HP 39gII graphing calculator

user's guide

Edition 1

Part Number NW249-90001

Printing History

Edition 1 November 2011

Contents

17 Lists

18 Matrices

19 Notes and Info

20 Variables and memory management

23 Appendix: Product Regulatory Information

Preface

Manual conventions

The following conventions are used in this manual to represent the keys that you press and the menu options that you choose to perform the described operations.

- Key presses are represented as follows:

  SIN ASIN E, COS ACOS F, Home Modes, etc.

- Shift keys, that is the key functions that you access by pressing the SHIFT key first, are represented as follows:

  SHIFT *CLEAR*, SHIFT *MODES*, SHIFT *ACOS*, etc.

- Numbers and letters are represented normally, as follows:

  5, 7, A, B, etc.

- Menu options, that is, the functions that you select using the menu keys at the top of the keypad are represented as follows:
  STO ►, CANCL, OK.

- Input form fields and choose list items are represented as follows:

  `Function`, `Polar`, `Parametric`

- Your entries as they appear on the command line or within input forms are represented as follows:

  `2*X`^{2}`-3X+5`

Notice

This manual and any examples contained herein are provided as-is and are subject to change without notice. Except to the extent prohibited by law, Hewlett-Packard Company makes no express or implied warranty of any kind with regard to this manual and specifically disclaims the implied warranties and conditions of merchantability and fitness for a particular purpose and Hewlett-Packard Company shall not be liable for any errors or for incidental or consequential damage in connection with the furnishing, performance or use of this manual and the examples herein.

© 1994–1995, 1999–2000, 2003–2006, 2010–2011 Hewlett-Packard Development Company, L.P.

The programs that control your HP 39gII are copyrighted and all rights are reserved. Reproduction, adaptation, or translation of those programs without prior written permission from Hewlett-Packard Company is also prohibited.

For Hardware warranty information, please refer to the HP 39gII Quick Start Guide.

For Product Regulatory and Environment Information, please refer to the HP 39gII Quick Start Guide.

1

Getting started

On/off, cancel operations

To turn on

Press ON/C to turn on the calculator.

To cancel

When the calculator is on, the ON/C key cancels the current operation.

To turn off

Press SHIFT *OFF* to turn the calculator off.

To save power, the calculator turns itself off after several minutes of inactivity. All stored and displayed information is saved.

If you see the battery annunciator, then the calculator needs fresh batteries.

The Home view

Home is the calculator's home view and is common to all apps. If you want to perform calculations, or you want to quit the current activity (such as an app, a program, or an editor), press Home. All mathematical functions are available in the Home view. The name of the current app is displayed in the title of the home view.

Protective cover

The calculator is provided with a slide cover to protect the display and keyboard. Remove the cover by grasping both sides of it and pulling down.

You can reverse the slide cover and slide it onto the back of the calculator. This will help you keep track of the cover while you are using the calculator.

To prolong the life of the calculator, always place the cover over the display and keyboard when you are not using the calculator.

The display

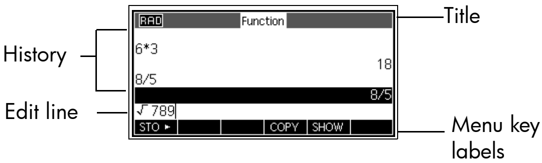
To adjust the contrast

To adjust the contrast, press and hold ON/C OFF, then press the + or - keys to increase or decrease the contrast. The contrast will change with each press of the + or - keys.

To clear the display

- Press *CANCEL* to clear the edit line.
- Press SHIFT *CLEAR* once to clear an active edit line and again to clear the display history.

Parts of the display



Menu key labels. The top row of keys on the HP 39gII keyboard (F1-F6) are the menu keys. These keys give you access to the menu items shown at the bottom of the display. STO ▸ is the label for the first menu key in the figure above. "Press STO ▸" means press the F1 menu key.

Edit line. The line of current entry.

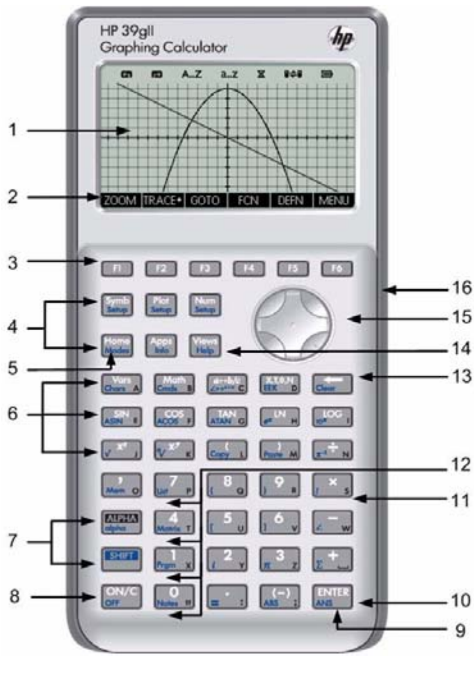
History. The Home display (Home Modes) shows up to 6 lines of history: the most recent input and output. Older lines scroll off the top of the display but are retained in memory.

Title. The name of the current app is displayed at the top of the Home view. RAD or DEG specifies whether Radians or Degrees is the current angle measurement mode. The ▼ and ▲ symbols indicate there is more history in the display. Press ▼ and ▲ to scroll in the history display.

Annunciators. Annunciators are symbols that appear above the title bar and give you important status information.

Annunciator	Description
	To activate, press [SHIFT]. Shift in effect for next keystroke. To cancel, press [SHIFT] again.
A..Z	To activate, press [ALPHA]. Alpha in effect for next keystroke. To lock, press [ALPHA] again. To cancel, press [ALPHA] a third time.
a..z	To activate, press [ALPHA] [SHIFT]. Lower-case alpha in effect for next keystroke. To lock, press [ALPHA] again. To cancel, press [ALPHA] a third time. To switch to upper-case, press [SHIFT].
	Low battery power.
	Busy.
	Data is being transferred via cable.

The keyboard

Number	Feature	HP 39gII
1	256 x 128 pixel display	
2	Context-sensitive menu	
3	F1-F6 menu keys	
4	HP Apps keys	
5	Modes	
6	Common math and science functions	
7	Shift keys	
8	On (cancel)	
9	Last Answer (ANS)	
10	Enter key	
11	Alphabetic entry	
12	Catalogs and editors	
13	Backspace (Clear)	
14	Help key	
15	Cursor keys	
16	USB Connectivity	

Menu keys

- On the calculator keyboard, the keys in the top row of keys (labeled F1-F16) are called menu keys. Their meanings depend on the context; that is, the view you are in.
- The bottom line of the display shows the labels for the menu keys' current meanings.

App control keys

The app control keys are:

Key	Meaning
Symb Setup	Displays the Symbolic view for the current app.

Key	Meaning (Continued)
Plot Setup	Displays the Plot view for the current app.
Num Setup	Displays the Numeric view for the current app.
Home Modes	Displays the Home view, for performing calculations.
Apps Info	Displays the App Library menu.
Views Help	Displays the VIEWS menu.

Entry/Edit keys

The entry and edit keys are:

Key	Meaning
ON/C OFF *(CANCEL)*	Cancels the current operation if the calculator is on by pressing ON/C OFF. Pressing SHIFT, then *OFF* turns the calculator off.
SHIFT	Accesses the function printed at the bottom left of a key.
ALPHA	Accesses the alphabetical characters printed at the bottom right of a key. Press ALPHA twice to lock this shift so you can enter a string of characters.
ENTER ANS	Enters an input or executes an operation. In calculations, ENTER ANS acts like "=". When OK or START is present as a menu key, ENTER ANS acts the same as pressing OK or START.
(–) ABS ;	Enters a negative number. To enter –25, press (–) ABS ; 25. *Note: this is not the same operation that the subtraction key performs* (– ∠ W).

Key	Meaning (Continued)
X,T,θ,N EEX D	Enters the independent variable by inserting *X*, *T*, θ, or *N* into the edit line, depending on the current active app.
← Clear	Backspace. Deletes the character to the left of the cursor.
SHIFT *CLEAR*	Clears all data on the screen. On a settings screen, for example Plot Setup, SHIFT *CLEAR* returns all settings to their default values.
◀ ▶ ▲ ▼	Moves the cursor around the display. Press SHIFT first to move to the beginning, end, top or bottom.
SHIFT *CHARS*	Displays a menu of all available characters. To type one, use the arrow keys to highlight it, and press OK. To select multiple characters, select each and press ECHO, then press OK.

Shifted keystrokes

There are two shift keys that you use to access the operations and characters printed on the bottom of the keys: SHIFT and ALPHA.

Key	Description
SHIFT	Press SHIFT to access the operations printed on the bottom (or bottom left) of a key. For instance, to access the Modes input form, press SHIFT and then press Home Modes because Modes is printed on the bottom of the Home key.

Key	Description (Continued)
[ALPHA]	Press the [ALPHA] key to access the alphabetic character printed on the bottom right of a key. For instance, to type Z, press [ALPHA] and then press [3] because Z is printed on the bottom right of the [3] key. For a lower case letter, press [ALPHA] then [SHIFT]. To type more than one letter, press [ALPHA] a second time to lock the Alpha shift.

Help

Press [SHIFT] [Views/Help] (Help) to enter the HP 39gII built-in Help system. The Help system always opens in your current context or view, giving you information about the current view and its menu items. Once in the Help system, you can navigate to other topics and find help on any view or command.

Example:

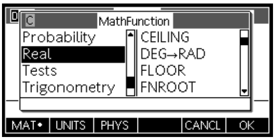
Press [Apps/Info] and select Function. Press [SHIFT] [Views/Help] (Help) to get help on the purpose of the Function App.

Math keys

Home ([Home/Modes]) is the place to do calculations.

Keyboard keys. The most common operations are available from the keyboard, such as the arithmetic (like [+]) and trigonometric (like [SIN]) functions. Press [ENTER/ANS] to complete the operation: [SHIFT] [x^2] 256 [ENTER/ANS] displays 16.

Math menu. Press [Math/Cmds] to open the Math menu. The Math menu is a comprehensive list of math functions that do not appear on the keyboard. It also includes categories for all other functions and constants. The functions are grouped by category, ranging in alphabetical order from `Calculus` to `Trigonometry`.



- Use the up- and down-arrow keys to scroll through the list. Use the right- and left-arrow keys to move between the category and item columns.

- Press **OK** to insert the selected command into the edit line at the curent cursor position.
- Press **CANCL** to dismiss the Math menu without selecting a command.
- Press **UNITS** to attach units to a number in the edit line.
- Press **PHYS** to display a menu of physical constants from the fields of chemistry, physics, and quantum mechanics. You can use these constants in calculations.
- Press **MATH** to return to the Math menu.

See the chapter *Using Mathematical Functions* for details.

HINT When using the Math menu, or any menu on the HP 39gII, the categories and items are numbered for your convenience. For example, `ITERATE` is the first item under `Loop`, which is the eighth category. With the Math menu open, press [8] [1] to insert the `ITERATE` function in the edit line at the cursor position. If there are more than 9 items in a category, the letters `A`, `B`, `C`, etc. are used. For example, the Matrix category uses the number `8`. In this category, the `RREF` command uses the letter `H`. With the Math menu open, press [9] [H] to insert the `RREF` command into the edit line. You do not need to press [ALPHA] to access the letter you want.

Program commands

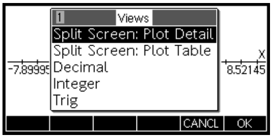
Pressing [SHIFT] *CMDS* displays the list of Program Commands. See the chapter *Programming* for more details.

Inactive keys

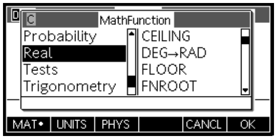
If you press a key that does not operate in the current context, a warning symbol like this ⚠ appears. There is no beep.

Menus

A menu offers you a choice of items. Menus are displayed in 1-3 columns.



- The ▼ arrow means more items below.
- The ▲ arrow means more items above.



To search a menu

- Press ▼ or ▲ to scroll through the list. If you press SHIFT ▼ or SHIFT ▲, you'll go all the way to the end or the beginning of the list. Highlight the item you want to select, then press OK (or ENTER ANS).
- If there are two columns, the left column shows general categories and the right column shows specific contents within a category. Highlight a general category in the left column, then highlight an item in the right column. The list in the right column changes when a different category is highlighted.
- If there are three columns, the left column shows a general category while the second column shows a useful sub-category. Highlight a general category, then highlight a sub-category of interest. Finally, select an item from the third column.
- To speed-search a list, type the number or letter of the category, followed by the number or letter of the item.

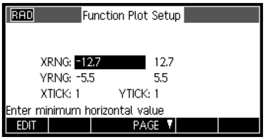
  For example, to find the List category in Math Cmds B, press 7 List P.

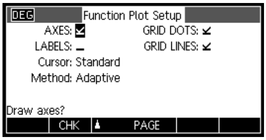
To cancel a menu

Press ON/C OFF (for *CANCEL)* or CANCL. This cancels the current operation.

Input forms

An input form shows several fields of information for you to examine and specify. After highlighting the field to edit, you can enter or edit a number (or expression). You can also select options from a list (CHOOS). Some input forms include items to check (CHK). See below for example input forms.





Reset input form values

To reset a field to its default value in an input form, move the cursor to that field and press Clear. To reset all field values in the input form to their default values, press SHIFT *CLEAR*.

Mode settings

You use the Modes input form to set the modes for Home.

HINT Although the numeric setting in Modes affects only Home, the angle setting controls Home and the current app. The angle setting selected in Modes is the angle setting used in both Home and the current app. To further configure an app, you use the *SETUP* keys (SHIFT Symb Setup, SHIFT Plot Setup, and SHIFT Num Setup).

Press SHIFT Home Modes (Modes) to enter the Home Modes input form. Press PAGE (F4) to enter the second page of the form and press PAGE (F3) to return to the first page

Setting	Options
Angle Measure	Angle values are: **Degrees.** 360 degrees in a circle. **Radians.** 2π radians in a circle. The angle mode you set is the angle setting used in both Home and the current app. This is done to ensure that trigonometric calculations done in the current app and Home give the same result.

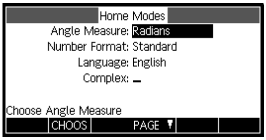
Setting	Options (Continued)
Number Format	The number format mode you set is the number format used in all Home view calculations. **Standard.** Full-precision display. **Fixed.** Displays results rounded to a number of decimal places. Example: 123.456789 becomes 123.46 in Fixed 2 format. **Scientific.** Displays results with an exponent, one digit to the left of the decimal point, and the specified number of decimal places. Example: 123.456789 becomes 1.23E2 in Scientific 2 format. **Engineering.** Displays result with an exponent that is a multiple of 3, and the specified number of significant digits beyond the first one. Example: 123.456E7 becomes 1.23E9 in Engineering 2 format.
Complex	If checked, allows operations involving complex numbers; if unchecked, only real-number operations are allowed.
Language	Choose language preference for menus and input forms.
Font Size	Choose a smaller or larger font for most display purposes.
Calculator Name	Calculator NameEnter a descriptive name to identify your calculator to the HP 39gII Connectivity Kit.
Textbook Display	Disable or enable Textbook Format Display for expressions entered in the Home and Symbolic views.

Setting a mode

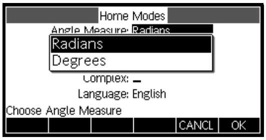
This example demonstrates how to change the angle measure from the default mode, radians, to degrees for the current app. The procedure is the same for changing number format, language, and complex number modes.

1. Press SHIFT *MODES* to open the `Home Modes` input form.

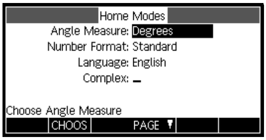
   The cursor (highlight) is in the first field, `Angle Measure`.



2. Press CHOOS to display a list of choices.



3. Use the up- and down-arrow keys to select Degrees and press OK. The angle measure changes to degrees.



4. Press Home Modes to return to Home.

HINT Whenever an input form has a list of choices for a field, you can press + to cycle through them instead of using CHOOS.

Mathematical calculations

The most commonly used math operations are available from the keyboard. Access to the rest of the math functions is via the Math menu (Math Cmds B).

To access programming commands, press SHIFT *CMDS*. See the chapter *Programming* for more information.

Where to start

The home base for the calculator is the Home view (Home Modes). You can do all calculations here, and you can access all Math Cmds B operations.

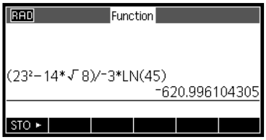
Entering expressions

- Enter an expression into the HP 39gII in the same left-to-right order that you would write the expression. This is called *algebraic entry.*
- To enter functions, select the key or Math menu item for that function. You can also enter a function by using the Alpha keys to spell out its name.
- Press ENTER/ANS to evaluate the expression you have in the edit line (where the blinking cursor is). An *expression* can contain numbers, functions, and variables.

Example

Calculate $\frac{23^2 - 14\sqrt{8}}{-3}\ln(45)$:

(23 x²
– 14
× SHIFT √ 8
)
÷ (–) 3 ×
LN 45)
ENTER



Long results

If the result is too long to fit on the display line, or if you want to see an expression in textbook format, press ▲ to highlight it and then press SHOW.

Negative numbers

Type (–) to start a negative number or to insert a negative sign.

To raise a negative number to a power, enclose it in parentheses. For example, $(-5)^2 = 25$, whereas $-5^2 = -25$.

