** `linregttest({list 1}, {list 2}, alternate hypothesis)`

**Description:** Calculates a linear regression on the given data and a  $t$ -test on the value of the slope  $\beta$  and the correlation coefficient,  $p$ , for the equation  $y = \beta x + \alpha$ . It tests the null hypothesis that the slopes of the lines of the two sets of data are equal against one of three alternatives (specified by the **alt hyp** value). If the alternative hypothesis is that one slope is less than the other slope, input -1. If the alternative hypothesis is that one slope is greater than the other, input 1. If the alternative hypothesis is that one slope is not equal to the other, input 0.

**Example:** `linregttest({20, 29, 30, 26, 21}, {28, 21, 27, 21, 20}, 0) = {t = -0.085676, p = 0.937122, y = -0.041063*x + 24.434783, s = 4.361207, r = -0.049404}`

- **One Proportion  $Z$  Test**

**Input:** `prop1ztest(P, x, n, alternate hypothesis)`

**Description:** Calculates a test for an unknown proportion of successes. The input is the count of successes in the sample  $x$  and the count of observations in the sample  $n$ . It tests the null hypothesis that the expected proportion is equal to the actual proportion, against one of three alternatives (specified by the **alt hyp** value). If the alternative hypothesis is that the expected proportion is less than the actual proportion, input -1. If the alternative hypothesis is that the expected proportion is greater than the actual proportion, input 1. If the alternative hypothesis is that the expected proportion is not equal to the actual proportion, input 0.

**Example:** `prop1ztest(0.75, 3000, 5000, 0) = {z = -24.494897, p = 0, q = 0.6}`

- **Two Proportion  $Z$  Test**

**Input:** `prop2ztest(x1, n1, x2, n2, alternate hypothesis)`

**Description:** Calculates a test to compare the proportion of successes from two populations. The input is the count of successes and the count of observations in each sample. It tests the null hypothesis that the two proportions are equal against one of three alternatives (specified by the **alt hyp** value). If the alternative hypothesis is that one proportion is less than the other, input -1. If the alternative hypothesis is that one proportion is greater than the other, input 1. If the alternative hypothesis is that one proportion is not equal to the other, input 0.

**Example:** `prop2ztest(75, 100, 50, 100, 0) = {z = 3.651484, p = 0.000261, q1 = 0.75, q2 = 0.5, q = 0.625}`

- **Two Sample *F* Test**

**Input 1:** `samp2ftest({list 1}, {list 2}, alternate hypothesis)`

**Description:** Calculates an *f*-test to compare two normal population standard deviations where the population means and standard deviations are all unknown. The ratio of sample variances  $Sx_1^2/Sx_2^2$  is used to test the null hypothesis that the ratios are equal against one of the three alternatives (specified by the alt hyp value). If the alternative hypothesis is that one population standard deviation is less than the other, input -1. If the alternative hypothesis is that one population standard deviation is greater than the other, input 1. If the alternative hypothesis is that one population standard deviation is not equal to the other, input 0.

**Example:** `samp2ftest({21, 23, 26, 30, 17}, {24, 18, 14, 26, 23}, 1) = {F = 1.0125, p = 0.495342}`

**Input 2:** `samp2ftest(Sx1, n1, Sx2, n2, alternate hypothesis)`

**Description:** See description for Input 1.

**Example:** `samp2ftest(4.9294, 5, 4.8989, 5, 1) = {F = 1.012491, p = 0.495345}`

- **Two Sample *t* Test**

**Input 1:** `samp2ttest({list 1}, {list 2}, alternate hypothesis)`

**Description:** Calculates an *f*-test to compare two normal population standard deviations where the population means and standard deviations are all unknown. The ratio of sample variances  $Sx_1^2/Sx_2^2$  is used to test the null hypothesis that the ratios are equal against one of the three alternatives (specified by the alt hyp value). If the alternative hypothesis is that one population standard deviation is less than the other, input -1. If the alternative hypothesis is that one population standard deviation is greater than the other, input 1. If the alternative hypothesis is that one population standard deviation is not equal to the other, input 0.

**Example:** `samp2ttest({21, 23, 26, 30, 17}, {24, 18, 14, 26, 23}, 1) = {t = 0.772187, p = 0.231102}`

**Input 2:** `samp2ttest(mean 1, Sx1, n1, mean 2, Sx2, n2, alternate hypothesis)`

**Description:** Tests the null hypothesis that the means of two populations, based on independent samples, are equal when both population standard deviations are unknown, against one of three alternatives (specified by the alt hyp value). Option 1: If one population mean is less than the other, input -1. Option 2: If one population mean is greater than the other, input 1. Option 3: If one population mean is not equal to the other, input 0.

**Example:** `samp2ttest(23.4, 4.9294, 5, 21, 4.8989, 5, 1) = {t = 0.772202, p = 0.231098}`

- **Two Sample *Z* Test**

**Input 1:** `samp2ztest(standard dev 1, standard dev 2, {list 1},{list 2}, alternate hypothesis)`

**Description:** Tests the null hypothesis that the means of two populations, based on independent, are equal when both population standard deviations are known, against one of three alternatives (specified by the alt hyp value). Option 1: If one population mean is less than the other, input -1. Option 2: If one population mean is greater than the other, input 1. Option 3: If one population mean is not equal to the other, input 0.

**Example:** `samp2ztest(4.9294, 4.8989, {21, 23, 26, 30, 17}, {24, 18, 14, 26, 13}, 0) = {z = 1.415703, p = 0.156862}`

**Input 2:** `samp2ztest(stan dev 1, stan dev 2, mean 1, n1, mean 2, n2, alternate hypothesis)`

**Description:** See description for Input 1.

**Example:** `samp2ztest(4.9294, 4.8989, 23.4, 5, 19, 5, 0) = {z = 1.415703, p = 0.156862}`



- t-Test**  
**Input 1: `ttest((population mean, {list}, alternate hypothesis)`**  
**Description:** Tests the null hypothesis for one population mean with an unknown population standard deviation. The null hypothesis is tested against one of the alternatives (specified by the **alt hyp** value) that the actual population mean is less than (-1), greater than (1), or not equal to (0) the hypothesized population mean.  
**Example:** `ttest((25, {21, 23, 26, 30, 17}, 0) = {t = -0.725775, p = 0.508165}`
- Input 2: `ttest(population mean, mean, Sx, n, alternate hypothesis)`**  
**Description:** See description for Input 1.  
**Example:** `ttest(25, 23.4, 4.9294, 5, 0) = {t = -0.72579, p = 0.508157}`
- Z-Test**  
**Input 1: `ztest(population mean, standard dev, {list}, alternate hypothesis)`**  
**Description:** Tests the null hypothesis for one population mean with a known population standard deviation. The null hypothesis is tested against one of the alternatives (specified by the **alt hyp** value) that the actual population mean is less than (-1), greater than (1), or not equal to (0) the hypothesized population mean.  
**Example:** `ztest(25, 4.9294, {21, 23, 26, 30, 17}, 1) = {z = -0.72579, p = 0.766016}`
- Input 2: `ztest (population mean, standard dev, mean, n, alternate hypothesis)`**  
**Description:** See description for Input 1.  
**Example:** `ztest(25, 4.9294, 23.4, 5, 1) = {z = -0.72579, p = 0.766016}`

## Additional Statistical Functions

- One Way Analysis of Variance**  
**Input: `anova({list 1}, {list 2}, ..., {list k})`**  
**Description:** Finds a one-way analysis of variance for comparing the means of 2 to 20 populations. The null hypothesis that the population means are equal is tested against the alternative hypothesis that they are not equal.  
**Example:** `anova({3, 4, 2, 7}, {5, 6, 3, 8}, {5, 7, 9, 4}) = {F = 1.131737, p = 0.364389, {df = 2, SS = 10.5, MS = 5.25}, {df = 9, SSE = 41.75, MSE = 4.638889}}`
- Geometric Mean**  
**Input: `geomMean({x, y, z, ...})`**  
**Description:** Finds the geometric mean of a list of data. The geometric mean of a list of  $n$  numbers is the  $n^{\text{th}}$  root of the product of these numbers.  
**Example:** `geomMean({1, 5, 9, 7.2}) = 4.243`. The product of this list of numbers is 324 and the fourth root of 324 is approximately 4.243.
- Lower Quartile**  
**Input: `lowerQ({x, y, z, ...})`**  
**Description:** Finds the lower quartile of a list of data. The lower quartile is the median of the lower half of a set of data.  
**Example:** `lowerQ({5, 6, 7, 8, 9}) = 5.5`. The median of the lower half of this set of data is 5.5.

- **Maximum**  
**Input:**  $\text{max}(\{x, y, z, \dots\})$   
**Description:** Finds the maximum of a list of numbers. The maximum is the largest value in a set of data.  
**Example 1:**  $\text{max}(\{1, 2, -4, 5, 0\}) = 5$ . Five is the largest value in this set of data.  
**Example 2:**  $\text{max}(\text{primeSet}(12)) = 3$ . The largest number in the prime set of 12 is 3.
- **Arithmetic Mean**  
**Input:**  $\text{mean}(\{x, y, z, \dots\})$   
**Description:** Finds the arithmetic mean of a list of numbers. Also called the average, the mean is the sum of the values divided by the number of values used in computing the sum.  
**Example:**  $\text{mean}(\{1, 4, 0, 7, 7\}) = 3.8$ . The sum of the list is 19; the mean is  $19 \div 5 = 3.8$ .
- **Median**  
**Input:**  $\text{median}(\{x, y, z, \dots\})$   
**Description:** Finds the median of a list of data. For an odd number of data arranged in order the median is the middle number. For an even number of data arranged in order the median is the average of the two middle numbers.  
**Example:**  $\text{median}(\{1, 4, 0, 7, 7\}) = 4$ . When the data is arranged in order, the middle number is 4.
- **Minimum**  
**Input:**  $\text{min}(\{x, y, z, \dots\})$   
**Description:** Finds the minimum of a list of numbers. The minimum is the smallest value in a set of data.  
**Example 1:**  $\text{min}(\{1, 2, -4, 5, 0\}) = -4$ . Negative 4 is the smallest value in this list of data.  
**Example 2:**  $\text{min}(\text{primeSet}(12)) = 2$ . The smallest number in the prime set of 12 is 2.
- **Mode**  
**Input:**  $\text{mode}(\{x, y, z, \dots\})$   
**Description:** Finds the mode or modes of a list of data. The mode is the number or numbers that occurs most frequently in a set of data. If no number occurs more often than any other number, the set of data has no mode.  
**Example:**  $\text{mode}(\{1, 4, 0, 7, 7\}) = \{7\}$ . In this set of data 7 occurs most frequently. It is the mode.
- **Random Number**  
**Input:**  $\text{rand}(m, n)$   
**Description:** Finds a random integer  $k$  in the range  $m \leq k < n$  (or  $n \leq k < m$ , if  $n < m$ ).  
**Example:**  $\text{rand}(56, 12) = 49$ . The number forty-nine is a random integer between 56 and 12.
- **Population Standard Deviation**  
**Input:**  $\text{stdP}(\{x, y, z, \dots\})$   
**Description:** Finds the population standard deviation of a list of numbers. The standard deviation is equal to the square root of the variance.  
**Example:**  $\text{stdP}(\{1, 4, 0, 7, 7\}) = 2.93$ . The standard deviation is the square root of the variance.
- **Sample Standard Deviation**  
**Input:**  $\text{stdS}(\{x, y, z, \dots\})$   
**Description:** Finds the sample standard deviation of a list of numbers. The standard deviation is equal to the square root of the variance.  
**Example:**  $\text{stdS}(\{1, 4, 0, 7, 7\}) = 3.27$ . The standard deviation is the square root of the variance.

- **Upper Quartile**  
**Input:** **upperQ**({x, y, z, ...})  
**Description:** Finds the upper quartile of a list of data. The upper quartile is the median of the upper half of a set of data.  
**Example:** upperQ({5, 6, 7, 8, 9}) = 8.5. The median of the upper half of this set of data is 8.5.
- **Population Variance**  
**Input:** **varP**({x, y, z, ...})  
**Description:** Finds the population variance of a list of numbers. The variance is the mean value of the squares of the deviations from the mean of the data.  
**Example:** varP({1, 4, 0, 7, 7}) = 8.56. The variance of this set of data is 8.56.
- **Sample Variance**  
**Input:** **varS**({x, y, z, ...})  
**Description:** Finds the sample variance of a list of numbers. The variance is the mean value of the squares of the deviations from the mean of the data.  
**Example:** varS({1, 4, 0, 7, 7}) = 10.70. The variance of this set of data is 10.70.

## String

A string is a finite sequence of characters, like numerals, letters and symbols. The length of a string is determined by its number of characters. The length of a string can be zero or any positive integer. Strings are always contained in quotation marks, like the string “abcdef”. Substrings are any adjacent sequence of characters in a string.

To get to the string functions tap the following: **Functions → String**

- **Find First**  
**Input:** **findFirst**(string A, string B, position)  
**Description:** Finds the first occurrence after the indicated position (pos) of the string B in the string A or -1 if none exists.  
**Example:** findFirst(“abcabcdeb”, “b”, 1) = 2. The string ‘b’ first occurs after position 1, in position 2.
- **Find Last**  
**Input:** **findLast**(string A, string B, position)  
**Description:** Finds the last occurrence of the string B before the indicated position (pos) in the string A or -1 if none exists.  
**Example:** findLast(“abcabcdeb”, “b”, 6) = 5. The string ‘b’ last occurs before position 6, in position 5.
- **Length of a String**  
**Input:** **length**(string A)  
**Description:** Finds the length of a string.  
**Example:** length(“January”) = 7. The string “January” is 7 elements long.
- **Replace Substring**  
**Input:** **replace**(string A, string B, position, length)  
**Description:** Replaces the substring of A starting at an indicated position (pos) of indicated length (len) with B.  
**Example:** replace(“abcdef”, “xyz”, 1, 3) = “xyzdef”. The substring “abc” is replaced with “xyz”.

- **Find the Substring**  
**Input:** **substring(string A, position, length)**  
**Description:** Finds the substring of A starting at an indicated position (pos) of an indicated length (len).  
**Example:** `substring("abcdef", 3, 2) = "cd"`. The substring starting at position 3 that is 2 elements in length is "cd".

## Test

To get to the test functions tap the following: **Functions → Test**

- **Test a Condition**  
**Input:** **if(condition, x, y)**  
**Description:** Returns the value of x if the condition is true otherwise it returns the value of y.  
**Example:** `if(true, "Wednesday", "Friday") = Wednesday`. The value of b is true so the result is the value of x, Wednesday.
- **Select a Value**  
**Input:** **select(m, {list})**  
**Description:** Returns the value at position m of the list.  
**Example:** `select(10, {"a", "b", "c", "d", "e", "f", "g", "h", "i", "j"}) = "j"`. The letter in the tenth position of the list is "j".

## Trigonometric Functions

Trigonometric functions are defined in terms of an angle. The magnitude or size of an angle can be expressed in either radians or degrees. An angle that has a radian measure of 1 is equal to  $1/(2\pi)$  of a complete revolution. An angle that has a degree measure of 1 is equal to  $1/360$  of a complete revolution. All of the trigonometric functions are defined for complex numbers.

Remember, to find the inverse of a trigonometric function, the domain and/or range are restricted to obtain a one-to-one correspondence between input and output. This allows the graph of the function to pass the vertical line test. If a vertical line intersects a graph at most once, it is the graph of a function.

To get to the trigonometric functions tap the following: **Functions → Trigonometric**

- **Sine**  
**Input:** **sin(x)**  
**Description:** Denotes the trigonometric sine function. Sine is the ratio of the length of the side that is opposite the angle to the length of the hypotenuse.  
**Example:** `sin(pi/4) = 0.7071` (using Radians mode)
- **Cosine**  
**Input:** **cos(x)**  
**Description:** Denotes the trigonometric cosine function. Cosine is the ratio of the length of the side adjacent to the angle to the length of the hypotenuse.  
**Example:** `cos(pi/4) = 0.7071` (using Radians mode)

- **Tangent**  
**Input:**  $\tan(x)$   
**Description:** Denotes the trigonometric tangent function. Tangent is the ratio of the length of the side opposite the angle to the length of the side adjacent to the angle.  
**Example:**  $\tan(\pi/4) = 1.00$  (using Radians mode)
  
- **Cosecant**  
**Input:**  $\csc(x)$   
**Description:** Denotes the trigonometric cosecant function. Cosecant is the ratio of the hypotenuse to the length of the side opposite the angle.  
**Example:**  $\csc(\pi/4) = 1.4142$  (using Radians mode)
  
- **Secant**  
**Input:**  $\sec(x)$   
**Description:** Denotes the trigonometric secant function. Secant is the ratio of the hypotenuse to the length of the side adjacent to the angle.  
**Example:**  $\sec(\pi/4) = 1.4142$  (using Radians mode)
  
- **Cotangent**  
**Input:**  $\cot(x)$   
**Description:** Denotes the trigonometric cotangent function. Cotangent is the ratio of the length of the side adjacent to the angle to the length of the side opposite the angle.  
**Example:**  $\cot(1) = 0.64029$  (using Radians mode)

**Tip**

- For all of the trigonometric functions the default setting is radians. If your input is in degrees, select **Degrees** from the **Mode** section of the **Options** menu of your device.

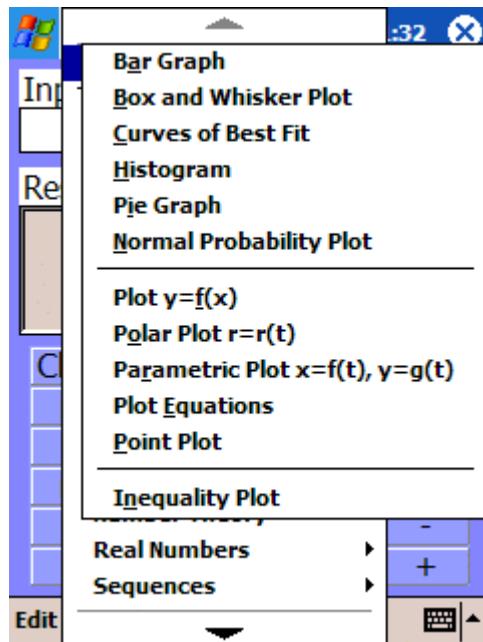
**Inverse Trigonometric Functions**

- **Inverse of Sine**  
**Input:**  $\arcsin(x)$  or  $\text{asin}(x)$   
**Description:** Denotes the inverse of the trigonometric sine function. The inverse of the sine function is defined by  $y = \arcsin x$  only if  $x = \sin y$  for  $-1 \leq x \leq 1$  and  $-\pi/2 \leq y \leq \pi/2$  radians.  
**Example 1:**  $\arcsin(1) = 1.5708$  radians  
**Example 2:**  $\arcsin(2) = 1.5708 - 1.3170*I$
  
- **Inverse of Cosine**  
**Input:**  $\arccos(x)$  or  $\text{arccos}(x)$   
**Description:** Denotes the inverse of the trigonometric cosine function. The inverse of the cosine function is defined by  $y = \arccos x$  only if  $x = \cos y$  for  $-1 \leq x \leq 1$  and  $0 \leq y \leq \pi$  radians.  
**Example:**  $\arccos(1) = 0$  radians
  