Scientific notation (powers of 10)

A number like 5×10^4 or 3.21×10^{-7} is written in *scientific notation*, that is, in terms of powers of ten. This is simpler to work with than 50000 or 0.000000321. To enter numbers like these, use *EEX*. This is easier than using × 10 x^y .

Example

Calculate $\frac{(4 \times 10^{-13})(6 \times 10^{23})}{3 \times 10^{-5}}$

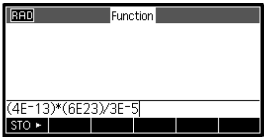
(4 SHIFT *EEX* (−)

13)

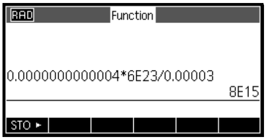
× (6 SHIFT *EEX*

23)

÷ 3 SHIFT *EEX* (−) 5



ENTER



Explicit and implicit multiplication

Implied multiplication takes place when two operands appear with no operator in between. If you enter `AB`, for example, the result is `A*B`.

However, for clarity, it is better to include the multiplication sign where you expect multiplication in an expression. It is clearest to enter `AB` as `A*B`.

Parentheses

You need to use parentheses to enclose arguments for functions, such as SIN(45). You can omit the final parenthesis at the end of an edit line. The calculator inserts it automatically.

Parentheses are also important in specifying the order of operation. *Without* parentheses, the HP 39gII calculates according to the order of *algebraic precedence* (the next topic). Following are some examples using parentheses.

Entering...	**Calculates...**
SIN 45 + SHIFT π	sin (45 + π)
SIN 45) + SHIFT π	sin (45) + π
SHIFT √ 85 × 9	$\sqrt{85} \times 9$
SHIFT √ (85 × 9)	$\sqrt{85 \times 9}$

Algebraic precedence order of evaluation

Functions within an expression are evaluated in the following order of precedence. Functions with the same precedence are evaluated in order from left to right.

1. Expressions within parentheses. Nested parentheses are evaluated from inner to outer.
2. Prefix functions, such as SIN and LOG.
3. Postfix functions, such as !
4. Power function, ^, NTHROOT.
5. Negation, multiplication, and division.
6. Addition and subtraction.
7. AND and NOT.
8. OR and XOR.
9. Left argument of | (where).
10. Equals, =.

Largest and smallest numbers

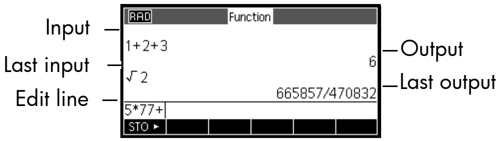
The HP 39gII represents 1×10^{-499} (as well as all numbers smaller than this) as zero. The largest number displayed is $9.99999999999 \times 10^{499}$. A greater result is displayed as this number.

Clearing numbers

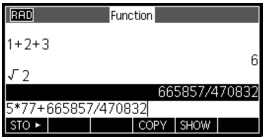
- [Clear] deletes the character to the left of the cursor; that is, it is a backspace key.
- *CANCEL* ([ON/C]) clears the edit line.
- [SHIFT] *CLEAR* clears all input and output in the display, including the display history.

Using previous results

The Home display ([Home/Modes]) shows you 4-6 lines of input/output history. An unlimited (except by memory) number of previous lines can be displayed by scrolling. You can retrieve and reuse any of these values or expressions.



When you highlight a previous input or result (by pressing [▲]), the COPY and SHOW menu labels appear.



To copy a previous line

Highlight the line (press [▲]) and press COPY. The number (or expression) is copied into the edit line.

Your last few entries are always copied to the clipboard, so in most cases, you can just paste a recent result. Press [SHIFT] [Paste] to open the clipboard, use [▼] and [▲] to highlight the result you want, and press OK.

To reuse the last result

Press [SHIFT] *ANS* (last answer) to put the last result from the Home display into an expression. *ANS* is a variable that is updated each time you press [ENTER].

To repeat a previous line

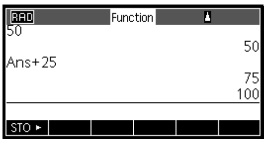
To repeat the very last line, just press [ENTER]. If the previous line is an expression containing *ANS*, the calculation is repeated iteratively.

Example

See how [SHIFT] *ANS* retrieves and reuses the last result (50), and [ENTER] updates *ANS* (from 50 to 75 to 100).

50 [ENTER] [+] 25

[ENTER] [ENTER]



You can use the last result as the first expression in the edit line without pressing [SHIFT] *ANS*. Pressing [+], [-], [×], [÷], (or other operators that require a preceding argument) automatically enters *ANS* before the operator.

You can reuse any other expression or value in the Home display by highlighting the expression (using the arrow keys), then pressing COPY.

The variable *ANS* is different from the numbers in Home's display history. A value in *ANS* is stored internally with the full precision of the calculated result, whereas the displayed numbers match the display mode.

HINT When you retrieve a number from *ANS*, you obtain the result to its full precision. When you retrieve a number from the Home's display history, you obtain exactly what was displayed.

Pressing [ENTER ANS] evaluates (or re-evaluates) the last input, whereas pressing [SHIFT] *ANS* copies the last result (as *ANS*) into the edit line.

Copy and paste

In addition to the COPY menu key that lets you copy expressions from the Home view, there is a more universal copy and paste clipboard that you can use. You can highlight the value or expression you want in most fields or the Home view history (e.g. F1(x) in the Function App) and then paste it into the edit line or into another compatible field. To copy a value or expression to the clipboard, press [SHIFT] [Copy (L]. To open the clipboard to select and paste a value or expression, press [SHIFT] [Paste) M].

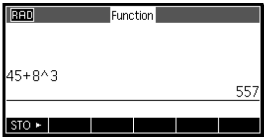
Storing a value in a variable

You can save an answer in a variable and use the variable in later calculations. There are 27 variables available for storing real values. These are *A* to *Z* and θ. See the chapter *Variables and memory management* for more details on variables. For example:

1. Perform a calculation.

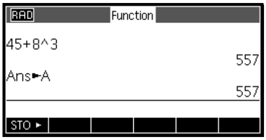
   45 + 8 x^y 3

   ENTER



2. Store the result in the *A* variable.

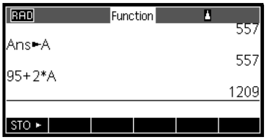
   STO ► ALPHA A ENTER



3. Perform another calculation using the *A* variable.

   95 + 2 × ALPHA

   A ENTER



Accessing the display history

Pressing ▲ enables the highlight bar in the display history. While the highlight bar is active, the following menu and keyboard keys are very useful:

Key	Function
▲, ▼	Scrolls through the display history.
COPY	Copies the highlighted expression to the position of the cursor in the edit line.
SHOW	Displays the current expression using Textbook Format Display.
← Clear	Deletes the highlighted expression from the display history, unless there is a cursor in the edit line.
SHIFT *CLEAR*	Clears all lines of display history and the edit line.

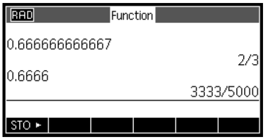
Clearing the display history

It's a good habit to clear the display history (SHIFT *CLEAR*) whenever you have finished working in Home. It saves calculator memory to clear the display history. Remember that *all* your previous inputs and results are saved until you clear them.

Numerical representations

Converting decimals to fractions

Any decimal result can be displayed as a decimal, a fraction, or a mixed number. Enter your expression in the Home view and then



press [a↔b/c] to toggle through fraction, mixed number, and decimal representations of the numerical result. For example, enter 18/7 to see the decimal result: 2.5714…. Press [a↔b/c] once to see $\frac{18}{7}$ and again to see $2+\frac{4}{7}$. The 39gII will approximate fraction and mixed number representations in cases where it cannot find exact ones. Enter $\sqrt{5}$ to see the decimal approximation: 2.236…. Press [a↔b/c] once to see $\frac{930249}{416020}$ and again to see $2+\frac{98209}{416020}$.

Pressing [a↔b/c] a third time will cycle back to the original decimal representation.

Converting decimals to degrees, minutes, and seconds

Any decimal result can de displayed in hexagesimal; that is, in units subdivided into groups of 60. This includes degrees, minutes, and seconds as well as hours, minutes, and seconds. For example, enter $\frac{11}{8}$ to see the decimal result: 1.375. Press [SHIFT] [a↔b/c] to see $1°22'30''$. Press [SHIFT] [a↔b/c] again to return to the decimal representation. The 39gII will produce the best approximation in cases where an exact result is not possible. Again, enter $\sqrt{5}$ to see the decimal approximation: 2.236…. Press [SHIFT] [a↔b/c] to see $2°14'9.844719''$.

Complex numbers

Complex results

If the Complex mode setting is checked, then the HP 39gII can return a complex number as a result for some math functions. A complex number appears as $x + y \times i$. For example, entering $\sqrt{-1}$ returns i and entering (4,5) returns $4 + 5 \times i$.

To enter complex numbers

Enter the number in either of these forms, where *x* is the real part, *y* is the imaginary part, and i is the imaginary constant, $\sqrt{-1}$:

- (x, y) or
- $x + iy$.

To enter i:

- press SHIFT ALPHA LOG or
- press Math, ▲ or ▼ keys to select `Constant`, ▶ to move to the right column of the menu, ▼ to select i, and OK.

Storing complex numbers

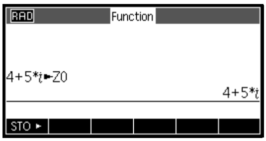
There are ten variables available for storing complex numbers: Z0 to Z9. To store a complex number in a variable:

- Enter the complex number, press STO ▸, enter the variable to store the number in, and press ENTER.

(4 , 5)

STO ▸

ALPHA Z 0 ENTER



Catalogs and editors

The HP 39gII has several catalogs and editors. You use them to create and manipulate objects. They access objects with stored data (lists of numbers or notes with text) that are independent of apps, as well as notes and programs attached to the current HP App.

- A *catalog* lists items, which you can delete or transmit, for example an app.
- An *editor* lets you create or modify items and numbers, for example a note or a matrix.

Catalog/Editor	**Keystrokes**	**To create and edit**
App library	[Apps Info]	HP Apps
Info	[SHIFT] [Apps Info] (Info)	Notes attached to the current HP App
List	[SHIFT] [7 List P] (List)	Lists
Matrix	[SHIFT] [4 Matrix T] (Matrix)	Matrices and vectors
Program	[SHIFT] [1 Prgm X] (Prgm)	Programs
Notes	[SHIFT] [0 Notes ''] (Notes)	Notes

2

Apps and their views

HP Apps

HP Apps are applications designed for the study and exploration of a branch of mathematics or to solve problems of one or more types. The following table lists the name of each HP App and gives a general description of its purpose.

App name	**Use this app to explore:**
Function	Real-valued, rectangular functions *y* in terms of *x*. Example: $y = 2x^2 + 3x + 5$.
Solve	Equations in one or more real-valued variables. Example: $x + 1 = x^2 - x - 2$.
Statistics 1Var	One-variable statistical data (*x*)
Statistics 2Var	Two-variable statistical data (*x* and *y*)
Inference	Confidence intervals and Hypothesis tests based on the Normal and Students-t distributions.
Parametric	Parametric relations *x* and *y* in terms of *t*. Example: $x = \cos(t)$ and $y = \sin(t)$.
Polar	Polar functions *r* in terms of an angle θ. Example: $r = 2\cos(4\theta)$.

App name	Use this app to explore: (Continued)
Sequence	Sequence functions *U* in terms of *n*, or in terms of previous terms in the same or another sequence, such as U_{n-1} and U_{n-2}. Example: $U_1 = 0$, $U_2 = 1$ and $U_n = U_{n-2} + U_{n-1}$.
Finance	Time Value of Money (TVM) problems and amortization tables.
Linear Solver	Solutions to sets of two or three linear equations.
Triangle Solver	Unknown values for the lengths and angles of triangles.
Data Streamer	Real-world data collected from scientific sensors.

In addition to these apps, which can be used in a variety of applications, the HP 39gII is supplied with three apps for exploring function families: The Linear, Quadratic, and Trig Explorers. These apps will retain their data so you can return to them and find them as you left them, but they are not designed to be customized and saved like the other HP Apps.

As you use an app to explore a lesson or solve a problem, you add data and definitions in the app's views. All of this information is automatically saved in the app. You can come back to the app at any time and the information is all still there. Or you can save the app with a name you give it and then use the original app for another problem or purpose. See the chapter *Extending Your Aplet Library* for more information regarding customizing and saving HP Apps.

App library

Apps are stored in the App library.

To open an app

Press Apps/Info to display the App library menu. Select the app and press START or ENTER/ANS.

From within an app, you can return to Home any time by pressing Home/Modes.

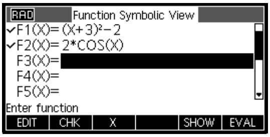
App views

The HP Apps all utilize the same set of views and it is this consistency in the use of views that make them easy to learn and to use. There are three major views, known as the Symbolic, Plot (Graphic), and Numeric views. These views are based on the symbolic, graphic, and numeric representations of mathematical objects and are accessed through the [Symb Setup], [Plot Setup], and [Num Setup] keys near the top of the keyboard. The SHIFT of these keys provides access to the view's setup, in which the view is configured. One additional user-defined view, Info, is provided to add notes to an app. Finally, the Views key provides access to any additional, special views an app may have. Note that not all HP Apps provide all 7 of the standard views, nor do all of them provide additional views via the Views key. The scope and complexity of each app determines its view set. However, the views provided are based on these seven views and the additional views provided by the Views key. These views are summarized below, using the Function app as an example.

Symbolic view

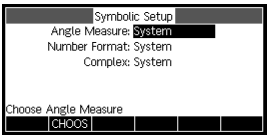
Press [Symb Setup] to display the app’s Symbolic view.

You use this view to define the function(s) or equation(s) that you want to explore.



Symbolic setup

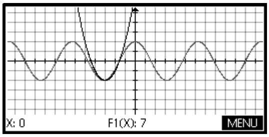
Press [SHIFT] *SETUP-SYMB* to display the app's Symbolic setup. The purpose of this view is to allow you to overwrite one or more of the Modes settings for an app. This view is not used by the Solvers and Explorers, as the few mode settings needed for each app can already be changed by using menu keys within the app.



Plot view

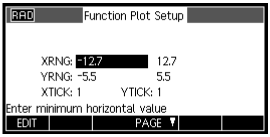
Press [Plot Setup] to display the app’s Plot view.

In this view, the relations that you have defined are displayed graphically.



Plot setup

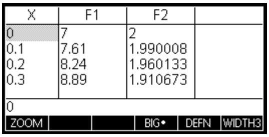
Press [SHIFT] *SETUP-PLOT*. Sets parameters to plot a graph.



Numeric view

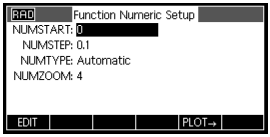
Press [Num Setup] to display the app’s Numeric view.

In this view, the relations that you have defined are displayed in tabular format.



Numeric setup

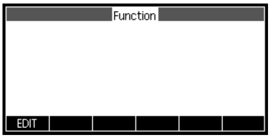
Press [SHIFT] *SETUP-NUM*. Sets parameters for building a table of numeric values.



Info view

Press [SHIFT] *INFO* to display the HP App's Info view.

This note is transferred with the app if it is sent to another calculator or to a PC. The Info view contains text to supplement an HP App.



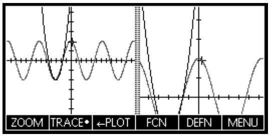
The Views menu

Besides the 7 views that all HP Apps can utilize, the Views key provides access to any special views or scaling options that an app may have or that some of the apps may share in common. These views and scaling options are summarized below.

Plot-Detail view

Press Views (Help)
Select `Plot-Detail`
OK

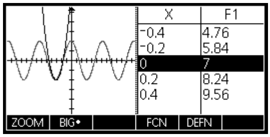
Splits the screen into the current plot and a user-defined zoom.



Plot-Table view

Press Views (Help)
Select `Plot-Table` OK

Splits the display, showing both the plot and tabular views.



Preset zooms

The Views menu also contains the same preset zooms from the Zoom menu:

- Auto Scale
- Decimal
- Integer
- Trig

These are described in more detail in the *Zoom options* section later in this chapter.

Standard app views

This section examines the options and functionality of the three main views (Symbolic, Plot, and Numeric), as well as their setups, for the Function, Polar, Parametric, and Sequence apps.

About the Symbolic view

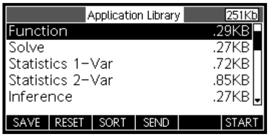
The Symbolic view is the *defining view* for the Function, Parametric, Polar, and Sequence apps. The other views are derived from the symbolic expression.

You can create up to 10 different definitions for each Function, Parametric, Polar, and Sequence app. You can graph any of the relations (in the same app) simultaneously by selecting them.

Defining an expression (Symbolic view)

Choose the app from the App Library.

Apps Info



Press ⏶ or ⏷ to select an app.

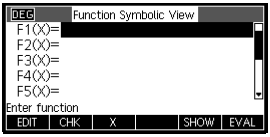
START

The Function, Parametric, Polar, and Sequence apps start in the Symbolic view.

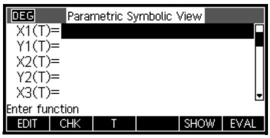
If the highlight is on an existing expression, scroll to an empty line—unless you don't mind writing over the expression—or, clear one line (Clear) or all lines (SHIFT *CLEAR*).

Expressions are selected (check marked) on entry. To deselect an expression, press CHK. All selected expressions are plotted.

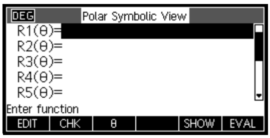
– **For a Function definition**, enter an expression to define *F(X)*. The only independent variable in the expression is *X*.



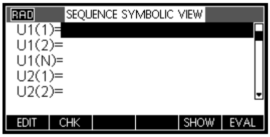
– **For a Parametric definition**, enter a pair of expressions to define *X(T)* and *Y(T)*. The only independent variable in the expressions is *T*.



– **For a Polar definition**, enter an expression to define *R*(θ). The only independent variable in the expression is θ.



– **For a Sequence definition**, either enter the first term, or the first and second terms for *U*. Then define the *n*th term of the sequence in terms of *N* or the prior terms, *U(N–1) and/or U(N–2)*. The expressions should produce real-valued sequences with integer domains. Or define the Nth term as a non-recursive expression in terms of N only.



– *Note*: you will have to enter the second term if the HP 39gII is unable to calculate it automatically. Typically if Ux(N) depends on Ux(N–2) then you must enter Ux(2).

Evaluating expressions

In apps

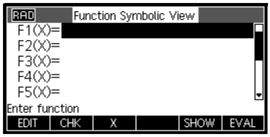
In the Symbolic view, a variable is a symbol only, and does not represent one specific value. To evaluate a function in Symbolic view, press EVAL. If a function calls another function, then EVAL resolves all references to other functions in terms of their independent variable.

1. Choose the Function app.

   Apps Info

   Select `Function`

   START



2. Enter the expressions in the Function app's Symbolic view.

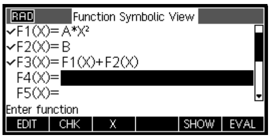
   ALPHA A × X

   x² OK

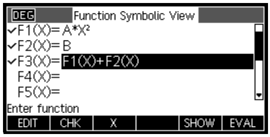
   ALPHA B OK

   ALPHA F1 (X) +

   ALPHA F2 (X) OK

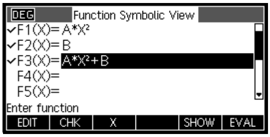


3. Highlight F3(X).



4. Press **EVAL**

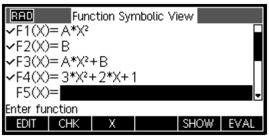
   Note how the values for F1(X) and F2(X) are substituted into F3(X).

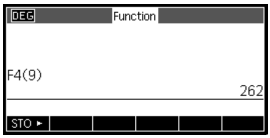


In Home

You can also evaluate any function expression in Home by entering it into the edit line and pressing ENTER/ANS.

For example, define F4 as below. In Home, type F4(9) and press ENTER/ANS. This evaluates the expression, substituting 9 in place of X in F4.





Symb view keys

The following table describes the keys that you use to work with the Symbolic view.

Key	Meaning
EDIT	Copies the highlighted expression to the edit line for editing. Press OK when done.
CHK	Checks/unchecks the current expression (or set of expressions). Only checked expression(s) are evaluated in the Plot and Numeric views.
X	Enters the independent variable in the Function app. Or, you can use the X,T,θ,N key on the keyboard.
T	Enters the independent variable in the Parametric app. Or, you can use the X,T,θ,N key on the keyboard.

Key	Meaning (Continued)
θ	Enters the independent variable in the Polar app. Or, you can use the X,T,θ,N EEX D key on the keyboard.
N	Enters the independent variable in the Sequence app. Or, you can use the X,T,θ,N EEX D key on the keyboard.
SHOW	Displays the current expression in Textbook Format.
EVAL	Resolves all references to other definitions in terms of variables.
Vars Chars A	Displays a menu for entering variable names or contents of variables.
Math Cmds B	Displays the menu for entering math operations.
SHIFT *CHARS*	Displays special characters. To enter one, place the cursor on it and press OK. To remain in the Chars menu and enter another special character, press ECHO.
← Clear	Deletes the highlighted expression or the current character in the edit line.
SHIFT *CLEAR*	Deletes all expressions in the list or clears the edit line.

About the Plot view

After entering and selecting (check marking) the expression in the Symbolic view, press Plot Setup. To adjust the appearance of the graph or the interval that is displayed, you can change the Plot view settings.

You can plot up to ten expressions at the same time. Select the expressions you want to be plotted together.

Plot setup

Press SHIFT `Setup-Plot` to define any of the settings shown in the next two tables.

1. Highlight the field to edit.

– If there is a number to enter, type it in and press ENTER/ANS or OK.

– If there is an option to choose, press CHOOS, highlight your choice, and press ENTER/ANS or OK. As a shortcut to CHOOS, just highlight the field to change and press + to cycle through the options.

– If there is an option to select or deselect, press CHK to check or uncheck it.

2. Press PAGE ▼ to view more settings.

3. When done, press Plot Setup to view the new plot.

Plot setup settings

The fields in the Plot setup are:

Field	**Meaning**
`XRNG, YRNG`	Specifies the minimum and maximum horizontal (*X*) and vertical (*Y*) values for the plotting window.
`TRNG`	Parametric app: Specifies the t-values (*T*) for the graph.
`θRNG`	Polar app: Specifies the angle (θ) value range for the graph.
`NRNG`	Sequence app: Specifies the index (*N*) values for the graph.
`TSTEP`	For Parametric plots: the increment for the independent variable.
`θSTEP`	For Polar plots: the increment value for the independent variable.
`SEQPLOT`	For Sequence app: Stairstep or Cobweb types.
`XTICK`	Horizontal spacing for tickmarks.
`YTICK`	Vertical spacing for tickmarks.

Those items with space for a checkmark are settings you can turn on or off. Press PAGE ▼ to display the second page.

Field	Meaning
AXES	Draws the axes.
LABELS	Labels the axes with XRNG and YRNG values.
GRID DOTS	Draws grid points using XTICK and YTICK spacing.
GRID LINES	Draws grid lines using XTICK and YTICK spacing.
Cursor	Choose between the Standard cursor and Inverting or Blinking cursors.
Method	Choose between the default Adaptive method for drawing accurate graphs and just plotting Fixed-Step Segments or Fixed-Step Dots.

Reset Plot setup

To reset the default values for all plot settings, press SHIFT *CLEAR* in the Plot Setup. To reset the default value for a field, highlight the field, and press Clear.

Exploring the graph

The Plot view gives you a selection of keys and menu keys to explore a graph further. The options vary from app to app.

Plot view keys

The following tables describe the keys that you use to work with the Plot view.

Key	Meaning
SHIFT *CLEAR*	Erases the plot and axes.
Views Help	Offers additional pre-defined views for splitting the screen and for scaling ("zooming") the axes.
STOP	Stops refining the graph

Key	Meaning (Continued)
MENU	Turns menu-key labels on and off. When the labels are off, pressing MENU turns them back on.
ZOOM	Displays the Zoom menu list.
TRACE	Turns trace mode on/off.
GOTO	Opens an input form for you to enter an *X* (or *T* or *N* or θ) value. Enter the value and press OK. The cursor jumps to the point on the graph that you entered.
FCN	Function app only: displays a list of commands for analyzing functions (see the chapter *Function app* for more details).
DEFN	Displays the current, *defining* expression. Press MENU to restore the menu.

The following tables detail the use of the arrow keys.

Key	Meaning (with trace mode off)
◀ ▶	Moves cursor one pixel left and right, respectively.
▲ ▼	Moves cursor one pixel up and down, respectively.
SHIFT ◀ SHIFT ▶	Moves cursor to far left or right edge of the display, respectively.
SHIFT ▲ SHIFT ▼	Moves cursor to the top or bottom of the display, respectively.

Key	Meaning (with trace mode on)
◀ ▶	Moves cursor one pixel left and right, respectively on the current graph.
▲ ▼	Switches the tracer from one graph to to the previous or next, respectively, in the list of symbolic definitions.
SHIFT ▲ SHIFT ▼	Moves the tracer to the leftmost or rightmost point on the current graph.
SHIFT ▲ SHIFT ▼	Not applicable with trace mode on.

Trace a graph

Press the ◀ and ▶ keys to move the trace cursor along the current graph (left or right respectively). The display also shows the current coordinate position (*x*, *y*) of the cursor. Trace mode and the coordinate display are automatically set when a plot is drawn.

To move between relations

If there is more than one relation displayed, press ▲ or ▼ to move between relations.

To jump directly to a value

To jump straight to a value rather than using the Trace function, use the GOTO menu key. Press GOTO, then enter a value. Press OK to jump to the value.

To turn trace on/off

If the menu labels are not displayed, press MENU first.

- Turn off trace mode by pressing TRACE•.
- Turn on trace mode by pressing TRACE.

Zoom within a graph

One of the menu key options is ZOOM. Zooming redraws the plot on a larger or smaller scale. It is a shortcut for changing the Plot Setup.

The `Set Factors...` option enables you to set the factors by which you zoom in or zoom out, and whether the zoom is centered about the cursor.

Zoom options

Press ZOOM, select an option, and press OK. (If ZOOM is not displayed, press MENU.) Not all options are available in all apps.

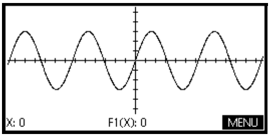
Option	Meaning
`Center on Cursor`	Re-centers the plot around the current position of the cursor *without* changing the scale.
`Box...`	Lets you draw a box to zoom in on.
`In`	Divides horizontal and vertical scales by the X-factor and Y-factor. For instance, if zoom factors are 4, then zooming in results in 1/4 as many units depicted per pixel. (see `Set Factors...`)
`Out`	Multiplies horizontal and vertical scales by the X-factor and Y-factor (see `Set Factors...`).
`X In`	Divides horizontal scale only, using X-factor.
`X Out`	Multiplies horizontal scale only, using X-factor.
`Y In`	Divides vertical scale only, using Y-factor.
`Y Out`	Multiplies vertical scale only, using Y-factor.
`Square`	Changes the vertical scale to match the horizontal scale. (Use this after doing a Box Zoom, X-Zoom, or Y-Zoom.)
`Set Factors...`	Sets the X-Zoom and Y-Zoom factors for zooming in or zooming out. Includes option to recenter the plot before zooming.

Option	Meaning (Continued)
`Auto Scale`	Rescales the vertical axis so that the display shows a representative piece of the plot, for the supplied *x* axis settings. (For Sequence and Statistics apps, autoscaling rescales both axes.) The autoscale process uses the first selected function only to determine the best scale to use.
`Decimal`	Rescales both axes so each pixel = 0.1 units. Resets default values for `XRNG` (-12.7 to 12.7) and `YRNG` (-5.5, 5.5).
`Integer`	Rescales horizontal axis only, making each pixel =1 unit.
`Trig`	Rescales horizontal axis so 1 pixel = π/24 radians or 7.58 degrees; rescales vertical axis so 1 pixel = 0.1 unit.
`Un-zoom`	Returns the display to the previous zoom, or if there has been only one zoom, un-zoom displays the graph with the original plot settings.

Zoom examples

The following screens show the effects of zooming options on a plot of $3\sin x$.

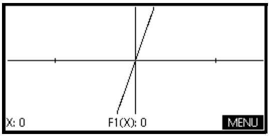
Plot of $3\sin x$



Zoom In:

MENU ZOOM `In` OK

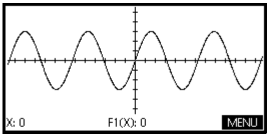
As a shortcut, press [Σ+] while in the Plot view to zoom in.



Un-zoom:

ZOOM `Un-zoom` OK

Note: press ▲ to move to the bottom of the Zoom list.

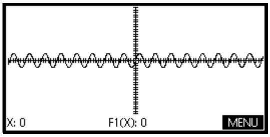


Zoom Out:

ZOOM `Out` OK

Now un-zoom.

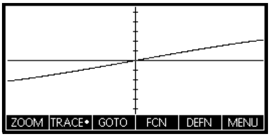
As a shortcut, press [–] while in the Plot view to zoom out.



X-Zoom In:

ZOOM `X In` OK

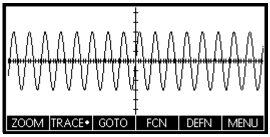
Now un-zoom.



X-Zoom Out:

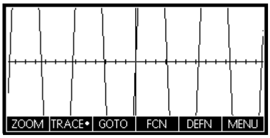
ZOOM `X Out` OK

Now un-zoom.



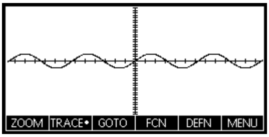
Y-Zoom In:

ZOOM `Y In` OK



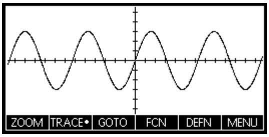
Y-Zoom Out:

ZOOM Y Out OK



Zoom Square:

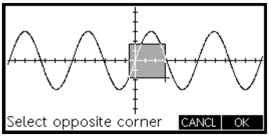
ZOOM Square OK



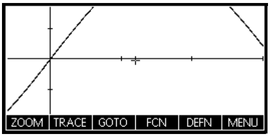
To box zoom

The Box Zoom option lets you draw a box around the area you want to zoom in on by selecting the endpoints of one diagonal of the zoom rectangle.

1. If necessary, press MENU to turn on the menu-key labels.
2. Press ZOOM and select Box...
3. Position the cursor on one corner of the rectangle. Press OK.
4. Use the cursor keys (▼, etc.) to drag to the opposite corner.



5. Press OK to zoom in on the boxed area.



To set zoom factors

1. In the Plot view, press MENU.
2. Press ZOOM.
3. Select `Set Factors...` and press OK.
4. Enter the zoom factors. There is one zoom factor for the horizontal scale (`XZOOM`) and one for the vertical scale (`YZOOM`).

   Zooming out *multiplies* the scale by the factor, so that a greater scale distance appears on the screen. Zooming in *divides* the scale by the factor, so that a shorter scale distance appears on the screen.

Views menu options

Press Views/Help, select an option, and press OK.

Option	Meaning
`Plot-Detail`	Splits the screen into the current plot and a zoom.
`Plot-Table`	Splits the screen into the plot and a numeric table.
`Auto Scale`	Rescales the vertical axis so that the display shows a representative portion of the plot, based on the current `XRNG`. For Sequence and Statistics apps, auto scale rescales both axes. The auto scale process uses the first selected function only to determine the best scale to use.
`Decimal`	Rescales both axes so each pixel = 0.1 unit. Resets default values for `XRNG`(–12.7 to 12.7) and `YRNG` (–5.5 to 5.5).
`Integer`	Rescales horizontal axis only, making each pixel=1 unit.
`Trig`	Trig Rescales horizontal axis so 1 pixel = π/48 radians or 3.75 degrees.

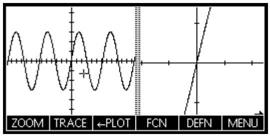
Plot-Detail

The Plot-Detail view can give you two simultaneous views of the plot.

1. Press Views. Select `Plot-Detail` and press OK.

   The graph is plotted twice. You can now zoom in on the right side.

2. Press MENU ZOOM, select the zoom method and press OK or ENTER. This zooms the right side. Here is an example of split screen with `Zoom In`.



   – The Plot menu keys are available as for the full plot (for tracing, coordinate display, equation display, and so on).

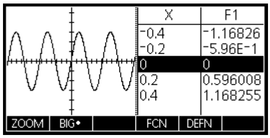
   – The <-- menu key copies the right plot to the left plot.

3. To un-split the screen, press Plot. The left side takes over the whole screen.

Plot-Table

The Plot-Table view gives you a plot view and a table view simultaneously.

1. Press Views. Select `Plot-Table` and press OK. The screen displays the plot on the left side and a table of numbers on the right side.



2. To move up and down in the table, use the ◄ and ► cursor keys. These keys move the trace point left or right along the plot, and in the table, the corresponding values are highlighted.

3. To move between functions, use the ▲ and ▼ cursor keys to move the cursor from one graph to another.

4. To return to a full Numeric (or Plot) view, press Num (or Plot).

Decimal scaling

Decimal scaling is the default scaling. If you have changed the scaling to `Trig` or `Integer`, you can change it back with `Decimal`.

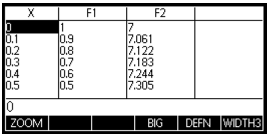
Integer scaling

Integer scaling compresses the axes so that each pixel is 1×1 and the origin is near the screen center.

Trigonometric scaling

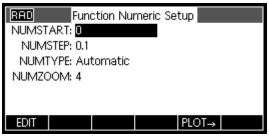
Use trigonometric scaling whenever you are plotting an expression that includes trigonometric functions. Trigonometric plots are more likely to intersect the axis at points factored by π.

About the Numeric view



After entering and selecting (check marking) the expression or expressions that you want to explore in the Symbolic view, press Num Setup to view a table of data values for the independent and dependent variables.

Setting up the table (Numeric view setup)



Press SHIFT *NUM* to define any of the table settings. Use the Numeric Setup input form to configure the table.

1. Highlight the field to edit. Use the arrow keys to move from field to field.
   - If there is a number to enter, type it in and press ENTER ANS or OK. To modify an existing number, press EDIT.
   - **Shortcut:** press the PLOT→ key to copy values from the Plot Setup into `NUMSTART` and `NUMSTEP`. Effectively, the PLOT→ menu key allows you to make the table values match the tracer values in the graph view.
2. When done, press Num Setup to view the table of numbers.