- **Inverse of Tangent**  
**Input:**  $\text{atan}(x)$  or  $\text{arctan}(x)$   
**Description:** Denotes the inverse of the trigonometric tangent function. The inverse of the tangent function is defined by  $y = \text{arctan } x$  only if  $x = \tan y$  for every  $x$  and  $-\pi/2 < y < \pi/2$  radians.  
**Example:**  $\text{arctan}(1) = 0.7854$  radians

- **Inverse of Cosecant**  
**Input:** **acsc(x)** or **arccsc(x)**  
**Description:** Denotes the inverse of the trigonometric cosecant function. The inverse of the cosecant function is defined by  $y = \text{arccsc } x$  only if  $x = \text{csc } y$  for  $|x| = 1$  and  $-\pi/2 \leq y \leq \pi/2$  radians,  $y \neq 0$ .  
**Example:**  $\text{arccsc}(1) = 1.5708$  radians
- **Inverse of Secant**  
**Input:** **asec(x)** or **arcsec(x)**  
**Description:** Denotes the inverse of the trigonometric secant function. The inverse of the secant function is defined by  $y = \text{arcsec } x$  only if  $x = \text{sec } y$  for  $|x| \geq 1$  and  $y$  in  $[0, \pi/2)$  or in  $(\pi/2, \pi]$ .  
**Example:**  $\text{arcsec}(1) = 0$  radians
- **Inverse of Cotangent**  
**Input:** **acot(x)** or **arccot(x)**  
**Description:** Denotes the inverse of the trigonometric cotangent function. The inverse of the cotangent function is defined by  $y = \text{arccot } x$  only if  $x = \text{cot } y$  for  $x \neq 0$  and  $-\pi/2 \leq y \leq \pi/2$  radians.  
**Example:**  $\text{arccot}(1) = 0.7854$  radians
- **Arctan2**  
**Input:** **arctan2(y, x)**  
**Description:** Finds the angle counterclockwise from the  $x$ -axis to the vector  $(x, y)$  position. It equals  $\text{arctan}(y/x)$  when  $x \neq 0$  and returns the appropriate angle measure if  $x = 0$ .  
**Example:**  $\text{arctan2}(1,1) = 0.7854$  radians
- **Convert Radians to Degree**  
**Input:** **rad2deg(angle)**  
**Description:** Converts an angle measure in radians to degrees.  
**Example:**  $\text{rad2deg}(1) = 57.29578$
- **Convert Degrees to Radians**  
**Input:** **deg2rad(angle)**  
**Description:** Converts an angle measure in degrees to radians.  
**Example:**  $\text{deg2rad}(57.29578) = 1$

## Plots Menu and Graphing

*Portrait 4* has twelve different graphing templates including: **bar graphs**, **box and whisker plots**, **best-fit or regression curve plots**, **histograms**, **pie graphs**, **normal probability plots**, **functions plots**, **polar plots**, **parametric plots**, **graphs of equations ( $f(x, y) = g(x, y)$ )**, **point plots**, and **graphs of inequality regions**. The plots are accessed from the Functions menu.



Selecting any plot from the menu opens a graphing template. Each of graphing templates allows you to input your own data and, at the top of a button, draw new graphs. All of the graphs have many features in common. The appearance of each graph can be altered in many ways using the menus available. The Plots are found at the top of the Functions menu when it is opened. Tap open Plots and select the graph of your choice. There is no limit (other than the memory limits of your device) to the number of functions, equations or the amount of data you can graph. The screen shot below provides an overview of a typical graphing window.

The graph of the data entered below is drawn in this space. Use the File, Edit and Tools menus to make changes to a graph.

Change the size of the graphing or the data entry window by sliding this horizontal bar up and down. Tap the up arrow to maximize the data entry window. Tap the down arrow to maximize the graph window.

Tap the pencil icon to draw a graph.

	$f(x)$	$x$ from
1		
2		
3		
4		

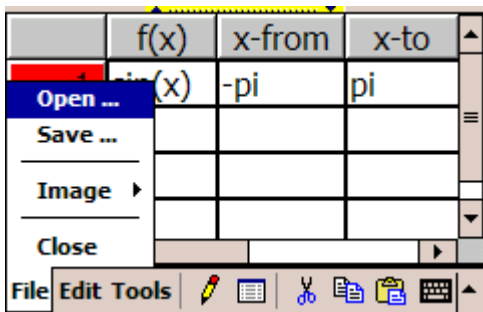
Enter the functions or other data that you want to graph in these cells.

Tap this icon to see a table of values. Keep in mind that the default setting for the number of points drawn for a graph is 100. If you do not change this setting, the table of values will also contain 100 points. Tap on the colored cell in front of the function to change properties.

File Edit Tools

## File Menu

The File menu enables you to open previously saved data or to save data. This menu also enables you to copy and/or export any graph images you create. You can also close the graphing window from this menu.



**Open** – Opens a plot. It recalls the data from a previously saved plot. When you tap **Open**, the **Open** window template is brought forward. Select your data by double tapping on your choice from the list of files. This restores the data. Tap on the pencil icon to redraw the graph.

**Save** – Saves the plot data. When the **Save** window opens, name your file and save it. You can save it using **All Files**. If you save it this way, tap on **All Files** to open it.

### Image:

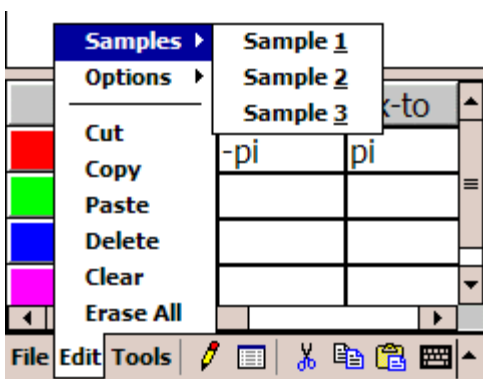
**Copy** – Graphs can be copied and pasted into a word processing document or graphical display software. Tap on copy, open up your word document and then paste the graph into the document.

**Export** – Allows for saving the graphical image as a bitmap file, which can be moved onto a desktop computer for printing, viewing, or other uses.

**Close** – This closes the plotting window. Any unsaved data will be lost.

## Edit Menu

The Edit menu provides access to the cut, copy, paste, delete, clear and erase all operations. The standard keyboard shortcuts, Ctrl x, Ctrl c and Ctrl v, can also be used for cut, copy and paste operations. The Edit menu also provides sample data and options for changing graph features.



**Samples** – Provides sample data to graph.

**Cut** – Cuts the selection in the data entry window and puts it on the clipboard.

**Copy** – Copies the selection in the data entry window and puts it on the clipboard.

**Paste** – Inserts clipboard contents into the data entry window at the selection point.

**Delete** – Deletes the selected data only. If no data is selected nothing is deleted. If all of the data is selected all the data is deleted. Delete rows by selecting the row header.

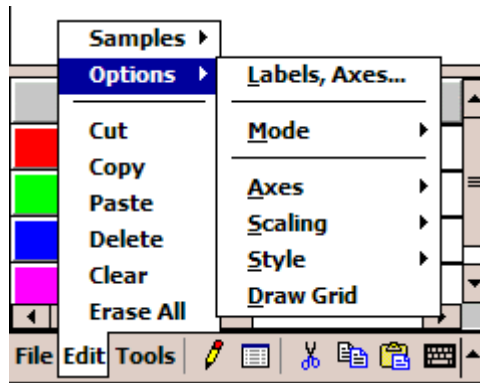
**Clear** – Erases the current data but leaves the graph.

**Erase All** – Removes all of the data as well as the graph.

### Options

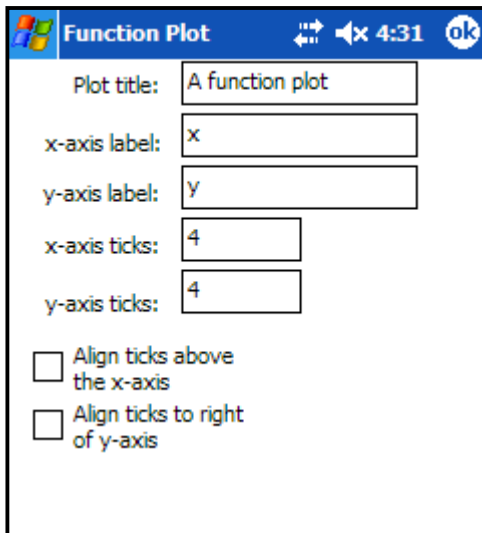
The Options menu lets you change the features of the graphs. The title, the number of tick marks on the axes, and the type of axes used can all be altered using the choices in **Options**. Not all options are available for all graphs.





### Options - Labels

Tap on Labels to open the Plot Labels window. In this window you edit the labels on the graph. To enter new information, delete the default labels and enter the new ones. Once changes have been made, tap OK and the changes are automatically applied to your graph.



**Plot title:** Enter a title. The default title for each graph is the name of the type of graph.

**x-axis label:** Label the  $x$ -axis. The default label is  $x$ .

**y-axis label:** Label the  $y$ -axis. The default label is  $y$ .

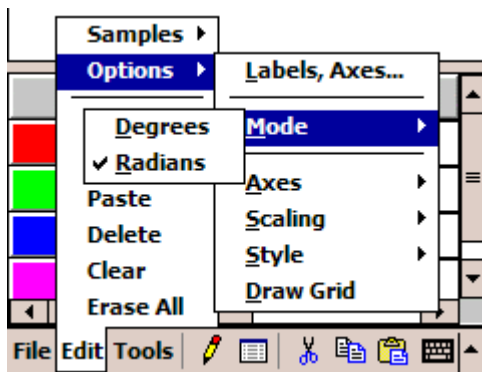
**x-axis ticks:** Put tick marks on the  $x$ -axis. The default is 4.