Numeric view settings

The following table details the fields on the Numeric Setup input form.

Field	Meaning
NUMSTART	The independent variable's starting value.
NUMSTEP	The size of the increment from one independent variable value to the next.
NUMTYPE	Type of numeric table: Automatic or BuildYourOwn. To build your own table, you must type each independent value into the table yourself.
NUMZOOM	Sets the zoom factor for zooming in or out on a row of the table.

Reset numeric settings

To reset the default values for all table settings, press SHIFT *CLEAR*.

Exploring the table of numbers

Num view menu keys

The following table details the menu keys that you use to work with the numerical table.

Key	Meaning
ZOOM	Displays the Zoom menu list.
BIG	Toggles between two character sizes.
DEFN	Displays the *defining* function expression for the highlighted column. To cancel this display, press DEFN•.
WIDTH3	Toggles between showing 1, 2, 3, or 4 columns of dependent variable values.

Zoom within a table

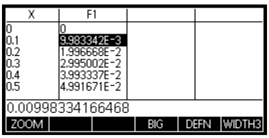
Zooming recalculates the table of numbers with greater or lesser common differences among *X*-values.

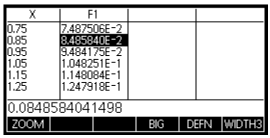
Zoom options

The following table lists the zoom options:

Option	Meaning
`In`	Decreases the step value for the independent variable so a narrower range is shown. Uses the `NUMZOOM` factor in Numeric Setup.
`Out`	Increases the step value for the independent variable so that a wider range is shown. Uses the `NUMZOOM` factor in Numeric Setup.
`Decimal`	Changes intervals for the independent variable to 0.1 units. Starts at zero (shortcut to changing `NUMSTART` and `NUMSTEP`).
`Integer`	Changes intervals for the independent variable to 1 unit. Starts at zero (shortcut to changing `NUMSTART` and `NUMSTEP`).
`Trig`	Changes intervals for independent variable to π/24 radians or 7.5 degrees. Starts at zero.
`Un-zoom`	Returns the display to the previous zoom.

The display on the right is a Zoom In of the display on the left. The `ZOOM` factor is 4.





HINT To jump to an independent variable value in the table, use the arrow keys to place the cursor in the independent variable column, then enter the value to jump to.

Automatic recalculation

You can enter any new value in the *X* column. When you press ENTER/ANS, the values for the dependent variable(s) are recalculated, and the entire table is regenerated with the same interval between *X*-values.

Building your own table of numbers

The default `NUMTYPE` is `Automatic`, which fills the table with data for regular intervals of the independent (*X*, *T*, θ, or *N*) variable. With the `NUMTYPE` option set to `Build YourOwn`, you fill the table yourself by typing in the independent-variable values you want. The dependent values are then calculated and displayed.

Build a table

1. Start with an expression defined (in Symbolic view) in the app of your choice. *Note: Function, Polar, Parametric, and Sequence apps only.*

2. In the Numeric Setup (SHIFT *NUM*), choose

   `NUMTYPE: BuildYourOwn`.

3. Open the Numeric view (Num Setup).

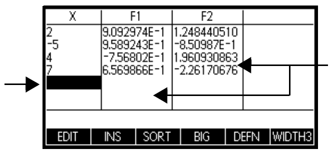
4. Clear existing data in the table (SHIFT *CLEAR*).

5. Enter the independent values in the left-hand column.

   Type in a number and press ENTER ANS . You do not have

   to enter them in order, because the SORT function can rearrange them. To insert a number between two others, use INS .

You enter numbers into the X column



F1 and F2 entries are generated automatically

Clear data

Press SHIFT *CLEAR*, OK to erase the data from a table.

BuildYourOwn table keys

Besides the BIG and DEFN menu keys, you can use the following keys to explore the table when `BuildYour Own` is active.

Key	Meaning
EDIT	Puts the highlighted independent value (*X*, *T*, θ, or *N*) into the edit line. Pressing ENTER/ANS replaces this variable with its current value.
INS	Inserts a zero value at the position of the highlight. Replace a zero by typing the number you want and pressing ENTER/ANS.
SORT	Sorts the independent variable values into ascending or descending order. Press SORT and select the ascending or descending option from the menu, and press OK.
← Clear	Deletes the highlighted row.
SHIFT *CLEAR*	Clears *all* data from the table.

Example: plotting a circle

Plot the circle, $x^2 + y^2 = 25$. First rearrange it to read $y = \pm\sqrt{25 - x^2}$.

To plot both the positive and negative *y*-values, use two equations as follows:

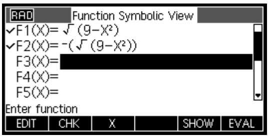
$y = \sqrt{25 - x^2}$ and $y = -\sqrt{25 - x^2}$

1. In the Function app, specify the functions.

Apps/Info *Select*

`Function` START

SHIFT √ x² (Copy 25



- X,T,θ,N x^2) ENTER

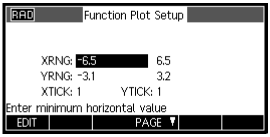
(–) SHIFT x^2 (25

- X,T,θ,N x^2) ENTER

2. Reset the graph setup to the default settings.

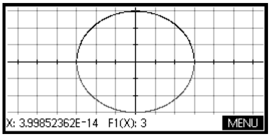
SHIFT *SETUP-PLOT*

SHIFT *CLEAR*



3. Plot the two functions.

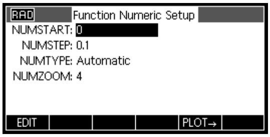
Plot Setup



4. Reset the numeric setup to the default settings.

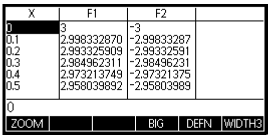
SHIFT *SETUP-NUM*

SHIFT *CLEAR*



5. Display the functions in numeric form.

Num Setup



3

Function app

About the Function app

The Function app enables you to explore up to 10 real-valued, rectangular functions y in terms of x. For example, $y = 1 - x$ and $y = (x-1)^2 - 3$.

Once you have defined a function you can:

- create graphs to find roots, intercepts, slope, signed area, and extrema
- create tables to evaluate functions at particular values

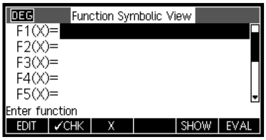
This chapter demonstrates the basic tools of the Function app by stepping you through an example.

Getting started with the Function app

Throughout this chapter, we will use an example involving two functions: a linear, $y = 1 - x$, and a quadratic, $y = (x-1)^2 - 3$.

Open the Function app

1. Open the Function app.



   Apps Info *Select*

   `Function`

   RESET OK START

   The Function app starts in the Symbolic view.

   The Symbolic view is the *defining view* for the Function app. The other views are derived from any symbolic expressions defined here.

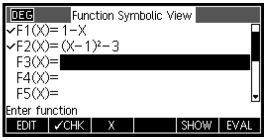
Define the expressions

There are 10 function definition fields on the Function app's Symbolic view. They are labelled `F1(X)` through `F9(X)` and `F0(X)`. Highlight the function definition field you want to use, and enter an expression. You can press EDIT to edit an existing expression or just start typing to enter a new expression. Press Clear to delete an existing expression, or SHIFT *Clear* to clear all expressions.

2. Enter the linear function in `F1(X)`.

1 [−] [X,T,θ,N] [ENTER]

3. Enter the quadratic function in `F2(X)`.



[(] [X,T,θ,N] [−]

1 [)] [x^2] [−]

3 [ENTER]

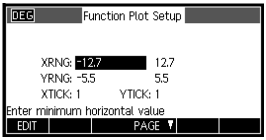
NOTE *You can use the* X *menu key to assist in the entry of equations. It has the same effect as pressing* [X,T,θ,N] .

Set up the plot

You can change the scales of the x- and y-axes and the spacing of the axis tick marks.

4. Display plot settings.

SHIFT *SETUP-PLOT*

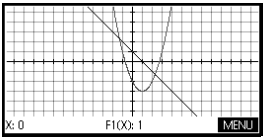


Note: for our example, you can leave the plot settings at their default values. If your settings do not match this example, press SHIFT *CLEAR to restore the default values.*

Plot the functions

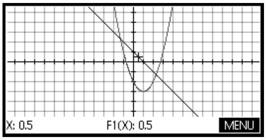
5. Plot the functions.

   [Plot Setup]



Trace a graph

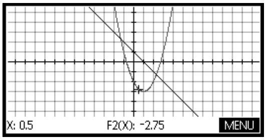
6. Trace the linear function.
   [►] or [◄]



Note: by default, the tracer is active.

7. Jump from tracing the linear function to the quadratic function.

   [▲] or [▼]



Change the scale

You can change the scale to see more or less of your graph. This can be done in four ways:

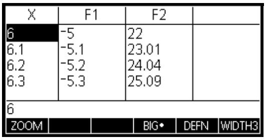
- Press [+] to zoom in or [-] to zoom out on the current cursor coordinates. This method uses the zoom factors set in the Zoom menu. The default for both x and y is 2.
- Use the Plot Setup to define `XRNG` and `YRNG` exactly as you want.
- Use the Zoom menu to zoom in or out, horizontally or vertically, or both, etc.
- Use the Views menu to select a pre-defined window.

You can also use *Autoscale*, in either the Zoom or Views menus, to choose a vertical range for the current horizontal range, based on your function definitions.

Display the Numeric view

1. Display the Numeric view.

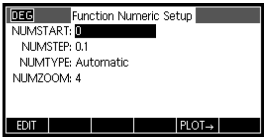
   Num Setup



Set up the table

2. Display the Numeric setup.

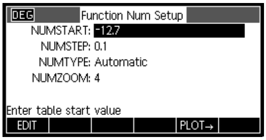
   SHIFT *SETUP-NUM*



You can set the starting value and step value for the x-column, as well as the zoom factor for zooming in or out on a row of the table. You can also choose the table type. Press SHIFT *CLEAR* to reset all values to their defaults.

3. Match the table settings to the pixel columns in the graph view.

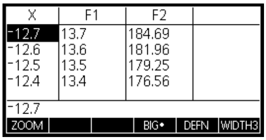
   PLOT→ OK



Explore the table

4. Display the table of values.

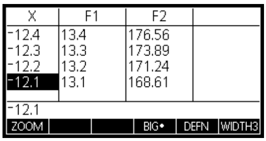
   Num Setup



To navigate around a table

5. Move to x = –12.1.

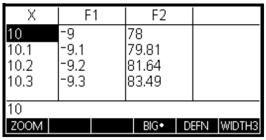
   ▼ *6 times.*



To go directly to a value

6. Move directly to X =10.

   1 0 **OK**



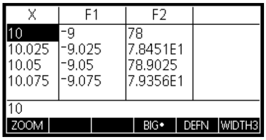
NOTE to navigate directly to a value, ensure the cursor is in the independent variable column, in this case, x, before typing the desired value.

To access the zoom options

7. Zoom in on X = 10 by a factor of 4. *Note:* `NUMZOOM` *has a setting of 4.*

   ZOOM `In`

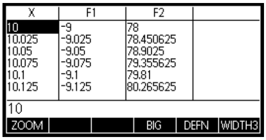
   OK



To change font size

8. Display table numbers in smaller font.

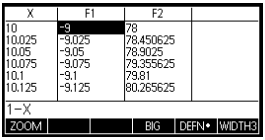
   BIG•



To display the symbolic definition of a column

9. Display the symbolic definition for the F1 column.

   ▶ **DEFN**



The symbolic definition of F1 is displayed at the bottom of the screen.

To change column width

10. Press **WIDTH4** 3 times to toggle from showing 3 function columns to showing 4, then 1, then 2.

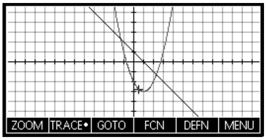
Function app interactive analysis

From the Plot view (Plot Setup), you can use the functions on the FCN menu to find roots, intersections, slopes, signed areas and extrema for a function defined in the Function app (and any Function-based apps). The FCN functions act on the currently selected graph.

Display the Plot menu

1. Display the Plot view menu.

Plot Setup MENU



To find a root of the quadratic function

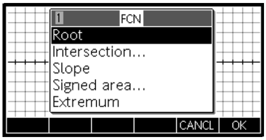
2. Move the cursor so that it is near x= 3.

▲ or ▼ to select the quadratic

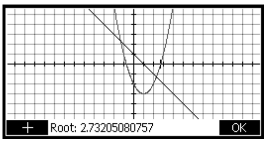
▶ or ◀ to move the cursor near $x = 3$

FCN *Select* `Root`

OK



The root value is
displayed at the bottom of the screen.

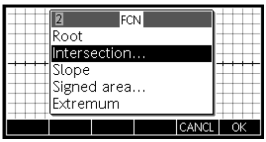


Note: if there is more than one root (as in our example), the coordinates of the root closest to the current cursor position are displayed.

To find the intersection of the two functions

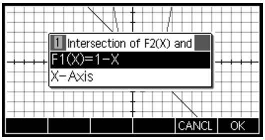
3. Find the intersection of the two functions.

MENU FCN ▼ OK



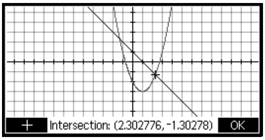
4. Choose the function whose intersection with the quadratic function you wish to find.

OK to select `F1(X)`



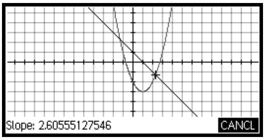
The coordinates of the intersection point are displayed at the bottom of the screen.

Note: if there is more than one intersection (as in our example), the coordinates of the intersection point closest to the current cursor position are displayed.



To find the slope of the quadratic function

5. Find the slope of the quadratic function at the intersection point.



MENU **FCN** ▼ ▼

Select `Slope`

OK

The slope value is displayed at the bottom of the screen. You can use the left- and right-cursor keys to trace along the curve and see the slope at other points. You can also use the up- and down-cursor keys to jump to another function and see the slope at points on that graph. Press **CANCL** to quit and return to the Plot view.

To find the signed area between the two functions

6. To find the area between the two functions in the range $-1.3 \le x \le 2.3$, first move the cursor to `F1(X)` and select the signed area option.

▲ or ▼ to select the linear

MENU

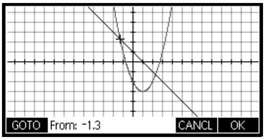
FCN ▼ ▼ ▼

Select `Signed area`

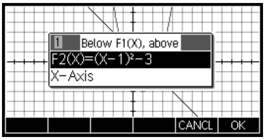
OK

7. Move the cursor to $x = -1.3$ by pressing ▶ or ◀ *to move to x = –1.3*

OK



8. Press OK to accept using `F2(X)` as the other boundary for the integral.

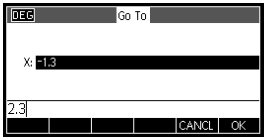


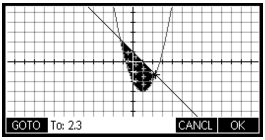
9. Choose the end value for x.

GOTO

2.3

OK



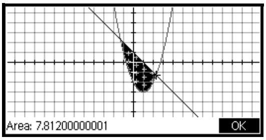


The cursor jumps to $x = 2.3$ on the linear function and the area is shaded. The shading shows "+" (plus) if the area is positive, and "-" (minus) if negative.

10. Display the numerical value of the integral.

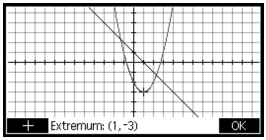
OK *to display the value*

OK *to return to the plot menu*



To find the extremum of the quadratic

11. Move the cursor to the quadratic equation and find the extremum of the quadratic.



(to move the tracer to the quadratic)

FCN

Select `Extremum`

OK

The coordinates of the extremum are displayed at the bottom of the screen.

HINT The `ROOT` and `EXTREMUM` functions return one value only even if the function has more than one root or extremum. The function finds the value closest to the position of the cursor. You need to re-locate the cursor to find other roots or extrema that may exist.

The FCN Variables

The results of the FCN functions are saved in the following variables:

- Root
- Isect
- Slope
- SignedArea
- Extremum

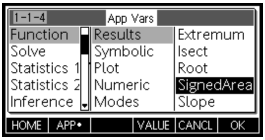
The FCN functions are:

Function	Description
Root	Select Root to find the root of the current function nearest the cursor. If no root is found, but only an extremum, then the result is labeled Extremum: instead of Root:. The cursor is moved to the root value on the x-axis and the resulting x-value is saved in a variable named Root.
Extremum	Select Extremum to find the maximum or minimum of the current function nearest the cursor. The cursor moves to the extremum and the coordinate values are displayed.The resulting value is saved in a variable named Extremum.
Slope	Select Slope to find the numeric derivative of the current function at the current position of the cursor. The result is saved in a variable named Slope.
Signed area	Select Signed area to find the numeric integral. (If there are two or more expressions checkmarked, then you will be asked to choose the second expression from a list that includes the x-axis.) Select a starting point, then move the cursor to select an ending point. The result is saved in a variable named SignedArea.
Intersection	Select Intersection to find the intersection of the graph you are currently tracing and another graph. You need to have at least two selected expressions in the Symbolic view. Finds the intersection closest to the tracer coordinates. Displays the coordinate values and moves the cursor to the intersection. The resulting x-value is saved in a variable named Isect.

To access FCN variables

The FCN variables are contained on the Vars menu.