**y-axis ticks:** Put tick marks on the  $y$ -axis. The default is 4.

**Align ticks above the  $x$ -axis:** Place tick marks and values above the  $x$ -axis.

**Align ticks to the right of the  $y$ -axis:** Place tick marks and values to the right of the  $y$ -axis.

### Options - Mode

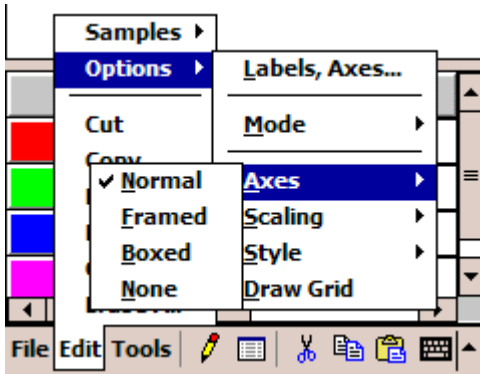


**Degrees** – Interprets or reports angle measures in degrees.

**Radians** – Interprets or reports angle measures in radians.

A check mark indicates your selection.

## Options - Axes



**Normal** – The  $x$ - and  $y$ -axes intersect at the origin. The normal view of a box plot is a number line.

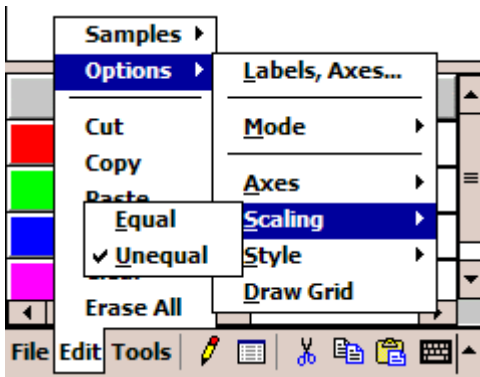
**Framed** – The  $x$ - and  $y$ -axes intersect at the lowest  $x$ - and  $y$ -coordinates.

**Boxed** – Places a box around the framed view of the graph.

**None** – Removes all of the axes.

- A check mark indicates your selection.
- Pie graphs and histograms do not have these options.

## Options - Scaling

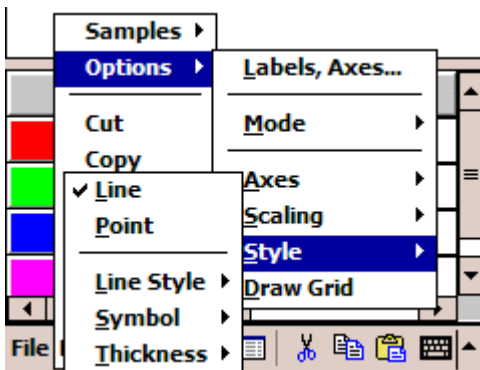


**Equal** – Uses the same scale on the  $x$ - and  $y$ -axes, preserving the true proportions of the graph.

**Unequal** – Assigns different scales to the  $x$ - and  $y$ -axes, so the graph fills the available space.

- A check mark indicates your selection.
- Box and whisker plots, pie graphs and bar graphs do not have a scaling option.

## Options - Style



**Line** – Draws the graph with a line.

**Point** – Draws the graph with points.

**Line Style** – Choose between **Dashed** and **Solid** lines.

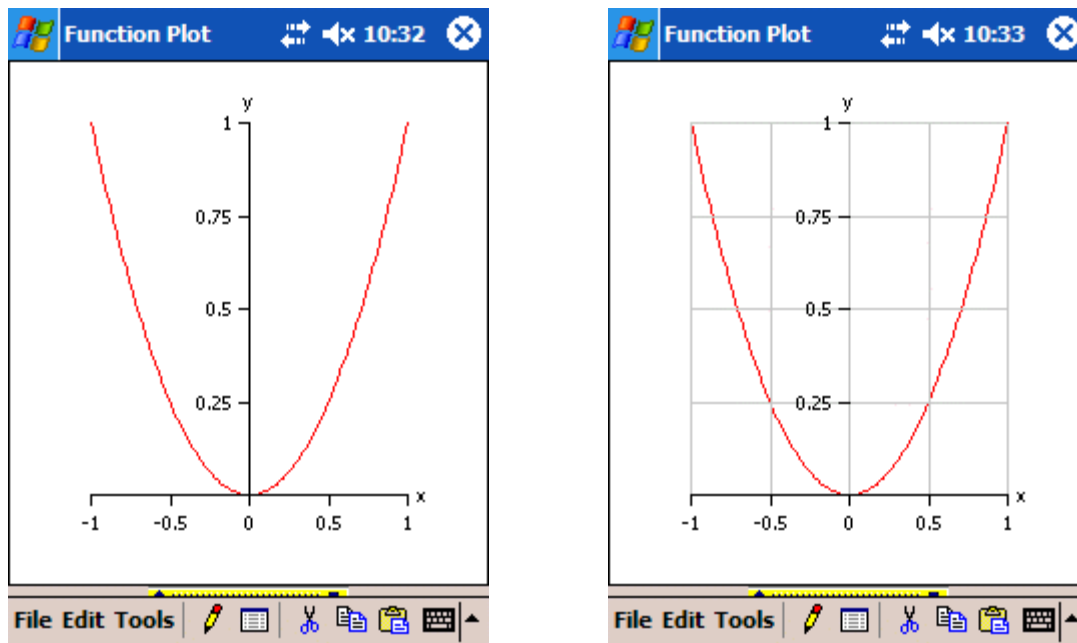
**Thickness** – Choose between **Thin**, **Medium**, and **Thick**.

**Symbol** – Choose between **Circle**, **Cross**, **Diamond**, **Point**, and **Square**.

- Pie graphs and bar graphs do not have these style options.
- Graphs of inequalities can only have the line thickness altered.
- Point plots have an additional symbol type (**None**), which indicates that the points themselves are not to be plotted. This can be useful if you set the **Style** to **Line** instead of point.
- The **Style** menu for histograms is very different from that of the other graphs. It includes being able to show the histogram with **Cumulative Frequencies** and/or a **Frequency Polygon**.
- If you choose the dashed line option and then change the thickness of the dashed line to either medium or thick, the line no longer appears to be dashed.

## Options – Draw Grid

This option adds a graph paper background to a graph when selected.



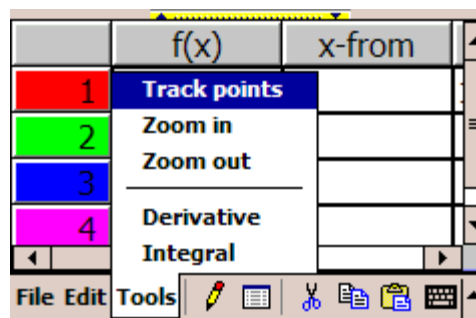
**Insert Column** – Adds a column so data for another graph on the same set of axes can be entered and plotted. This option is on bar graphs, box and whisker plots, curves of best fit, normal probability plots and point plots.


**Delete Column** – Removes a column that has been added. If several columns have been added, highlight the column you want to remove and tap on **Delete Columns**. If a column is not highlighted, the last column added is deleted. This command will not remove the original column. This option is on bar graphs, box and whisker plots, curves of best fit, normal probability plots and point plots.

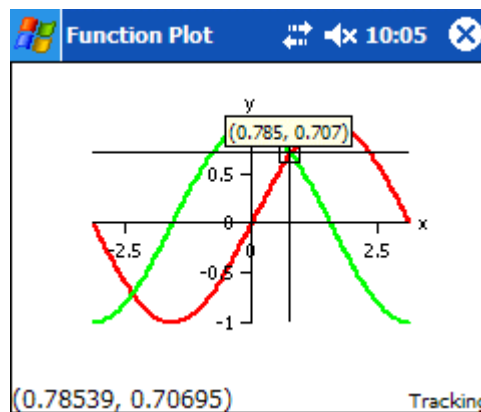
## Tools Menu

The **TOOLS** menu allows you to gather more specific information about the graphs. With these tools you can find specific points on a graph, as well as zoom in and out on a particular portion of a graph. These tools are not available for all graphs.

If a tool is in use it is shown in the bottom right corner of the graphing window. To switch to another tool, select it from the **TOOLS** menu. To turn off a tool, select it again.



**Track points** – This tool provides a  and crosshairs that can be placed anywhere on the graph. It shows the specific  $x$ - and/or  $y$ -coordinates, in the bottom left corner of the frame, as it is moved along the graph. The values are shown to five decimal places. If this is too many, round up or down appropriately. The **Track points** tool is activated only if there is a plot in the graphing window.



**Snap to Intersections/Snap to Intercepts** – If you have two or more functions that intersect, you can find the precise coordinates of the intersections. When the tracking crosshairs are close to an intersection, they “snap” to the intersection. A small box pops up near the intersection, displaying the coordinates in a shortened form. More precise values are displayed in the lower left corner of the graph window. The tracking tool will also “snap” to the  $x$ - and  $y$ -intercepts on a graph.

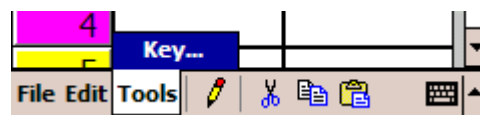
**Zoom in** – When **Zoom in** is selected place your stylus on the screen and draw a box around the portion of the graph that you want to zoom in on. When you remove your stylus, it automatically zooms in on the selected area. To return to the original view, tap the pencil icon. The **Zoom in** tool can be activated only if there is a plot in the graphing window.

**Zoom out** – When **Zoom out** is selected each tap on the screen returns you to a previously zoomed view of the graph. To return to the original view, tap the pencil icon. The **Zoom out** tool can be activated only if there is a plot in the graphing window.

**Derivative** – This tool is available for the function plots (Plot  $y = f(x)$ ) only. With **Derivative** selected, tap on a point on the graph to see the tangent line to the function at that point. The equation of the derivative is displayed in the bottom left corner of the graphing window.

**Integral** – This tool is available for the function plots (Plot  $y = f(x)$ ) only. With **Integral** selected, tap on a point on the graph and drag the stylus to a second point. The integral of the function between these two points is displayed in the bottom left corner of the graphing window.

**Key** – Provides pertinent information about the graph. Not all graphs have a key.



## Changing Colors, Points and Other Properties

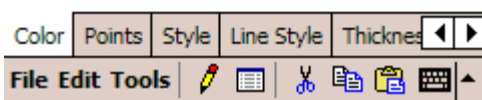
A double-tap on a colored cell in a column or row header in a data entry window opens a **Properties** window.

The **Properties** window allows you to make changes to a specific graph. For example, if two functions are plotted on the same set of axes, opening the **Properties** of one of the functions changes the properties of the graph for that function only. The properties of the other function remain unchanged.

	f(x)	x-from	x-to
1	sin(x)	-pi	pi
2	cos(x)	-pi	pi
3			

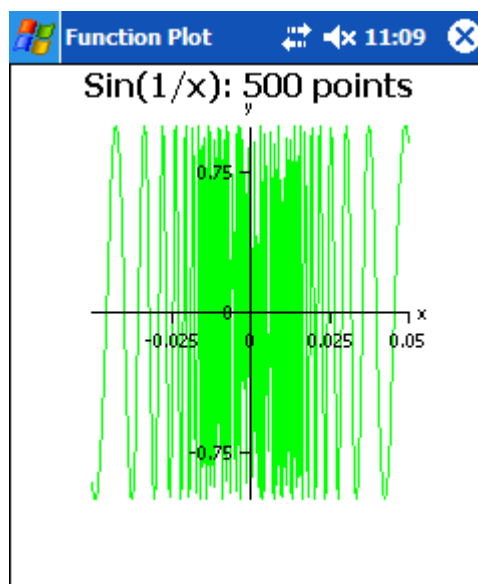
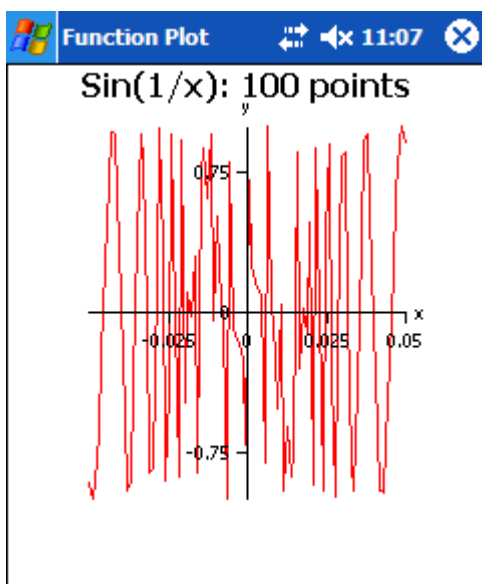


To change the **color** of a graph or portion of a graph, select the color you want the graph to be and tap OK. The graph appears in the new color after you tap on the pencil icon. If you want to change the color of the graph after it has been drawn, change the color the same way. Always tap the pencil icon.



The number of **Points** plotted can be changed in many of the graphing windows. In some cases it may be helpful to change the number of points plotted, as **the default value of 100 points** may not provide a true representation of the function. For example, open the Plot  $y=f(x)$  graphing window and enter the function  $\sin(1/x)$ . Enter a domain from  $-0.05$  to  $0.05$  and tap the pencil icon. Enter this information again in the next row but this time open the Properties window. Change the color of the graph and change the number of points plotted to 500. Once you have made your changes tap OK and tap on the pencil icon to see a much more accurate representation of  $\sin(1/x)$ .

Remember, the number of points plotted also has an impact on the table of values. Each point plotted is represented in the table of values. If your plot 500 points the table of values will provide a value for each of the 500 points.

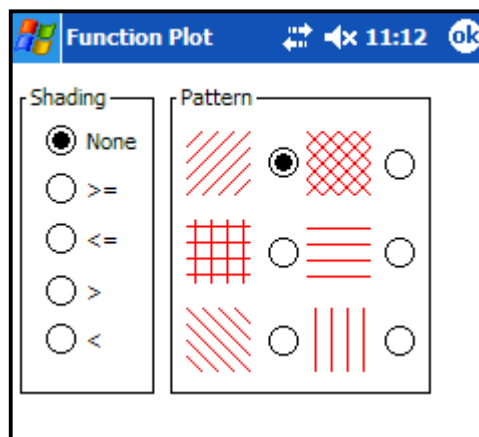


The **Style** features can also be changed from the Properties window. The default settings for the style features are the same as the Edit menu on the graph.

- To change the **Line Style** of a graph open the Properties box; select either Solid or Dashed.
- To change the **Thickness** of a graph open the Properties box; select Thin, Medium or Thick.
- To change the **Symbol** of a graph open the Properties box; select Circle, Cross, Diamond, Point, or Square.
- After a choice is made, **always** tap on the pencil icon to redraw the graph with the new selections.







The Properties window for the Curves of Best Fit graphing window has an additional feature called **Curves**. Use this property to indicate the type of curve you want to fit to your data. For more information see the section on Curves of Best Fit on page 71.

The Properties window for the function plots (Plot  $y = f(x)$ ) graphing window has an additional feature called **Shading**. Use this property to shade portions of the graph that are less than, less than or equal to, greater than or greater than or equal to the function. The feature also lets you select from six different patterns of shading.



## Other Graphing Icons

There are icons on the task bar of each graphing window. The icons to the right of the File, Edit, and Tools menus perform the following functions:

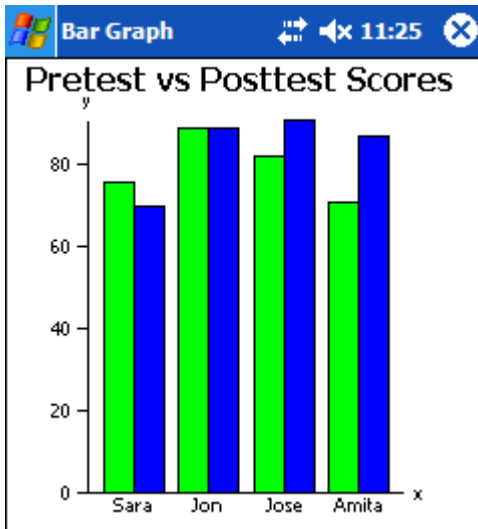
-  – Draw the plot when new data is entered or any of the options have been changed.
-  - Tap on this button to send the information in the data entry window to the Table Editor to see a table of values. To return to the data entry window tap on this button again. Remember, the number of points plotted has an impact on the table of values. Each point plotted is represented in the table of values. To change the number of points plotted, open Properties window by double-tapping on a colored cell in a column or row header in a data entry window.
-  - Tap on the scissors to cut the highlighted selection and put it on the clipboard.
-  - Tap on the double pages to copy the highlighted selection and put it on the clipboard.
-  - Tap on the clipboard to insert or paste text from the clipboard.
-  - Tap on this icon to open the keyboard. To close the keyboard tap on the icon again.

## Helpful Hints for Graphing

The following tips apply to all of the graphing templates. It will be helpful to keep them in mind as you begin to explore each of the graphing templates.

- If you make a mistake entering data, tap on a cell so that it is selected and delete or re-enter the data properly.
- If you enter inappropriate data or a nonsensical operation and plot it, an error message appears.
- If you require additional rows for data, tap in a cell that accepts data and tap Enter. An empty row appears below the row the selected cell is in.
- If you want to graph more than one set of data, go to the Edit menu. Tap Insert Columns.
- To change the width of columns in the data entry window place the cursor at the edge of the gray cell at the top of the column. Drag the edge of the cell left or right to make the column smaller or larger.

## Bar Graphs



A bar graph is a graph in which data values are shown by means of the length of rectangular bars. The graph to the left uses a bar graph to compare the scores from a pretest and a posttest taken by four different students.

### Entering Data

1. Tap in a cell in a column labeled Data.
2. Enter your data in the cell.
3. Enter a label for your data. This is optional but proper labels make a graph more informative.
4. Move to the next empty cell. Enter the next piece of data. Continue until all of your data is entered.
5. From **Graph type** in the **Options** menu (under **Edit**), tap on the type of bar graph you want.
6. Tap on the pencil icon to graph the data.

	Label	Data	Data
1	Sara	76	70
2	Jon	89	89
3	Jose	82	91
4	Amita	71	87
5			

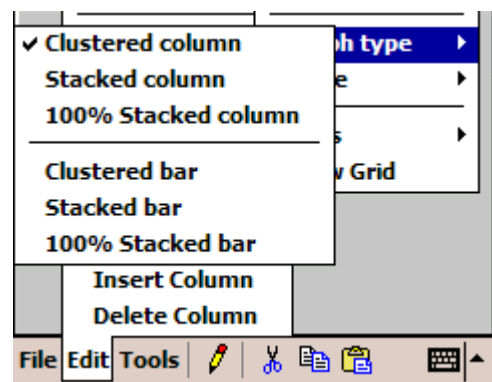
### Note

- The data must be an expression whose result is a positive or negative whole number or decimal.
- Zero can be input, however, a bar representing zero may not appear on the graph if your **Axes** are set to **Normal**.
- Unassigned variables cannot be represented on a bar graph.

### Bar Graph Types

When you are graphing two or more sets of data you can select the type of bar graph that you want to use to compare the sets of data. The bar graph window has six choices available. They are found in the **Edit** menu, in **Options** under **Graph type**.

**Clustered Column** – A clustered column groups or clusters comparisons among individuals or categories. For example, in the bar graph above, the pretest and posttest scores for each student are clustered together. It is easy to see if a student's score improved from the pretest to the posttest.



**Stacked Column** – Stacked bar graphs show the relationship of individual items or categories to the whole. The heights of the individual columns vary accordingly.