To access FCN variables in the Home view:



Home Modes Vars Chars A

APP•

Select `Function Results`

▶ ▲ or ▼ to choose a variable

OK

You can access and use the FCN variables to define functions in the Symbolic view the same way as you do in the Home view.

4

Solve app

About the Solve app

The Solve app solves an equation or an expression for one of its *unknown variables*. You define an equation or expression in the Symbolic view, then supply values for all the variables *except one* in the Numeric view. Solve works only with real numbers.

Note the differences between an equation and an expression:

- An *equation* contains an equals sign. Its solution is a value for the unknown variable that makes both sides of the equation have the same value.
- An *expression* does not contain an equals sign. Its solution is a *root*, a value for the unknown variable that makes the expression have a value of zero.

You can use the Solve app to solve an equation for any one of its variables. In addition, if the equation or expression is a polynomial in a single variable and there is more than one solution for the variable, then ALT appears in the menu. Pressing this menu key will display a list of all real solutions for the variable.

You can solve the equation as many times as you want, using new values for the knowns and highlighting a different unknown for which to solve.

NOTE You can only have one equation checked at a time. Other apps can have multiple equations checked, but not the Solve app. Once solved, the app carries the values of solved variables into new equations, and you can solve for new variables using the recently calculated values. *It is not possible to solve for more than one variable at once. Simultaneous linear equations, for example, should be solved using the Linear Solver app, matrices or graphs in the Function app.*

Getting started with the Solve app

Suppose you want to find the acceleration needed to increase the speed of a car from 16.67 m/sec (60 kph) to 27.78 m/sec (100 kph) in a distance of 100 m.

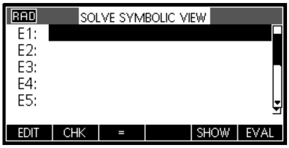
The equation to solve is:

$$V^2 = U^2 + 2AD$$

Open the Solve app

1. Open the Solve app.

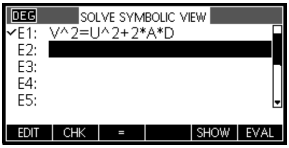
Apps Info *Select* `Solve`
START



The Solve app starts in the Symbolic view, where you specify the expression or equation to solve. You can define up to ten equations (or expressions), named `E0` to `E9`. Each equation can contain up to 27 real variables, named `A` to `Z` and `θ`.

Define the equation

2. Define the equation.



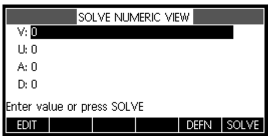
ALPHA V x² SHIFT = ALPHA U x² + 2 ×
ALPHA A × ALPHA D ENTER

Note: you can use the = *menu key to assist in the entry of equations.*

Enter known variables

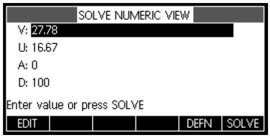
3. Display the Solve numeric view screen.

Num Setup



In the Numeric view, you specify the values of the known variables, highlight the variable that you want to solve for, and press SOLVE.

4. Enter the values for the known variables.

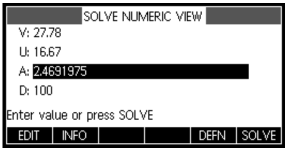


2 7 [.] 7 8 [ENTER] 1 6 [.] 6 7 [ENTER] [▼] 1 0 0 [ENTER]

Solve the unknown variable

5. Solve for the unknown variable (A).

[▼] [▼] SOLVE



Therefore, the acceleration needed to increase the speed of a car from 16.67 m/sec (60 kph) to 27.78 m/sec (100 kph) in a distance of 100 m is approximately 2.47 m/s^2.

Because the variable *A* in the equation is linear we know that we need not look for any other solutions.

Plot the equation

The Plot view shows one graph for each side of the selected equation. You can choose any of the variables to be the independent variable.

The current equation is $V^2 = U^2 + 2AD$.

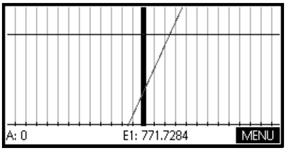
Select *A* as the variable. The plot view will now plot two equations. One of these is $Y = V^2$, with $V = 27.78$, that is, $Y = 771.7284$. This graph will be a horizontal line. The other graph will be $Y = U^2 + 2AD$, with $U = 16.67$ and $D = 100$, that is, $Y = 200A + 277.8889$. This graph is also a line. The desired solution is the value of *A* where these two lines intersect.

6. Plot the equation for variable `A`.

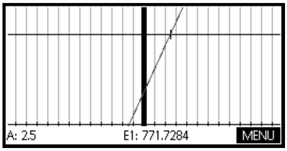
   Views Help

   Select `Auto Scale`

   OK



7. Trace along the graph representing the left side of the equation until the cursor nears the intersection.



   Note the value of `A` displayed near the bottom left corner of the screen.

   The Plot view provides a convenient way to find an approximation to a solution instead of using the Numeric view Solve option.

Solve app's Numeric view keys

The Solve app's Numeric view keys are:

Key	Meaning
EDIT	Copies the highlighted value to the edit line for editing. Press OK when done.
INFO	Displays information about the nature of the solution found.
PAGE ▼	Displays other pages of variables, if any.
ALT	If available, displays a list of multiple solutions for the selected variable.
DEFN	Displays the symbolic definition of the current expression. Press OK when done.
SOLVE	Finds a solution for the highlighted variable, based on the values of the other variables.

Key	Meaning (Continued)
Clear	Clears highlighted variable to zero *or* deletes current character in the edit line, if the edit line is active.
SHIFT *CLEAR*	Resets all variable values to zero *or* clears the edit line, if cursor is in the edit line.

Interpreting results

After Solve has returned a solution, press INFO in the Numeric view for more information. You will see one of the following three messages. Press OK to clear the message.

Message	Condition
`Zero`	The Solve app found a point where both sides of the equation were equal, or where the expression was zero (a root), within the calculator's 12-digit accuracy.
`Sign Reversal`	Solve found two points where the difference between the two sides of the equation has opposite signs, but it cannot find a point in between where the value is zero. Similarly, for an expression, where the value of the expression has different signs but is not precisely zero. This might be because either the two points are neighbours (they differ by one in the twelfth digit), or the equation is not real-valued between the two points. Solve returns the point where the value or difference is closer to zero. If the equation or expression is continuously real, this point is Solve's best approximation of an actual solution.
`Extremum`	Solve found a point where the value of the expression approximates a local minimum (for positive values) or maximum (for negative values). This point may or may not be a solution. Or: Solve stopped searching at 9.99999999999E499, the largest number the calculator can represent. Note that the value returned is probably not valid.

If Solve could not find a solution, you will see one of the following two messages.

Message	**Condition**
`Bad Guess(es)`	The initial guess lies outside the domain of the equation. Therefore, the solution was not a real number or it caused an error.
`Constant?`	The value of the equation is the same at every point sampled.

HINT It is important to check the information relating to the solve process. For example, the solution that the Solve app finds is not a solution, but the closest that the function gets to zero. Only by checking the information will you know that this is the case.

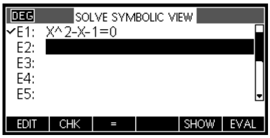
Multiple solutions

Consider the polynomial equation:

$$x^2 - x - 1 = 0$$

Since this equation is quadratic for *x*, there can be (and in this case are) two solutions. In the case of polynomials, the HP 39gII offers a quick way to find multiple solutions.

1. Select the Solve app and enter the equation.



[Apps Info]

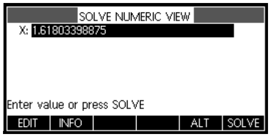
Select `Solve` START

[ALPHA] X [x^y] 2 [-]

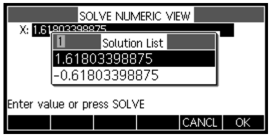
[ALPHA] X [-] 1

2. Solve for *x*.

Num Setup SOLVE



ALT appears in the menu to alert you that there are multiple solutions.



Press ALT to see the list of solutions and to select the one you want.

Using variables in equations

You can use any of the real variable names, *A* to *Z* and θ. Do not use variable names defined for other types, such as *M*1 (a matrix variable).

Home variables

All home variables (other than those for app settings, like `Xmin` and `Ytick`) are *global*, which means they are *shared* throughout the different apps of the calculator. A value that is assigned to a home variable anywhere remains with that variable wherever its name is used.

Therefore, if you have defined a value for *T* (as in the above example) in another app or even another Solve equation, that value shows up in the Numeric view for this Solve equation. When you then redefine the value for *T* in this Solve equation, that value is applied to *T* in all other contexts (until it is changed again).

This sharing allows you to work on the same problem in different places (such as Home and the Solve app) without having to update the value whenever it is recalculated.

HINT As the Solve app uses existing variable values, be sure to check for existing variable values that may affect the solve process. (You can use SHIFT *CLEAR* to reset all values to zero in the Solve app's Numeric view if you wish.)

App variables

Functions defined in other apps can also be referenced in the Solve app. For example, if you define $F1(X)=X^2+10$ in the Function app, you can enter $F1(X)=50$ in the Solve app to solve the equation $X^2+10=50$.

5

Statistics 1Var app

About the Statistics 1Var app

The Statistics 1Var app can store up to ten data sets at one time. It can perform one-variable statistical analysis of one or more sets of data.

The Statistics 1Var app starts with the Numeric view which is used to enter data. The Symbolic view is used to specify which columns contain data and which column contains frequencies.

You can also compute statistics values in Home and recall the values of specific statistics variables.

The values computed in the Statistics 1Var app are saved in variables, and many of these variables are listed by the STATS function accessible from the Statistics 1Var app's Numeric view.

Getting started with the Statistics 1Var app

The following example is about the heights of students in a classroom. We will use the example to introduce the structure and function of the Statistics 1Var app. You are measuring the heights of students in a classroom to find the mean height. The first five students have the following measurements: 160cm, 165cm, 170cm, 175cm, 180cm.

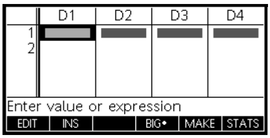
1. Open the Statistics 1Var app.

   ALPHA *Select*

   `Statistics 1Var`

   RESET OK

   START



2. Enter the measurement data.

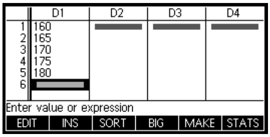
160 ENTER/ANS

165 ENTER/ANS

170 ENTER/ANS

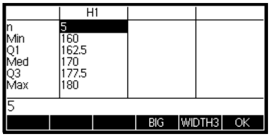
175 ENTER/ANS

180 ENTER/ANS



3. Find the mean of the sample.

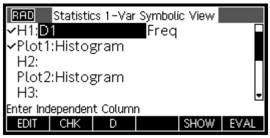
   Press STATS to see the statistics calculated from the sample data in `D1`.



   Note that the title of the column of statistics is `H1`. There are 5 data set definitions available for one-variable statistics: `H1–H5`. If data is entered in `D1`, `H1` is automatically set to use `D1` for data, and the frequency of each data point is set to 1. You can select other columns of data from the Symbolic view of the app.

4. Press OK to close the statistics window.

   Press Symb/Setup to see the data set definitions.



   The first column indicates the associated column of data for each data set definition, and the second column indicates the constant frequency, or the column that holds the frequencies.

Statistics 1Var app's Symb View keys

The keys you can use from this window are:

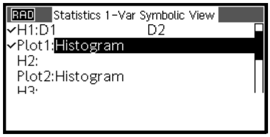
Key	Meaning
EDIT	Copies the column variable (or variable expression) to the edit line for editing. Press OK when done.
CHK	Checks/unchecks the current data set. Only the checkmarked data set(s) are computed and plotted.
D	Typing aid for the column names.
SHOW	Displays the current expression is Textbook Format. Press OK when done.
EVAL	Evaluates the highlighted expression, resolving any references to function expressions.
Vars Chars A	Displays the menu for entering variable names or contents of variables.
Math Cmds B	Displays the menu for entering math operations.
Clear	Deletes the highlighted variable *or* the character to the left of the cursor in the edit line.
SHIFT *CLEAR*	Resets default specifications for the data sets *or* clears the edit line (if it was active).

To continue our example, suppose that the heights of the rest of the students in the class are measured, but each one is rounded to the nearest of the five values first recorded. Instead of entering all the new data in D1, we shall simply add another column, D2, that holds the frequencies of our five data points in D1.

Height (cm)	Frequency
160	5
165	3
170	8
175	2
180	1

5. Move the highlight bar into the right column of the H1 definition and enter the column variable name D2.

   D 2



6. Return to the numeric view.

   Num Setup

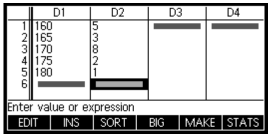
7. Enter the frequency data shown in the above table.

   ▶ 5 ENTER ANS 3 ENTER ANS

   8 ENTER ANS

   2 ENTER ANS

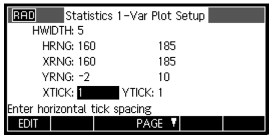
   1 ENTER ANS



8. Display the computed statistics.

   STATS

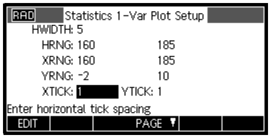
   The mean height is approximately 167.63cm.



9. Setup a histogram plot for the data.

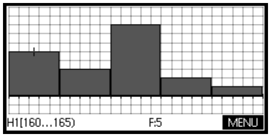
OK SHIFT *SETUP-PLOT*

Enter set up information appropriate to your data.



10. Plot a histogram of the data.

Plot Setup



Entering and editing statistical data

The Numeric view (Num Setup) is used to enter data into the Statistics 1Var app. Each column represents a variable named `D0` to `D9`. After entering the data, you must define the data set in the Symbolic view (Symb Setup).

HINT A data column must have at least two data points for one-variable statistics.

You can also store statistical data values by copying lists from Home into statistics data columns. For example, in Home, `L1` STO ▸ `D1` stores a copy of the list `L1` into the data-column variable `D1`.

Statistics 1Var app's Num View keys

The Statistics 1Var app's Numeric view keys are:

Key	Meaning
EDIT	Copies the highlighted item into the edit line.
INS	Inserts a zero value above the highlighted cell.

Key	Meaning (Continued)
SORT	Sorts the specified *independent* data column in ascending or descending order, and rearranges a specified dependent (or frequency) data column accordingly.
BIG	Switches between larger and smaller font sizes.
MAKE	Opens a dialog box for creating a sequence based on an expression and storing it in a data column.
STATS	Computes descriptive statistics for each data set specified in the Symbolic view.
Clear	Deletes the currently highlighted value.
SHIFT *CLEAR*	Clears the current column or all columns of data. Press SHIFT *CLEAR* to display a menu list, then select the current column or all columns option, and press OK.
SHIFT *CURSOR KEY*	Moves to the first or last row, or first or last column.

Save data

The data that you enter is automatically saved. When you are finished entering data values, you can press a key for another Statistics view (like Symb Setup), or you can switch to another app or Home.

Edit a data set

In the Numeric view of the Statistics 1Var app, highlight the data value to change. Type a new value and press ENTER/ANS, or press EDIT to copy the value to the edit line for modification. Press ENTER/ANS after modifying the value on the edit line.

Delete data

- To delete a single data item, highlight it and press Clear. The values below the deleted cell will scroll up one row.
- To delete a column of data, highlight an entry in that column and press SHIFT *CLEAR*. Select the column name and press OK.
- To delete all columns of data, press SHIFT *CLEAR*. Select `All columns` and press OK.

Insert data

Highlight the entry *following* the point of insertion. Press INS, then enter a number. It will write over the zero that was inserted.

Sort data values

1. In Numeric view, highlight the column you want to sort, and press SORT.
2. Specify the Sort Order. You can choose either `Ascending` or `Descending`.
3. Specify the `INDEPENDENT` and `DEPENDENT` data columns. Sorting is by the *independent* column. For instance, if Age is `D1` and Income is `D2` and you want to sort by Income, then you make `D2` the independent column for the sorting and `D1` the dependent column.
   - To sort just one column, choose None for the dependent column.
   - For one-variable statistics with two data columns, specify the frequency column in the `Frequency` field.
4. Press OK.

Computed statistics

Pressing STATS displays the results in the following table.

Statistic	Definition
n	Number of data points.
Min	Minimum data value in data set.
Q1	First quartile: median of values to left of median.
Med	Median value of data set.
Q3	Third quartile: median of values to right of median.
Max	Maximum data value in data set.
ΣX	Sum of data values (with their frequencies).
ΣX^2	Sum of the squares of the data values.
$\bar{x}$	Mean of the data values.
sX	Sample standard deviation of the data set.
σX	Population standard deviation of the data set.
seX	Standard error of the data set.

When the data set contains an odd number of values, the data set's median value is not used when calculating Q1 and Q3 in the table above. For example, for the following data set:

{3,5,7,8,15,16,17}

only the first three items, 3, 5, and 7 are used to calculate Q1, and only the last three terms, 15, 16, and 17 are used to calculate Q3.

Plotting

You can plot:

- Histograms
- Box-and-Whisker plots
- Normal Probability plots
- Line plots
- Bar graphs
- Pareto charts

Once you have entered your data and defined your data set, you can plot your data. You can plot up to five box-and-whisker plots at a time; however, with the other types, you can only plot one of them at a time.

To plot statistical data

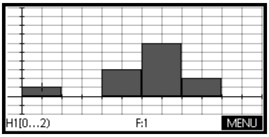
1. In the Symbolic view (Symb Setup), select (CHK) the data sets you want to plot.
2. Select the plot type. Highlight the Plot field for your data set, press the CHOOS menu key, and scroll to the plot type you want. Press the OK menu key when you have made your choice.
3. For any plot, but especially for a histogram, adjust the plotting scale and range in the Plot Setup view. If you find histogram bars too fat or too thin, you can adjust them by changing the `HWIDTH` setting.
4. Press Plot Setup . If you have not adjusted the Plot Setup yourself, you can try Views Help *select* `Auto Scale` OK .

`AutoScale` can be relied upon to give a good starting scale which can then be adjusted in the Plot Setup view.

Plot types

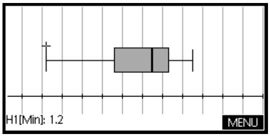
Histogram

The numbers below the plot mean that the current bar (where the cursor is) starts at 0 and ends at 2 (not including 2), and the frequency for this column, (that is, the number of data elements that fall between 0 and 2) is 1. You can see information about the next bar by pressing ▶.



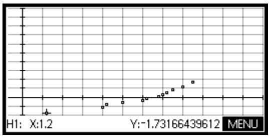
Box-and-Whisker plot

The left whisker marks the minimum data value. The box marks the first quartile, the median (where the cursor is), and the third quartile. The right whisker marks the maximum data value. The numbers below the plot mean that this column has a minimum of 1.2.



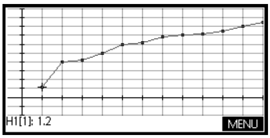
normal probability plot

The normal probability plot is used to determine whether or not sample data is more or less normally distributed. The more linear the data appear, the more likely that the data is normally distributed.



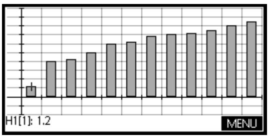
Line plot

The line plot connects points of the form (*x*, *y*), where *x* is the row number of the data point and *y* is the value of the data point.



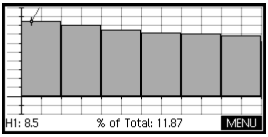
Bar graph

The bar graph shows the value of a data point as a vertical bar placed along the x-axis at the row number of the data point.



Pareto chart

A pareto chart places the data in descending order and displays each with its percentage of the whole.



Setting up the plot (Plot Setup view)

The Plot Setup (SHIFT *SETUP-PLOT*) sets most of the same plotting parameters as it does for the other built-in HP Apps. Settings unique to the Statistics 1Var app are as follows:

Histogram width

`HWIDTH` enables you to specify the width of a histogram bar. This determines how many bars will fit in the display, as well as how the data is distributed (how many values each bar represents).

Histogram range

`HRNG` enables you to specify the range of values for a set of histogram bars. The range runs from the left edge of the leftmost bar to the right edge of the rightmost bar. You can limit the range to exclude any values you suspect are outliers.

Exploring the graph

The Plot view has menu keys for zooming, tracing, and coordinate display. There are also scaling options under Views Help.

Statistics 1Var app's Plot View keys

The Plot view keys are:

Key	Meaning
SHIFT *CLEAR*	Erases the plot.
Views Help	Offers additional pre-defined views for splitting the screen and autoscaling the axes.
SHIFT ◄ SHIFT ►	Moves cursor to far left or far right.
ZOOM	Displays the Zoom menu.
TRACE	Turns trace mode on/off. The white box appears next to the option when Trace mode is active.
DEFN	Displays the definition of the current statistical plot.
MENU	Toggles the menu off and on.

6

Statistics 2Var app

About the Statistics 2Var app

The Statistics 2Var app can store up to ten data sets at one time. It can perform two-variable statistical analysis of one or more sets of data.

The Statistics 2Var app starts with the Numeric view which is used to enter data. The Symbolic view is used to specify which columns contain data and which column contains frequencies.

You can also compute statistics values in Home and recall the values of specific statistics variables.

The values computed in the Statistics 2Var app are saved in variables, and many of these variables are listed by the STATS function accessible from the Statistics 2Var app's Numeric view.

Getting started with the Statistics 2Var app

The following example is based on the advertising and sales data in the table below. In the example, you will enter the data, compute summary statistics, fit a curve to the data, and predict the effect of more advertising on sales.

Advertising minutes (independent, x)	**Resulting Sales ($) (dependent, y)**
2	1400
1	920
3	1100
5	2265
5	2890
4	2200

Open the Statistics 2Var app

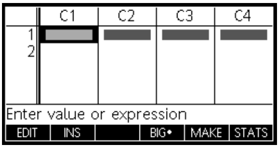
1. Clear existing data and open the Statistics 2Var app.

   [Apps Info]

   Select `Statistics 2Var`

   RESET OK

   START



   The Statistics 2Var app starts in the Numeric view.

Enter data

2. Enter the data into the columns.

   2 [ENTER ANS] 1 [ENTER ANS]

   3 [ENTER ANS] 5 [ENTER ANS]

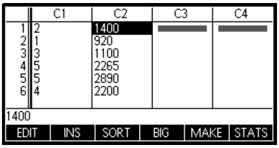
   5 [ENTER ANS] 4 [ENTER ANS]

   [▶] *to move to the next column*

   1400 [ENTER ANS] 920 [ENTER ANS]

   1100 [ENTER ANS] 2265 [ENTER ANS]

   2890 [ENTER ANS] 2200 [ENTER ANS]

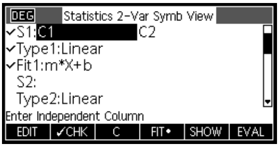


Choose data columns and fit

3. Specify the columns that hold the data you want to analyze.

   [Symb Setup]

   You could have entered your data into columns other than `C1` and `C2`.

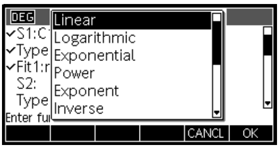


4. Select a fit.

   [▼] CHOOS

   Select `Linear`

   OK

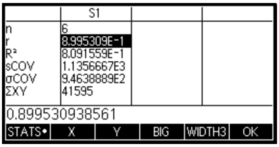


You can create up to five explorations of two-variable data, named `S1` to `S5`. In this example, we will create just one: `S1`.

Explore statistics

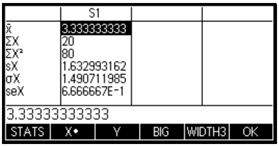
5. Find the correlation, r, between advertising time and sales.

   Num Setup STATS

   The correlation is *r*=0.8995…

6. Find the mean advertising time ($\bar{x}$) and the mean sales ($\bar{y}$).

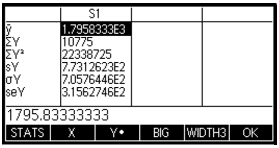
   X

   The mean advertising time, $\bar{x}$, is approximately 3.3 minutes.

   Y

   The mean sales, $\bar{y}$, is approximately $1,796.

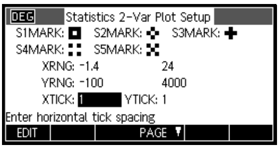
   OK

Setup plot

7. Change the plotting range to ensure all the data points are plotted (and select a different point mark, if you wish).

   SHIFT *SETUP-PLOT*

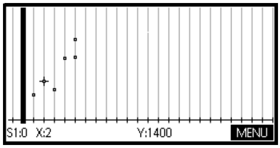
   ▼

   (−) ABS ; 100 ENTER ANS

   4000 ENTER ANS

Plot the graph

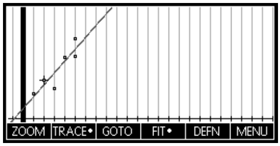
8. Plot the graph.



Draw the regression curve

9. Draw the regression curve (a curve to fit the data points).

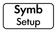
MENU FIT•

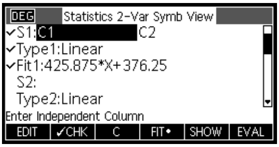


This draws the regression line for the best linear fit.

Display the equation

10. Return to the Symbolic view.





The slope (*m*) is 425.875. The *y*-intercept (*b*) is 376.25.

Predict values

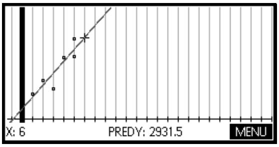
Predict the sales figure if advertising were to go up to 6 minutes.

11. Return to the Plot view.

12. Trace to x=6 on the linear fit.

▽ to move the tracer to the fit

▷ 40 times to find x=6



The model predicts that sales would rise to $2,931.50 if advertising were increased to 6 minutes.

Entering and editing statistical data

The Numeric view (Num Setup) is used to enter data into the Statistics 2Var app. Each column represents a variable named `C0` to `C9`. After entering the data, you must define the data set in the Symbolic view (Symb Setup).

HINT A data column must have at least four data points to provide valid two-variable statistics.

You can also store statistical data values by copying lists from Home into Statistics data columns. For example, in Home, `L1` STO ▸ `C1` stores a copy of the list `L1` into the data-column variable `C1`.

Statistics 2Var app’s NUM view keys

The Statistics 2Var app’s Numeric view keys are:

Key	Meaning
EDIT	Copies the highlighted item into the edit line.
INS	Inserts a zero value above the highlighted cell.

Key	Meaning (Continued)
SORT	Sorts the specified *independent* data column in ascending or descending order, and rearranges a specified dependent (or frequency) data column accordingly.
BIG•	Switches between larger and smaller font sizes.
MAKE	Opens a dialog box to create a column of data based on an expression.
STATS	Computes descriptive statistics for each data set specified in the Symbolic view.
Clear	Deletes the currently highlighted value.
SHIFT *CLEAR*	Clears the current column or all columns of data. Press SHIFT *CLEAR* to display a menu list, then select the current column or all columns option, and press OK.
SHIFT *CURSOR KEY*	Moves to the first or last row, or first or last column.

Save data

The data that you enter is automatically saved. When you are finished entering data values, you can press a key for another Statistics view (like Symb Setup), or you can switch to another app or Home.

Edit a data set

In the Numeric view of the Statistics 2Var app, highlight the data value to change. Type a new value and press ENTER ANS, or press EDIT to copy the value to the edit line for modification. Press ENTER ANS after modifying the value on the edit line.

Delete data

- To delete a single data item, highlight it and press [Clear]. The values below the deleted cell will scroll up one row.
- To delete a column of data, highlight an entry in that column and press [SHIFT] *CLEAR*. Select the column name.
- To delete all columns of data, press [SHIFT] *CLEAR*. Select `All columns`.

Insert data

Highlight the entry *following* the point of insertion. Press INS, then enter a number. It will write over the zero that was inserted.

Sort data values

1. In Numeric view, highlight the column you want to sort, and press SORT.
2. Specify the Sort Order. You can choose either `Ascending` or `Descending`.
3. Specify the `INDEPENDENT`, `DEPENDENT`, and (if applicable) the `FREQUENCY` data columns. Sorting is by the *independent* column. For instance, if Age is `C1` and Income is `C2` and you want to sort by Income, then you make `C2` the independent column for the sorting and `C1` the dependent column.
   - To sort just one column, choose None for the dependent column.
   - For one-variable statistics with two data columns, specify the frequency column as the dependent column.
4. Press OK.

Defining a regression model

The Symbolic view includes an expression (Fit1 through Fit5) that defines the regression model, or "fit", to use for the regression analysis of each two-variable data set.

There are three ways to select a regression model:

- Accept the default option to fit the data to a straight line.
- Select one of the available fit options in the Symbolic view.
- Enter your own mathematical expression in the Symbolic view. This expression will be plotted, *but it will not be fitted to the data points*.

Angle Setting

You can ignore the angle measurement mode *unless* your Fit definition (in the Symbolic view) involves a trigonometric function. In this case, you should specify in the Symbolic setup whether the trigonometric units are to be interpreted as degrees or radians.

Choose the fit

1. Press Symb/Setup to display the Symbolic view. Highlight the *Type* number (`Type1` through `Type5`) you want to define.
2. Press CHOOS and select from the list. Press OK when done. The regression formula for the fit is displayed in the Symbolic view.

Fit models

Eleven fit models are available:

Fit model	Meaning
`Linear`	(Default.) Fits the data to a straight line, $y = mx+b$. Uses a least-squares fit.
`Logarithmic`	Fits to a logarithmic curve, $y = m \ln x + b$.
`Exponential`	Fits to an exponential curve, $y = be^{mx}$.
`Power`	Fits to a power curve, $y = bx^m$.
`Exponent`	Fits to an exponent curve, $y = ab^x$.
`Inverse`	Fits to an inverse variation, $y = \frac{m}{x+b}$

Fit model	Meaning (Continued)
`Logistic`	Fits to a logistic curve, $y = \frac{L}{1 + ae^{(-bx)}}$ where *L* is the saturation value for growth. You can store a positive real value in *L*, or—if *L*=0—let *L* be computed automatically.
`Quadratic`	Fits to a quadratic curve, $y = ax^2+bx+c$. Needs at least three points.
`Cubic`	Fits to a cubic polynomial, $y = ax^3 + b^2x + cx + d$
`Quartic`	Fits to a quartic polynomial, $y = ax^4 + bx^3 + cx^2 + dx + e$
`Trigonometric`	Fits to a trigonometric curve, $y = a \cdot \sin(bx + c) + d$. Needs at least three points.
`User Defined`	Define your own expression (in the Symbolic view.)

To define your own fit

1. Display the Symbolic view.
2. Highlight the Fit expression (`Fit1`, etc.) for the desired data set.
3. Type in an expression and press ENTER (ANS). The independent variable must be X, and the expression must not contain any unknown variables. Example: $1.5 \times \cos x + 0.3 \times \sin x$.

Computed statistics

When you press STATS, there are three sets of statistics available. By default, the statistics involving both the independent and dependent columns are shown. Press X to see the statistics involving just the independent column or Y to display the statistics derived from the dependent column. Press STATS to return to the default view. The tables below describe the statistics displayed in each view.

Here are the statistics computed when you press STATS.

Statistic	Definition
n	The number of data points.
r	Correlation coefficient of the independent and dependent data columns, based only on the linear fit (regardless o the fit type chosen). Returns a value from -1 to 1, where 1 and -1 indicate best fits.
R^2	The coefficient of determination, which is the square of the correlation coefficient. The value of this statistics is dependent on the Fit type chosen.
sCOV	Sample covariance of independent and dependent data columns.
σ COV	Population covariance of independent and dependent data columns.
ΣXY	Sum of *xy* products.

Here are the statistics displayed when you press X.

Statistic	Definition
$\bar{x}$	Mean of *x*- (independent) values.
ΣX	Sum of *x*-values.
ΣX^2	Sum of x^2-values.
sX	The sample standard deviation of the independent column.
σX	The population standard deviation of the independent column.
serrX	the standard error of the independent column

Here are the statistics displayed when you press Y.

Statistic	Definition
$\bar{y}$	Mean of *y*- (dependent) values.
ΣY	Sum of *y*-values.
$ΣY^2$	Sum of y^2-values.
sY	The sample standard deviation of the dependent column.
σY	The population standard deviation of the dependent column.
serrY	The standard error of the dependent column.

Plotting

Once you have entered your data (Num Setup), defined your data set and your fit model (Symb Setup), you can plot your data. You can plot up to five scatter plots at a time.

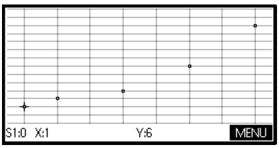
To plot statistical data

1. In Symbolic view (Symb Setup), select (CHECK) the data sets you want to plot.
2. Adjust the plotting scale and range in the Plot Setup view.
3. Press Plot Setup. If you have not adjusted the Plot Setup yourself, you can try Views Help *select* `Auto Scale` OK.

Auto Scale can be relied upon to give a good starting scale which can then be adjusted in the Plot Setup.

Tracing a Scatter Plot

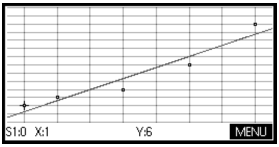
The numbers below the plot indicate that the cursor is at the first data point for `S1`, at (1, 6). Press ▶ to move to the next data point and display information about it.



Fitting a curve

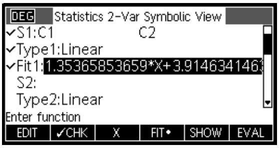
Press MENU FIT•. The graph of the fit will be displayed with the scatter plot. Press ▼ to move the tracer to the graph of the fit. Press ▶ and ◀ to trace along the fit and DEFN to see the equation of the fit.

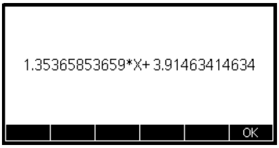
Press Symb/Setup to see the equation of the fit in the Fit1 field. To see the full equation, highlight the fit equation and press



SHOW

The expression in `Fit2` shows the slope (m=1.98082191781) and the *y*-intercept (b=2.26575).





Correlation Coefficient, r

The correlation coefficient is stored in the variable *r*. It is a measure of fit to a *linear* curve only. Regardless of the fit model you have chosen, *r* relates to the linear model. The value of r can range from -1 to 1, where -1 and 1 indicate best fits.

Coefficient of Determination, R^2

The Coefficient of Determination is a measure of the goodness of fit of your model, regardless of whether that model is linear or not. A measure of 1 indicates a perfect fit.

HINT In order to access the *r* and R^2 variables after you plot a data set, you must press Num Setup to access the numeric view and then STATS to display the correlation values. The values are stored in the variables when you access the Numeric view Stats page.