**100% Stacked Column** – These stacked columns show the relationship of individual items or categories to 100%. The heights of the columns are all the same. They all go to 100%. Each item in a column represents a portion of the 100%. This type of graph is only appropriate for positive data.

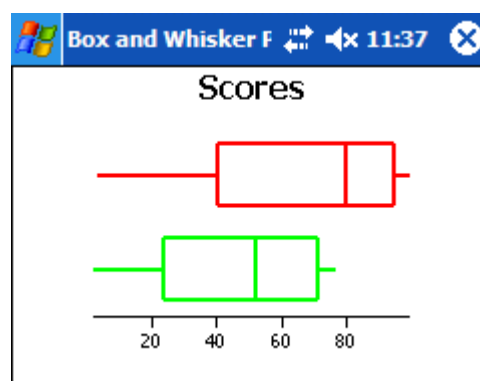
**Clustered Bar** – Like the clustered column a clustered bar groups or clusters comparisons among individuals or categories. Categories are organized vertically and values horizontally.

**Stacked Bar** – Similar to the stacked column except that the categories are organized vertically and values horizontally.

**100% Stacked Bar** – Similar to the 100% stacked column except that the categories are organized vertically and values horizontally. This type of graph is only appropriate for positive data.

## Box and Whisker Plots

A box and whisker plot is a graphic way of showing a summary of data using the median, quartiles and extremes of the data. The whiskers, or lines, on the graph represent the range of the data, the box represents the quartiles, and the line in the box represents the median.



## Entering Data

	Data	Data
1	100	77
2	3	66
3	80	45
4	78	52
5	90	2

1. Tap on the first cell in the column labeled Data.
2. Enter your first piece of data in the first row of the column.
3. Move to the second row in the column and enter your data.
4. Continue entering data in this way so that each row in the column contains a separate piece of data.
5. Once your data is entered, tap on the pencil icon to draw a graph of the data.

### Note

- Data may be entered in the form of integers, decimal numbers, symbolic constants and/or calculations, or other numeric expressions.

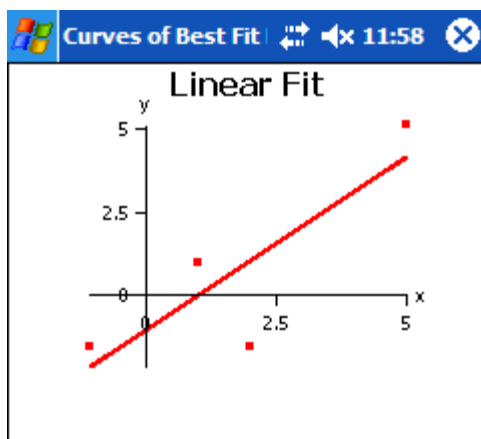
## Additional Information

Additional information about the box and whisker plots is available by opening the **TOOLS** menu and selecting **Key**. This opens a key window that will give you the minimum value (**Min.**) and maximum value (**Max.**) of the data as well as the upper quartile (**U.Q.**), the lower quartile (**L.Q.**) and the median (**Med.**). If you have input incorrect data and tried to graph it, you will get a message saying **empty plot** in this area. The information in this key can be copied and pasted into a word processing document by using the **Ctrl C** (copy) and **Ctrl V** (paste) shortcuts on your keyboard. Highlight the information you want to copy and use **Ctrl C** to copy it and then **Ctrl V** to paste it in the location you want. The column headers do not copy and paste.



## Curves of Best Fit

Curves of best fit plots a set of points and draws a curve that comes closest to fitting that set of points.



### Entering Data

x-from	x-to	y-from	y-to
	x	y	
1	-5	5	
2	-4	4	
3	-3	3	
4	-2	2	
5	0	0	

1. Tap in one of the spaces in the column labeled x or y.
2. Enter your x and y values Row 1 is the first point, row 2 is the second point, etc. If you have more than one set of data, add more columns by opening the Edit menu and selecting Insert Columns.
3. Enter the x-from, x-to, (domain) y-from and y-to (range) values that you want your data plotted within. This step is optional but you will build a better graph if you include appropriate limits.
4. Once your x and y values (and the limits) have been entered, tap on the pencil icon. This opens the Curves property window.
5. Choose the type of curve that you want to fit to the data. To fit an  $n$ -degree polynomial you must also enter the degree in the box provided.
6. Tap ok and the graph will be drawn.

### Note

- The x and y values may be integers, decimal numbers, symbolic constants or calculations.
- If you select No Regression in the Curves section of the Properties window, the points are graphed without a best-fit line.
- Entering the domain and range to plot the data within is optional but you will build a better graph if you include this. If you do enter this information it has to be entered for both x and y.

## Types of Curves

- **No Regression** – Select **no regression**. A curve will not be fit to the data.
- **Linear Regression** – Select **linear**. Fits the equation  $y = ax + b$  to the data using a least-squares fit.
- **Quadratic Regression** – Select **quadratic**. Fits the equation  $y = ax^2 + bx + c$  to the data using a least-squares fit. This equation is a second-degree polynomial.
- **Cubic Regression** – Select **cubic**. Fits the equation  $y = ax^3 + bx^2 + cx + d$  to the data using a least-squares fit. This equation is a third degree polynomial.
- **Quartic Regression** – Select **quartic**. Fits the equation  $y = ax^4 + bx^3 + cx^2 + dx + e$  to the data using a least-squares fit. This equation is a fourth-degree polynomial.
- **Median line** - Select **median-median**. Fits the equation  $y = ax + b$  to the data using the resistant line technique.
- **Exponential Regression** – Select **exponential**. Fits the equation  $y = a*b^x$  by transforming the data to  $x$  and  $\ln(y)$  and then fits a straight line to the transformed data using a least-squares fit. The  $y$ -values in your data must be positive or you will get an error message stating it was unable to fit an exponential regression to the data.
- **Logarithmic Regression** – Select **logarithmic**. Fits the equation  $y = a*b^x$  by transforming the data to  $\ln(x)$  and  $y$  and then fits a straight line to the transformed data using a least-squares fit. The  $x$ -values in your data must be positive or you will get an error message stating it was unable to fit a logarithmic regression to the data.
- **Power Regression** – Select **power**. Fits the equation  $y = a*b^x$  by transforming the data to  $\ln(x)$  and  $\ln(y)$  and then fits a straight line to the transformed data using a least-squares fit. The  $x$ - and  $y$ -values in your data must be positive or you will get an error message, *unable to fit a power regression to the data*.
- **Logistic Regression** – Select **logistic**. Fits the equation  $y = c/(1 + e^{(a-bx)})$  to the data using an iterative least-squares fit.
- **Sinusoidal Regression** – Select **sinusoidal**. Fits the equation  $y = a \cos(wx) + b \sin(wx) + c$  to the data using an iterative least-squares fit. You should have at least four points in your data set to fit a sinusoidal curve with at least two points per cycle.
- **Polynomials of degree  $n$**  – Select **polynomial (of degree)** then enter the number of the degree of the polynomial you want. It fits the appropriate  $n^{\text{th}}$ -degree polynomial equation to the data. For example, 6 will fit a polynomial of degree 6. Only a portion of the equations of the curves for these large degree polynomials will be visible in the key that displays the equations. You can tap in the equation cell and use the cursor keys to view the rest of the equation.

## Additional Information

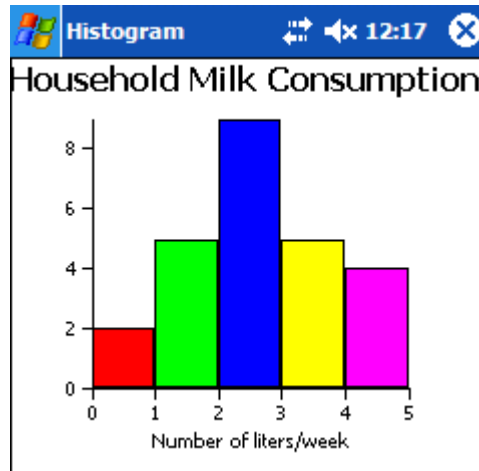
Additional information about the curves of best fit is available by opening the **TOOLS** menu and selecting **Key**. This key will give you the **equations** of the lines you have graphed. It will also give you the **mean squared error, MSE**, and the **correlation coefficient,  $r$** , for the transformed data (certain fit types may not provide this information). If the data you have entered does not lend itself to the type of curve you want to fit to it you will get an error message, *the curve cannot be fit to the data*.

Curves of Best Fit P	
Description	
1	$y = 1.026*x + -0.994, \text{MSE} = 4.3$

The information in this key can be copied and pasted into a word processing document (or Plot Window, like the Function Plot) by using the **Ctrl C** (copy) and **Ctrl V** (paste) shortcuts on your keyboard. Highlight the information you want to copy. Use **Ctrl C** to copy it and then **Ctrl V** to paste it in the location you want.

# Histograms

A histogram is a graph that is similar to a bar graph except that it indicates the frequency distribution of data. Also, there are no spaces between the bars. It is used to display continuous data that can be organized into intervals. The intervals are placed on the horizontal axis and the frequencies are on the vertical axis. The height of a bar or rectangle that is drawn corresponds to the frequency of data within that interval.



## Entering Data

# of classes: 5		
	Data	Frequency
1	0	2
2	1	5
3	2	9
4	3	5
5	4	3
6	5	1

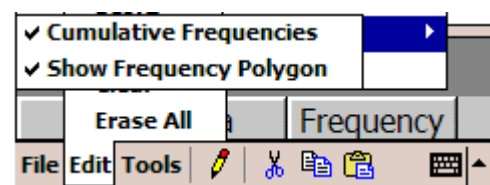
1. Tap on one of the cells in a column labeled Data.
2. Enter your data in the cells available.
3. Tap in the Frequency column and record the number of times that piece of data occurs in the set of data. The data in row 3 indicates that 2 occurs in the entire set of data 9 times. The frequency is 9.
4. Continue entering the data and frequencies in this manner until all of your data has been entered.
5. Determine the number of classes or intervals that you want the data divided into. Enter this number in the cell beside # of classes. It is best to have from 5 to 20 classes. The larger the amount of data, the more intervals can be used.
6. Tap on the pencil icon to draw the graph.

## Note

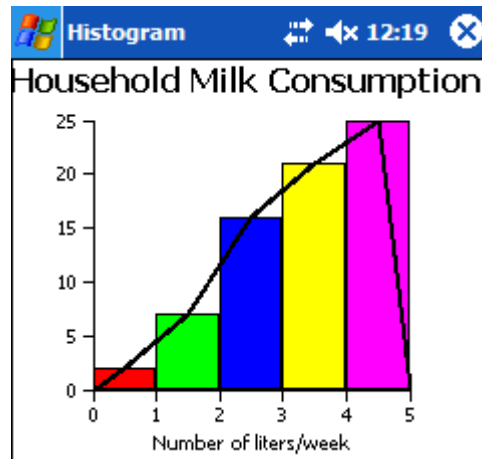
- Each piece of data must be an expression whose result is a positive or negative whole number or a positive or negative decimal number.
- Unassigned variables cannot be represented on a histogram.
- The number of classes must be a positive integer.
- Frequencies must be positive integers.

## Style Option

The Style menu for histograms is very different from that of the other graphs. It includes being able to show the histogram with Cumulative Frequencies and/or a Frequency Polygon.



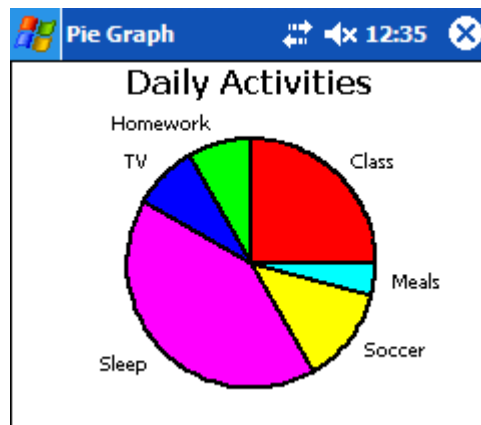
**Cumulative Frequencies** – A graph displaying the cumulative frequencies of the data takes each interval after the first and adds the frequencies of the intervals together. The second interval is now an accumulation of the frequencies of the first and second interval. The third interval is an accumulation of the frequencies of the first, second and third intervals. In the histogram to the right the first interval shows that 2 households consume 0 to 1 liters of milk per week. The second interval adds the 2 households from the first interval to the 5 households that drink 1 to 2 liters of milk per week to get a cumulative frequency of 7, which is displayed in the second interval. This continues until all 25 households in the survey are accounted for in the final interval.



**Show Frequency Polygon** – The frequency polygon draws a line graph over the bars of the histogram, connecting the midpoints of each bar. This histogram also shows the frequency polygon related to the data. This may be useful for emphasizing trends in the data.

## Pie Graphs

A pie or circle graph is a graph in which parts or sections of a circle represent data. The area of each section of the circle is drawn so that it represents the percentage with which a piece of data occurs relative to the whole.



## Entering Data

1. Tap on one of the spaces in the column labeled Data.
2. Enter each piece of data.
3. Move to the second row in the column. Enter the next piece of data. Continue entering data in this way so that each row in the column contains a piece of data.
4. Type in labels for your data. This is optional, but labels help to produce a more informative graph.
5. Tap on the pencil icon to draw the graph.

	Label	Data
1	Class	6
2	Homework	2
3	TV	2
4	Sleep	10
5	Soccer	3
6	Meals	1

## Note

- Each piece of data must be an expression that results in a positive whole number or a positive decimal number.
- Negative numbers, zero and unassigned variables cannot be represented on a pie graph.

## Additional Information

	Label	Percentages
1	Class	25
2	Homework	8.33
3	TV	8.33
4	Sleep	41.67
5	Soccer	12.5
6	Meals	4.17

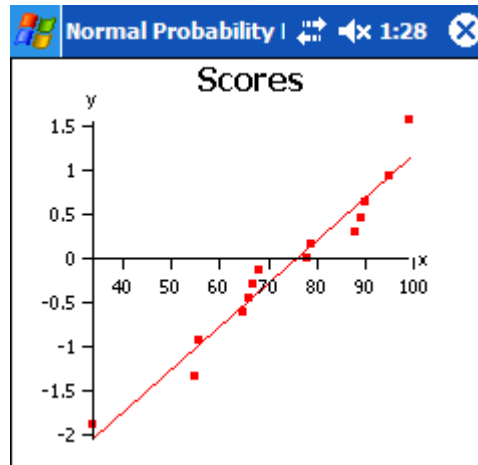
Additional information about the pie graph is available by opening the **Tools** menu and selecting **Key**. Tapping on this opens a window that will give you the percentages that each section of the pie represents. If you have input incorrect data in the data boxes and try to graph it, you get a message saying **empty plot** in this area.

The information in this key can be copied and pasted into a word processing document by using the **Ctrl C** (copy) and **Ctrl V** (paste) shortcuts on your keyboard. Highlight the information

you want to copy and use **Ctrl C** to copy it and then **Ctrl V** to paste it in the location you want. The column headers do not copy and paste.

## Normal Probability Plot

A normal probability plot assesses whether or not a data set is normally distributed. When plotted against a theoretical normal distribution the points should form a straight line. Points not on this straight line indicate a deviation from the normal distribution.



## Entering Data

	Data
1	68
2	89
3	95
4	99
5	67
6	56
7	70

1. Tap in a cell in the column labeled **Data**.
2. Enter your data set. Row 1 is the first point, row 2 is the second point, and these points represent the  $x$ -coordinate. If you have more than one set of data, add more columns by opening the **Edit** menu and selecting **Insert Columns**.
3. Once the data set has been entered, tap the pencil icon. The  $z$  values for each point are automatically calculated and these represent the  $y$ -coordinates.
4. The graph will be drawn.

## Additional Information

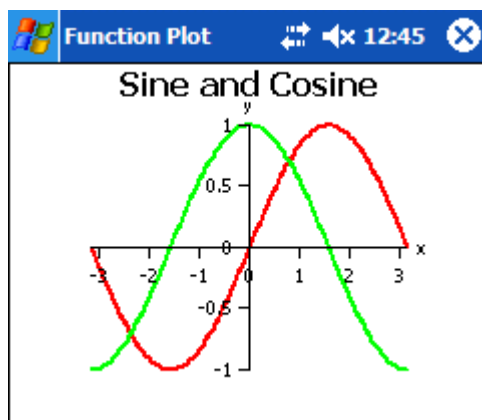
Additional information about the normal probability plot is available by opening the **TOOLS** menu and selecting **Key**. This key will give you the **equation** of the line you have graphed. It will also give you the **mean squared error, MSE**, and the **correlation coefficient,  $r$** , for the data.

	Description	MSE	$r$
1	$y = 0.051*x + -3.81$	0.047	0.976

The information in this key can be copied and pasted into a word processing document (or Plot Window, like the Function Plot) by using the **Ctrl C** (copy) and **Ctrl V** (paste) shortcuts on your keyboard. Highlight the information you want to copy. Use **Ctrl C** to copy it and then **Ctrl V** to paste it in the location you want.

## Graphs of Functions

The graph of a function is a graph of the equation  $y = f(x)$  where  $f$  represents any function and the value of  $y$  is dependent on the value of  $x$ . The graph is the set of all points where the  $x$ - and  $y$ -coordinates are  $(x, f(x))$ .



## Entering Data

1. Tap in one of the cells in the column labeled  $f(x)$ .
2. Enter your function. Use “^” for exponents and “\*” for multiplication.
3. Enter the domain ( $x$ -from and  $x$ -to) that you want your equation evaluated from and to. Use the **Tab** key to move over to these spaces or tap in them with your stylus. Entering the  $y$  limits is optional but you will build a better graph if appropriate  $y$ -from and  $y$ -to values are included.
4. Tap on the pencil icon to draw the graph of the equation.

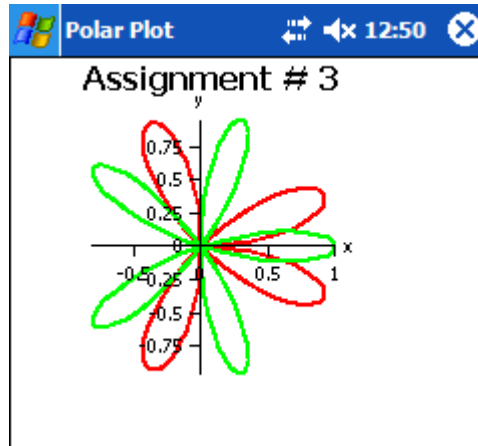
	$f(x)$	$x$ -from	$x$ -to
1	$\sin(x)$	$-\pi$	$\pi$
2	$\cos(x)$	$-\pi$	$\pi$
3			

### Note

- If you do not enter  $x$  limits you will get a syntax error when you try to plot the function.
- Entering the range of  $y$  values is optional but including them can improve the graph’s appearance.
- If you are graphing several functions, the  $x$  and  $y$  limits do not have to be the same. However, the largest set of limits will be used for all functions so they all fit in the window on the same axes.
- The equations you want evaluated can be comprised of integers, symbolic constants, decimal numbers, or any functions or operations with numeric results.

## Polar Plots

Polar plots are a useful way of investigating plane curves, particularly circles, ellipses, spirals and other curves with similar symmetries. The set of ordered pairs that make up a polar plot are  $(r, t)$  where  $r$  is the distance from the origin and  $t$  is the angle measure around the origin (increasing counterclockwise) where zero represents the positive  $x$ -axis.