Plot setup

The Plot Setup (SHIFT *SETUP-PLOT*) sets most of the same plotting parameters as it does for the other built-in apps; in addition, it has one unique setting:

Plotting mark

`S1MARK` through `S5MARK` enables you to specify one of five symbols to use to plot each data set. Press CHOOS to change the highlighted setting.

Trouble-shooting a plot

If you have problems plotting, check that you have the following:

- The correct fit (regression model).
- Only the data sets to compute or plot are checkmarked (Symbolic view).
- The correct plotting range. Try using Views Auto Scale (instead of Plot Setup), *or* adjust the plotting parameters (in Plot Setup) for the ranges of the axes.
- Ensure that both paired columns contain data, and that they are the same length.
- Ensure that a paired column of frequency values is the same length as the data column to which it refers.

Exploring the graph

The Plot view has menu keys for zooming, tracing, and coordinate display. There are also scaling options under Views.

Statistics 2Var app's Plot view keys

Key	Meaning
SHIFT *CLEAR*	Erases the plot.
Views Help	Offers additional pre-defined views for splitting the screen and auto-scaling the axes.
SHIFT ◀ SHIFT ▶	Moves cursor to far left or far right.
ZOOM	Displays the Zoom menu.
TRACE•	Turns trace mode on/off. The white dot appears next to the option when Trace mode is active.
FIT•	Turns fit mode on or off. Turning FIT• on draws a curve to fit the data points according to the current regression model.
GOTO	Enables you to specify a value on the line of best fit to jump to or a data point number to jump to.
DEFN	Displays the equation of the regression curve or the definition of the current statistical plot.
MENU	Hides and displays the menu key labels.

Calculating predicted values

The functions PREDX and PREDY estimate (predict) values for X or Y given a hypothetical value for the other. The estimation is made based on the equation that has been calculated to fit the data according to the specified fit.

Find predicted values

1. In the Plot view, draw the regression curve for the data set.
2. Press ⊙ to move to the regression curve.
3. Press GOTO and enter the value of X. The cursor jumps to the specified point on the curve and the coordinate display shows X and the predicted value of Y.

In the Home view:

- Enter PREDX(*y-value*) ENTER/ANS to find the predicted value for the independent variable given a hypothetical dependent value.
- Enter PREDY(*x-value*) to find the predicted value of the dependent variable given a hypothetical independent variable.

You can type PREDX and PREDY into the edit line, or you can copy these function names from the Commands menu under the Apps, Statistics 2Var category.

HINT In cases where more than one fit curve is displayed, the PREDX and PREDY functions use the first active fit defined in the Symbolic view.

7

Inference app

About the Inference app

The Inference app's capabilities include calculation of confidence intervals and hypothesis tests based on the Normal Z-distribution or Student's t-distribution.

Based on statistics from one or two samples, you can test hypotheses and find confidence intervals for the following quantities:

- mean
- proportion
- difference between two means
- difference between two proportions

Example data

When you first access an input form for an Inference test, by default, the input form contains example data. This example data is designed to return meaningful results that relate to the test. It is useful for gaining an understanding of what the test does, and for demonstrating the test. The calculator's on-line help provides a description of what the example data represents.

Getting started with the Inference app

This example describes the Inference app's options and functionality by stepping you through an example using the example data for the Z-Test on 1 mean.

Open the Inference app

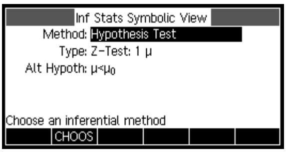
1. Open the Inference app.

   Apps Info

   Select `Inference`

   RESET OK START.

   The Inference app opens in the Symbolic view.



Inference app's Symbolic view options

The table below summarizes the options available in Symbolic view.

Hypothesis Tests	Confidence Intervals
Z-Test: 1 μ, the Z-Test on 1 mean	Z-Int: 1 μ, the confidence interval for 1 mean, based on the Normal distribution
Z-Test: $\mu_1 - \mu_2$, the Z-Test on the difference of two means	Z-Int: $\mu_1 - \mu_2$, the confidence interval for the difference of two means, based on the Normal distribution
Z-Test: 1 p, the Z-Test on 1 proportion	Z-Int: 1 p, the confidence interval for 1 proportion, based on the Normal distribution
Z-Test: $p_1 - p_2$, the Z-Test on the difference of two proportions	Z-Int: $p_1 - p_2$, the confidence interval for the difference of two proportions, based on the Normal distribution
T-Test: 1 μ, the T-Test on 1 mean	T-Int: 1 μ, the confidence interval for 1 mean, based on the Student's t-distribution
T-Test: $\mu_1 - \mu_2$, the T-Test on the difference of two means	T-Int: $\mu_1 - \mu_2$, the confidence interval for the difference of two means, based on the Student's t-distribution

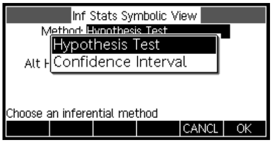
If you choose one of the hypothesis tests, you can choose the alternative hypothesis to test against the null hypothesis. For each test, there are three possible choices for an alternative hypothesis based on a quantitative comparison of two quantities. The null hypothesis is always that the two quantities are equal. Thus, the alternative hypotheses cover the various cases for the two quantities being unequal: $<$, $>$, and $\neq$.

In this section, we will use the example data for the Z-Test on 1 mean to illustrate how the app works and what features the various views present.

Select the inferential method

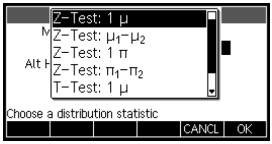
2. Select the `Hypothesis Test` inferential method.

   CHOOS
   Select `Hypothesis Test`
   OK



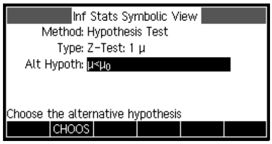
3. Define the type of test.

   CHOOS
   `Z-Test: 1 μ`
   OK



4. Select an alternative hypothesis.

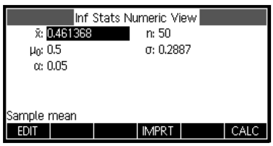
   CHOOS
   $\mu < \mu_0$
   OK



Enter data

5. Go to the Numeric view to see the default data.

   Num Setup



The table below lists the fields in this view for our current `Z-Test: 1 μ` example.

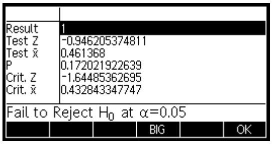
Field name	**Definition**
$\bar{x}$	Sample mean

Field name	Definition (Continued)
n	Sample size
μ_0	Assumed population mean
σ	Population standard deviation
α	Alpha level for the test

Display test results

6. Display the test results in numeric format.

CALC BIG

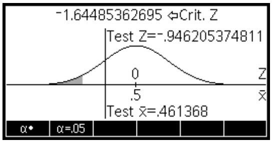


The test distribution value and its associated probability are displayed, along with the critical value(s) of the test and the associated critical value(s) of the statistic.

Plot test results

7. Display a graphical view of the test results.

OK Plot Setup



The graph of the distribution is displayed, with the test Z-value marked. The corresponding X-value is also shown, as well as the critical Z-value. Press the α menu key to see the critical Z-value as well. With the menu key active, you can use the left- and right-cursor keys to decrease and increase the α-level.

Importing sample statistics

The Inference app supports the calculation of confidence intervals and the testing of hypotheses based on data in the Statistics 1Var and Statistics 2Var apps. Computed statistics for a sample of data in a column in any Statistics-based app can be imported for use in the Inference app. The following example illustrates the process.

A calculator produces the following 6 random numbers:

0.529, 0.295, 0.952, 0.259, 0.925, and 0.592

Open the Statistics 1Var app

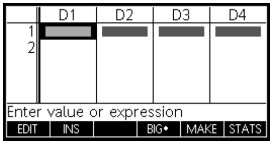
1. Open the Statistics 1Var app and reset the current settings.

   [Apps Info] *Select*

   `Statistics 1Var`

   RESET OK

   START



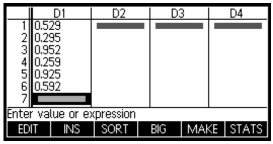
   The Statistics app opens in the Numeric view.

Enter data

2. In the D1 column, enter the random numbers produced by the calculator.

   [.] 529 [ENTER]

   [.] 295 [ENTER]



   [.] 952 [ENTER] [.] 259 [ENTER]

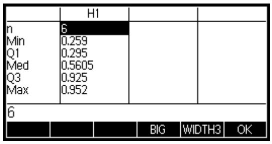
   [.] 925 [ENTER] [.] 592 [ENTER]

HINT If the Decimal Mark setting in the Modes input form ([SHIFT] `modes`) is set to Comma, use [Mem , o] instead of [.] .

Calculate statistics

3. Calculate statistics.

   STATS



   The mean of 0.592 seems a little large compared to the expected value of 0.5. To see if the difference is statistically significant, we will use the statistics computed here to construct a confidence interval for the true mean of the population of random numbers and see whether or not this interval contains 0.5.

4. Press OK to close the computed statistics window.

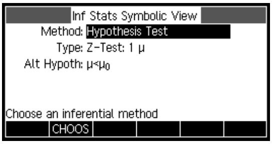
Open the Inference app

5. Open the Inference app and clear current settings.

   Apps Info *Select* `Inference`

   RESET OK

   START



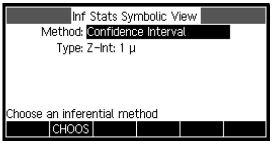
Select inference method and type

6. Select an inference method.

   CHOOS

   Select `CONF INTERVAL`

   OK

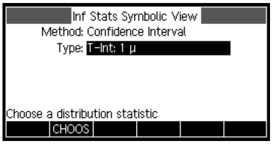


7. Select a distribution statistic type.

   ▼ CHOOS

   Select `T-Int: 1 μ`

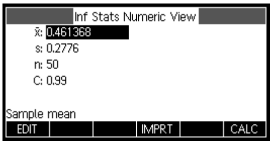
   OK



Set up the interval calculation

8. Set up the interval calculation. Note: The default values are derived from sample data from the on-line help example.

   Num Setup

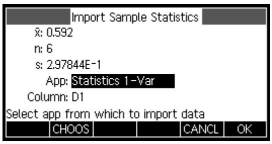


Import the data

9. Import the data from the Statistics app. Note: The data from D1 is displayed by default.

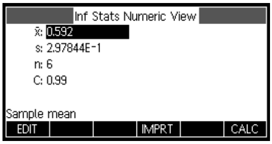
   IMPRT

   Use the App field to select the Statistics-based app from which you want to import data.



   Use the Column field to choose the column in that app where the data is stored. You can view the data before you import it. Press OK to import the statistics into the inference app.

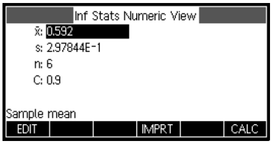
OK



10. Specify a 90% confidence interval in the C field.

▼ ▼ ▼ *to move to the C field*

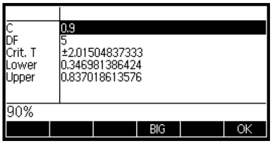
0.9 ENTER ANS



Display results numerically

11. Display the confidence interval in the Numeric view.

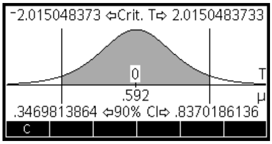
CALC BIG



Display results graphically

12. Display the confidence interval in the Plot view.

OK Plot Setup



You can see that the mean is contained within the 90% confidence interval (CI) of 0.3469814 to 0.8370186.

Hypothesis tests

You use hypothesis tests to test the validity of hypotheses that relate to the statistical parameters of one or two populations. The tests are based on statistics of samples of the populations.

The HP 39gII hypothesis tests use the Normal Z-distribution or Student's t-distribution to calculate probabilities.

One-Sample Z-Test

Menu name

Z-Test: 1 μ

On the basis of statistics from a single sample, the One-Sample Z-Test measures the strength of the evidence for a selected hypothesis against the null hypothesis. The null hypothesis is that the population mean equals a specified value, H_0: $\mu = \mu_0$.

You select one of the following alternative hypotheses against which to test the null hypothesis:

$$H_1 : \mu < \mu_0$$

$$H_1 : \mu > \mu_0$$

$$H_1 : \mu \neq \mu_0$$

Inputs

The inputs are:

Field name	Definition
$\bar{x}$	Sample mean.
n	Sample size.
μ_0	Hypothetical population mean.
σ	Population standard deviation.
α	Significance level.

Results

The results are:

Result	Description
Test Z	Z-test statistic.
Test $\bar{x}$	Value of $\bar{x}$ associated with the test Z-value.
P	Probability associated with the Z-Test statistic.
Critical Z	Boundary11 value(s) of Z associated with the α level that you supplied.

Result	Description
Critical $\bar{x}$	Boundary value(s) of $\bar{x}$ required by the α value that you supplied.

Two-Sample Z-Test

Menu name

Z-Test: $\mu_1 - \mu_2$

On the basis of two samples, each from a separate population, this test measures the strength of the evidence for a selected hypothesis against the null hypothesis. The null hypothesis is that the mean of the two populations are equal, H_0: $\mu_1 = \mu_2$.

You select one of the following alternative hypotheses to test against the null hypothesis:

$$H_1:\mu_1 < \mu_2$$

$$H_1:\mu_1 > \mu_2$$

$$H_1:\mu_1 \neq \mu_2$$

Inputs

The inputs are:

Field name	Definition
$\bar{x}_1$	Sample 1 mean.
$\bar{x}_2$	Sample 2 mean.
n_1	Sample 1 size.
n_2	Sample 2 size.
σ_1	Population 1 standard deviation.
σ_2	Population 2 standard deviation.
α	Significance level.

Results

The results are:

Result	Description
Test Z	Z-Test statistic.

Result	Description
Test $\Delta\ \bar{x}$	Difference in the means associated with the test Z-value.
P	Probability associated with the Z-Test statistic.
Critical Z	Boundary value(s) of Z associated with the α level that you supplied.
Critical $\Delta\ \bar{x}$	Difference in the means associated with the α level you supplied.

One-Proportion Z-Test

Menu name

Z-Test: $1\ \pi$

On the basis of statistics from a single sample, this test measures the strength of the evidence for a selected hypothesis against the null hypothesis. The null hypothesis is that the proportion of successes is an assumed value, $H_0: \pi = \pi_0$.

You select one of the following alternative hypotheses against which to test the null hypothesis:

$$H_1: \pi < \pi_0$$

$$H_1: \pi > \pi_0$$

$$H_1: \pi \neq \pi_0$$

Inputs

The inputs are:

Field name	Definition
x	Number of successes in the sample.
n	Sample size.
π_0	Population proportion of successes.
α	Significance level.

Results

The results are:

Result	Description
Test Z	Z-Test statistic.
Test $\hat{p}$	Proportion of successes in the sample.
P	Probability associated with the Z-Test statistic.
Critical Z	Boundary value(s) of Z associated with the α level that you supplied.
Critical $\hat{p}$	Proportion of successes associated with the level you supplied.

Two-Proportion Z-Test

Menu name

Z-Test: $\pi_1 - \pi_2$

On the basis of statistics from two samples, each from a different population, the Two-Proportion Z-Test measures the strength of the evidence for a selected hypothesis against the null hypothesis. The null hypothesis is that the proportions of successes in the two populations are equal, H_0: $\pi_1 = \pi_2$.

You select one of the following alternative hypotheses against which to test the null hypothesis:

$$H_1: \pi_1 < \pi_2$$

$$H_1: \pi_1 > \pi_2$$

$$H_1: \pi_1 \neq \pi_2$$

Inputs

The inputs are:

Field name	Definition
x_1	Sample 1 success count.
x_2	Sample 2 success count.
n_1	Sample 1 size.
n_2	Sample 2 size.
α	Significance level.

Results

The results are:

Result	Description
Test Z	Z-Test statistic.
Test $\Delta\ \hat{p}$	Difference between the proportions of successes in the two samples that is associated with the test Z-value.
P	Probability associated with the Z-Test statistic.
Critical Z	Boundary value(s) of Z associated with the α level that you supplied.
Critical $\Delta\ \hat{p}$	Difference in the proportion of successes in the two samples associated with the level you supplied.

One-Sample T-Test

Menu name

T-Test: 1 μ

The One-Sample T-Test is used when the population standard deviation is not known. On the basis of statistics from a single sample, this test measures the strength of the evidence for a selected hypothesis against the null hypothesis. The null hypothesis is that the sample mean has some assumed value, $H_0 : \mu = \mu_0$.

You select one of the following alternative hypotheses against which to test the null hypothesis:

$$H_1 : \mu < \mu_0$$

$$H_1 : \mu > \mu_0$$

$$H_1 : \mu \neq \mu_0$$

Inputs

The inputs are:

Field name	Definition
$\bar{x}$	Sample mean.
s	Sample standard deviation.
n	Sample size.
μ_0	Hypothetical population mean.
α	Significance level.

Results

The results are:

Result	Description
Test T	T-Test statistic.
Test $\bar{x}$	Value of $\bar{x}$ associated with the test t-value.
P	Probability associated with the T-Test statistic.
DF	Degrees of freedom.

Result	Description
Critical T	Boundary value(s) of T associated with the α level that you supplied.
Critical $\bar{x}$	Boundary value(s) of $\bar{x}$ required by the α value that you supplied.