### Entering Data

1. Tap on one of the cells in the column labeled  $r(t)$ .
2. Enter your function. Use “^” for exponents and “\*” for multiplication.
3. Enter the  $t$  values that you want your function evaluated from and to. Tab over to these spaces or tap in them with your stylus. This step is **not** optional for polar plots.
4. Tap on the pencil to draw the graph of the function.

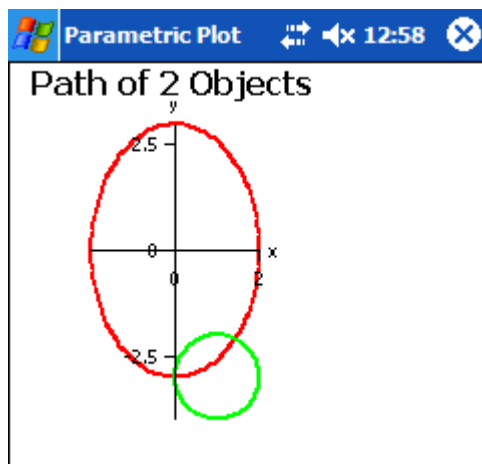
	$r(t)$	t-from	t-to
1	$\sin(4*t)$	0	pi
2	$\cos(5*t)$	0	pi
3			

### Note

- If you do not enter  $t$  limits you will get a syntax error when you try to plot the functions.
- If you are graphing more than one function, the  $t$  limits do not have to be the same. This can be useful, for graphing different sections of the same function in different colors.
- The functions you want evaluated can be comprised of integers, symbolic constants, decimal numbers, or any functions or operations with numeric results.

## Parametric Plots

A parametric plot is a graph of two functions on an  $xy$ -plane. The set of ordered pairs that make up the graph are  $(f(t), g(t))$ , where  $f$  and  $g$  are functions of another variable  $t$ .



### Entering Data

1. Tap in one of the cells in the column labeled  $x=f(t)$ .
2. Enter your function. Use “^” for exponents and “\*” for multiplication.
3. Enter a second function in the  $y=g(t)$  column.
4. Enter the  $t$  values that you want your functions evaluated from and to. This step is **not** optional for parametric plots.
5. Tap on the pencil icon to draw the graph of the functions.

	$x=f(t)$	$y=g(t)$	t-from
1	$2*\cos(t)$	$3*\sin(t)$	0
2	$1+\sin(t)$	$\cos(t)-3$	0
3			

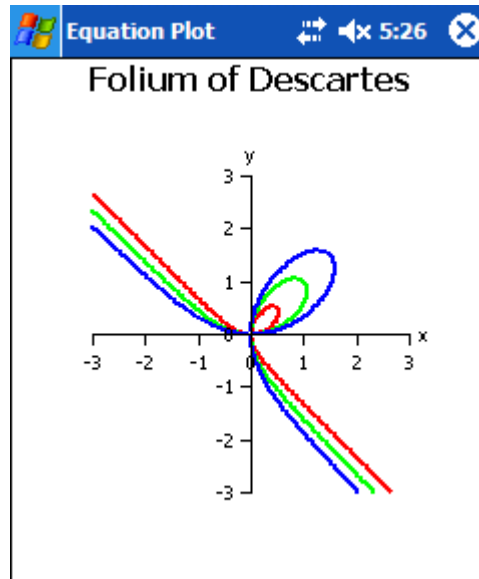
### Note

- If you do not enter t-from and t-to values you will get a syntax error when you plot the functions.
- If you are graphing more than one function, the  $t$  values do not have to be the same. This can be useful, for graphing different sections of the same function in different colors.
- The functions you want evaluated can be comprised of integers, symbolic constants, decimal numbers, or any functions or operations with numeric results.



## Graphs of Equations

An equation is a mathematical sentence with an equal (=) sign.



### Entering Data

1. Tap in one of the cells in the column labeled equation.
2. Enter your equation. Use “^” for exponents and “\*” for multiplication.
3. Enter the  $x$  and the  $y$  limits that you want your equation evaluated from and to.
4. Tap on the pencil icon to draw the graph of the equation.

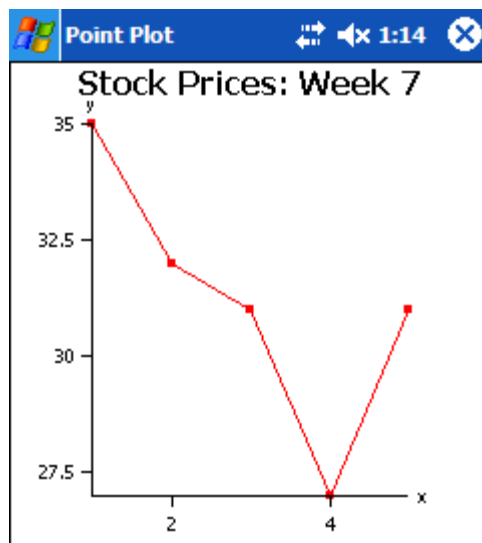
	equation	x-from
1	$x^3+y^3=x*y$	-2.5
2	$x^3+y^3=2*x*y$	-2.5
3	$x^3+y^3=3*x*y$	-2.5
4		

### Note

- The  $x$  and  $y$  limits that you want the equation evaluated from and to must be entered. If you do not enter them, you will get a syntax error when you try to plot the equations.
- If you are graphing more than one equation, the  $x$  and  $y$  limits do not have to be the same. However, the largest set of limits is used for all equations so they fit on the same set of axes.
- The equations you want evaluated can be comprised of integers, symbolic constants, decimal numbers, or any functions or operations with numeric results.

## Point Plots

A point plot is a graph that connects all of the points of data with a straight line.



### Entering Data

1. Tap in one of the spaces in the column labeled x.
2. Enter a value for x.
3. Enter in a value for y. Use the Tab key to move the cursor over to these spaces or tap in them with your stylus.
4. Tap on the pencil icon to draw the graph of the points.
5. To draw the line connecting the points go to the Options menu (under the Edit menu) and tap on Style and Line.

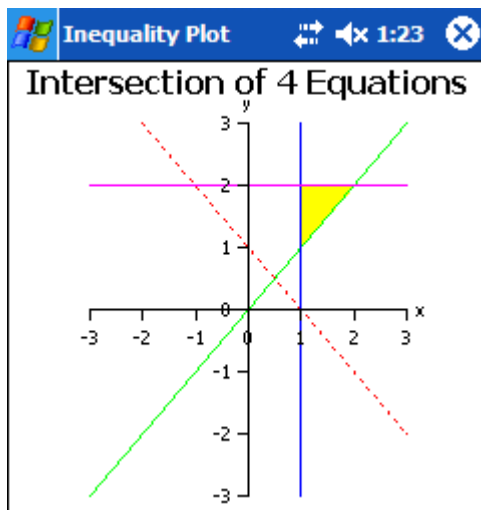
	x	y
1	1	35
2	2	32
3	3	31
4	4	27
5	5	31

### Note

- Data may take the form of integers, decimal numbers, symbolic constants or calculations.
- If you do enter inappropriate data and try to plot it, a syntax error message appears.

## Graphs of Inequalities

An inequality is a mathematical sentence that contains one of the following relations:  $>$  (greater than),  $<$  (less than),  $\geq$  (greater than or equal to),  $\leq$  (less than or equal to). A graph of inequalities shows the region of intersection of two or more inequalities. Shading indicates the region of intersection. If there is no shaded region in an inequality graph, then there is no set of points that satisfies all the inequalities.



### Entering Data

1. Tap in one of the cells in the column labeled Inequalities.
2. Enter your inequalities. Use “^” for exponents and “\*” for multiplication.
3. Enter the  $x$  and the  $y$  limits that you want your inequality evaluated from and to.
4. Tap on the pencil icon to draw the graph of the inequality.

x-from	x-to	y-from	y-to
-3	3	-3	3
	Inequalities		
1	$x+y>1$		
2	$x-y\leq 0$		
3	$x\geq 1$		
4	$y\leq 2$		

### Note

- The equation must contain an inequality sign and only linear inequalities can be evaluated.
- You **must** enter the  $x$  limits and the  $y$  limits that you want the equation evaluated under. If you do not enter them, you will get a syntax error when you try to plot the inequalities.
- The equations you want evaluated can be comprised of integers, symbolic constants, decimal numbers, or any functions or operations with numeric results.
- When evaluating graphs of inequalities, a dashed line indicates that the boundary does not belong to the solution set. However, if you set the line thickness to medium or thick, dashed lines cannot be distinguished from solid lines. A solid line indicates that the line itself does belong to the solution set.

## Trouble Shooting

### Problems with *Portrait 4*

If you experience any difficulties with *Portrait 4*, with the installation or the running of any part of the program, please, contact our web site immediately. As we are made aware of any problems users of *Portrait 4* are experiencing, we will put this information on our web site along with our solution.

You can find our web site at **www.mathresources.com**.

### Contacting MathResources Inc.

We would love to hear from you. If you have any comments and/or suggestions about *Portrait 4* or this User's Guide or if you have some great ideas about other products that you would like to see, please, let us know. Your feedback will help us improve on our current products and help us create better ones in the future.

You can reach us in the following ways:

- Email: [support@mathresources.com](mailto:support@mathresources.com)
- Visit our web site at **www.mathresources.com**
- Call us at: 1-800-720-1323 (within North America) **OR** 1-902-429-1323
- Fax us at: 1-902-492-7101

### Copyrights and Trademarks

*Portrait 4* is a registered trademark (pending) of MathResources Inc.

Windows and WindowsCE are registered trademarks of Microsoft Corporation.

*The User's Guide to Portrait 4* and *Portrait 4* are copyright of MathResources Inc.

No part of this manual or *Portrait 4* may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of MathResources Incorporated.