Two-Sample T-Test

Menu name

T-Test: $\mu_1 - \mu_2$

The Two-sample T-Test is used when the population standard deviation is not known. On the basis of statistics from two samples, each sample from a different population, this test measures the strength of the evidence for a selected hypothesis against the null hypothesis. The null hypothesis is that the two populations means are equal, H_0: $\mu_1 = \mu_2$.

You select one of the following alternative hypotheses against which to test the null hypothesis

$$H_1{:}\mu_1 < \mu_2$$

$$H_1{:}\mu_1 > \mu_2$$

$$H_1{:}\mu_1 \neq \mu_2$$

Inputs

The inputs are:

Field name	Definition
$\bar{x}_1$	Sample 1 mean.
$\bar{x}_2$	Sample 2 mean.
s_1	Sample 1 standard deviation.
s_2	Sample 2 standard deviation.
n_1	Sample 1 size.
n_2	Sample 2 size.
α	Significance level.

Field name	Definition
`Pooled`	Check this option to pool samples based on their standard deviations.

Results

The results are:

Result	Description
Test T	T-Test statistic.
Test $\Delta \bar{x}$	Difference in the means associated with the test t-value.
P	Probability associated with the T-Test statistic.
DF	Degrees of freedom.
Critical T	Boundary values of T associated with the α level that you supplied.
Critical $\Delta \bar{x}$	Difference in the means associated with the α level you supplied.

Confidence intervals

The confidence interval calculations that the HP 39gII can perform are based on the Normal Z-distribution or Student's t-distribution.

One-Sample Z-Interval

Menu name

Z-int: 1 μ

This option uses the Normal Z-distribution to calculate a confidence interval for μ, the true mean of a population, when the true population standard deviation, σ, is known.

Inputs

The inputs are:

Field name	Definition
$\bar{x}$	Sample mean.
n	Sample size.
σ	Population standard deviation.
C	Confidence level.

Results

The results are:

Result	Description
C	Confidence level.
Critical Z	Critical values for Z.
Lower	Lower bound for μ.
Upper	Upper bound for μ.

Two-Sample Z-Interval

Menu name

Z-int: $\mu_1 - \mu_2$

This option uses the Normal Z-distribution to calculate a confidence interval for the difference between the means of two populations, $\mu_1 - \mu_2$, when the population standard deviations, σ_1 and σ_2, are known.

Inputs

The inputs are:

Field name	Definition
$\bar{x}_1$	Sample 1 mean.
$\bar{x}_2$	Sample 2 mean.
n_1	Sample 1 size.
n_2	Sample 2 size.

Field name	Definition
σ_1	Population 1 standard deviation.
σ_2	Population 2 standard deviation.
C	Confidence level.

Results

The results are:

Result	Description
C	Confidence level.
Critical Z	Critical values for Z.
Lower	Lower bound for $\Delta\mu$.
Upper	Upper bound for $\Delta\mu$.

One-Proportion Z-Interval

Menu name

Z-Int: 1π

This option uses the Normal Z-distribution to calculate a confidence interval for the proportion of successes in a population for the case in which a sample of size, n, has a number of successes, x.

Inputs

The inputs are:

Field name	Definition
x	Sample success count.
n	Sample size.
C	Confidence level.

Results

The results are:

Result	Description
C	Confidence level.

Result	Description
Critical Z	Critical values for Z.
Lower	Lower bound for π.
Upper	Upper bound for π.

Two-Proportion Z-Interval

Menu name

Z-Int: $\pi_1 - \pi_2$

This option uses the Normal Z-distribution to calculate a confidence interval for the difference between the proportions of successes in two populations.

Inputs

The inputs are:

Field name	Definition
$\bar{x}_1$	Sample 1 success count.
$\bar{x}_2$	Sample 2 success count.
n_1	Sample 1 size.
n_2	Sample 2 size.
C	Confidence level.

Results

The results are:

Result	Description
C	Confidence level.
Critical Z	Critical values for Z.
Lower	Lower bound for $\Delta\pi$.
Upper	Upper bound for $\Delta\pi$.

One-Sample T-Interval

Menu name

T-int: 1 μ

This option uses the Student's t-distribution to calculate a confidence interval for μ, the true mean of a population, for the case in which the true population standard deviation, σ, is unknown.

Inputs

The inputs are:

Field name	Definition
$\bar{x}$	Sample mean.
s	Sample standard deviation.
n	Sample size.
C	Confidence level.

Results

The results are:

Result	Description
C	Confidence level.
DF	Degrees of freedom.
Critical T	Critical values for T.
Lower	Lower bound for μ.
Upper	Upper bound for μ.

Two-Sample T-Interval

Menu name

T-int: $\mu_1 - \mu_2$

This option uses the Student's t-distribution to calculate a confidence interval for the difference between the means of two populations, $\mu_1 - \mu_2$, when the population standard deviations, σ_1 and σ_2, are unknown.

Inputs

The inputs are:

Field name	Definition
$\bar{x}_1$	Sample 1 mean.
$\bar{x}_2$	Sample 2 mean.
s_1	Sample 1 standard deviation.
s_2	Sample 2 standard deviation.
n_1	Sample 1 size.
n_2	Sample 2 size.
C	Confidence level.
Pooled	Whether or not to pool the samples based on their standard deviations.

Results

The results are:

Result	Description
C	Confidence level.
DF	Degrees of freedom.
Critical T	Critical values for T.
Lower	Lower bound for $\Delta\mu$.
Upper	Upper bound for $\Delta\mu$.

8

Parametric app

About the Parametric app

The Parametric app allows you to explore parametric equations. These are equations in which both *x* and *y* are defined as functions of *t*. They take the forms $x = f(t)$ and $y = g(t)$.

Getting started with the Parametric app

The following example uses the parametric equations

$$x(t) = 5\sin t$$
$$y(t) = 5\cos t$$

Note: this example will produce a circle. For this example to work, the angle measure must be set to degrees.

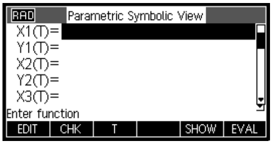
Open the Parametric app

1. Open the Parametric app.

   Apps Info *Select*

   `Parametric`

   RESET OK START



   Like the function app, the Parametric app opens in the Symbolic view.

Define the expressions

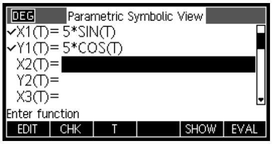
2. Define the expressions.

5 × SIN

X,T,θ,N) ENTER

5 × COS

X,T,θ,N) ENTER



Set angle measure

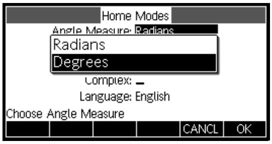
3. Set the angle measure to degrees.

SHIFT *MODES*

CHOOS

Select `Degrees`

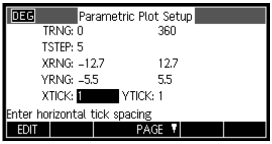
OK



Set up the plot

4. Set up the plot by displaying the graphing options.

SHIFT *PLOT-SETUP*

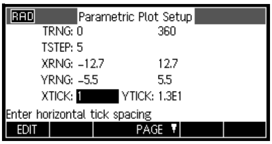


The Plot Setup input form has two fields not included in the Function app, `TRNG` and `TSTEP`. `TRNG` specifies the range of *t* values. `TSTEP` specifies the step value between *t* values.

5. Set the `TRNG` and `TSTEP` so that *t* steps from 0° to 360° in 5° steps.

▶ 360 OK

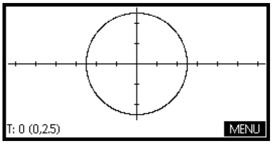
5 OK



Plot the expression

6. Plot the expression.

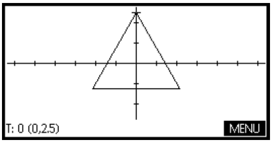
   Plot Setup



Explore the graph

7. Plot a triangle instead of a circle.

   SHIFT Plot Setup



   ▼ 120 OK

   PAGE ▼ ▼

   ▼

   Select `Fixed-Step Segments` OK Plot Setup

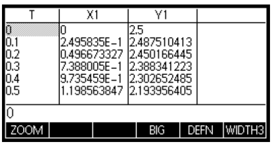
   A triangle is displayed rather than a circle (without changing the equation) because the changed value of `TSTEP` ensures that points being plotted are 120° apart instead of nearly continuous, and selecting `Fixed-Step Segments` connects the points 120° apart with line segments.

   You are able to explore the graph using the trace, zoom, split screen, and scaling functionality available in the Function app.

Display the numeric view

8. Display the Numeric view.

   Num Setup



T	X1	Y1
0	0	2.5
0.1	2.495835E−1	2.487510413
0.2	0.496673327	2.450166445
0.3	7.388005E−1	2.388341223
0.4	9.735459E−1	2.302652485
0.5	1.198563847	2.193956405

9. With a t-value selected, type in a replacement value, and see the table jump to that value. You can also zoom in or zoom out on any *t*-value in the table. You are able to explore the table using the zoom, build

your own table, and the split screen functionality available in the Function app.

9

Polar app

About the Polar app

The Polar app allows you to explore polar equations. Polar equations are equations in which *r* is defined in terms of θ. They take the form $r = f(\theta)$.

Getting started with the Polar app

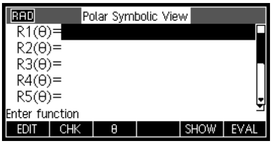
Open the Polar app

1. Open the Polar app.

   [Apps/Info] *Select* `Polar`

   RESET OK START

   Like the Function app, the Polar app opens in the Symbolic view.



Define the expression

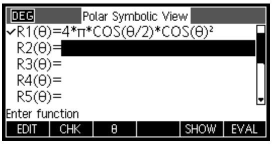
2. Define the polar equation $r = 4\pi\cos(\theta/2)\cos(\theta)^2$.

   [4] [×] [SHIFT]

   [3] [×] [COS]

   [X,T,θ,N] [÷] [2]

   [)] [×] [COS] [X,T,θ,N] [)] [x²] [ENTER]



Set angle measure

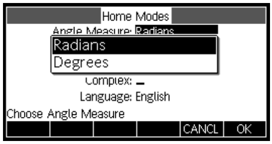
3. Set the angle measure to radians.

SHIFT *MODES*

CHOOS

Select `Radians`

OK



Set up the plot

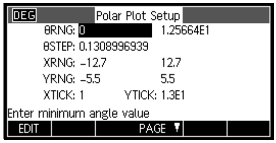
4. Set up the plot. In this example, we will use the default settings, except for the `θRNG` fields.

SHIFT *SETUP-PLOT*

SHIFT *CLEAR*

▶ 4 (Matrix T) SHIFT π

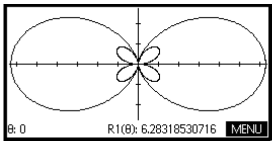
OK



Plot the expression

5. Plot the expression.

Plot (Setup)

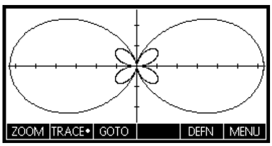


Explore the graph

6. Display the Plot view menu key labels.

MENU

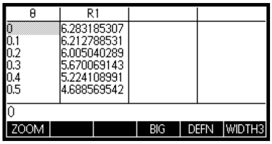
The Plot view options available are the same as those found in the Function app, except there is no FCN menu.



Display the Numeric view

7. Display the table of values for θ and R1 in the Numeric view.

Num Setup



8. With a θ-value selected, type in a replacement value and press OK, and see the table jump to that value. You can also zoom in or zoom out on any θ-value in the table.

10

Sequence app

About the Sequence app

The Sequence app allows you to explore sequences.

You can define a sequence named, for example, U1:

- in terms of n
- in terms of U1(n–1)
- in terms of U1(n–2)
- in terms of another sequence, for example, U2(n)
- in any combination of the above.

The Sequence app allows you to create two types of graphs:

  – A **Stairsteps** graph plots n on the horizontal axis and U_n on the vertical axis.

  – A **Cobweb** graph plots U_{n-1} on the horizontal axis and U_n on the vertical axis.

Getting started with the Sequence app

The following example defines and then plots an expression in the Sequence app. The sequence illustrated is the well-known Fibonacci sequence where each term, from the third term on, is the sum of the preceding two terms. In this example, we specify three sequence fields: the first term, the second term and a rule for generating all subsequent terms.

However, you can also define a sequence by specifying just the first term and the rule for generating all subsequent terms. You will, though, have to enter the second term if the HP 39gII is unable to calculate it automatically. Typically if the nth term in the sequence depends on $n-2$, then you must enter the second term.

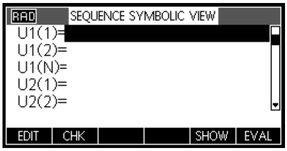
Open the Sequence app

1. Open the Sequence app.

   [Apps Info] *Select*

   `Sequence`

   RESET OK START

   The Sequence app starts in the Symbolic view.

Define the expression

2. Define the Fibonacci sequence, in which each term (after the first two) is the sum of the preceding two terms:

   $U_1 = 1, \ U_2 = 1, \ U_n = U_{n-1} + U_{n-2}$ for $n > 2$.

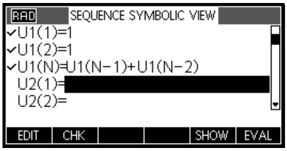
   In the Symbolic view of the Sequence app, highlight the U1(1) field and begin defining your sequence.

   1 [ENTER ANS] 1 [ENTER ANS]

   U1

   (N-1) [+] U1

   (N-2) [ENTER ANS]

   Note: You can use the U1, N, (N-1), *and* (N-2) *menu keys to assist in the entry of expressions.*

Set up the plot

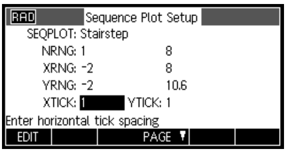
3. In Plot Setup, set the `SEQPLOT` option to `Stairstep and` reset the default plot settings by clearing the Plot Setup view.

   [SHIFT] *SETUP-PLOT*

   [SHIFT] *CLEAR*

   [▼] [▶] 8 [ENTER ANS]

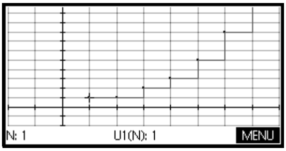
   [▶] 8 [ENTER ANS]

Plot the expression

4. Plot the Fibonacci sequence.

   [Plot Setup]



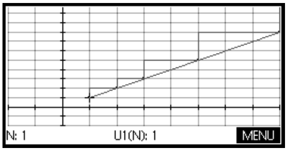
5. In Plot Setup, set the `SEQPLOT` option to `Cobweb`.

   [SHIFT] *SETUP-PLOT*

   [CHOOS] *Select* `Cobweb`

   [OK]

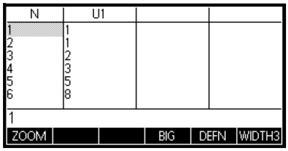
   [Plot Setup]



Display the numeric view

6. Display the Numeric view for this example.

   [Num Setup]



7. With any n-value selected, type in a replacement value, and see the table jump to that value.

11

Finance app

About the Finance app

The Finance app, or Finance Solver, provides you with the ability to solve time-value-of-money (TVM) and amortization problems. These problems can be used for calculations involving compound interest applications as well as amortization tables.

Compound interest is the process by which earned interest on a given principal amount is added to the principal at specified compounding periods, and then the combined amount earns interest at a certain rate. Financial calculations involving compound interest include savings accounts, mortgages, pension funds, leases, and annuities.

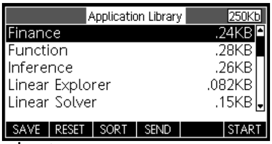
Getting Started with the Finance app

Suppose you finance the purchase of a car with a 5-year loan at 5.5% annual interest, compounded monthly. The purchase price of the car is $19,500, and the down payment is $3,000. What are the required monthly payments? What is the largest loan you can afford if your maximum monthly payment is $300? Assume that the payments start at the end of the first period.

1. Start the Finance app.

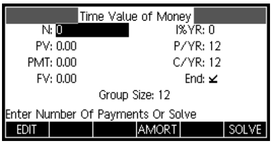
   Apps Info *select* `Finance`

   RESET OK START.



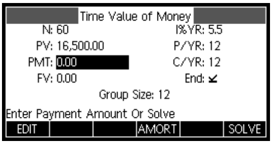
   The Finance app opens in the Numeric view.

2. *Select* `N`, type 5 x 12 and press ENTER ANS.

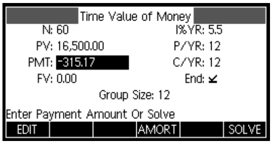


NOTE After you type in a value and press ENTER/ANS or OK, another variable is automatically highlighted. To manually navigate to a desired field, press the arrow keys. Be sure that values are entered for six of the seven TVM variables: `N`, `I%YR`, `PV`, `P/YR`, `PMT`, `C/YR`, and `FV`.

3. With `I%/YR` highlighted, type 5.5 and press ENTER/ANS.
4. With `PV` highlighted, type 19,500-3,000 and press ENTER/ANS.
5. Leave `P/YR` and `C/YR` both at `12` (their default values). Leave `End` as the payment option. Also, leave Future Value, `FV=0.00`.

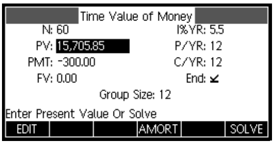


6. With `PMT` highlighted, press SOLVE to obtain a payment of -315.17 (i.e., `PMT` = -$315.17) as shown.



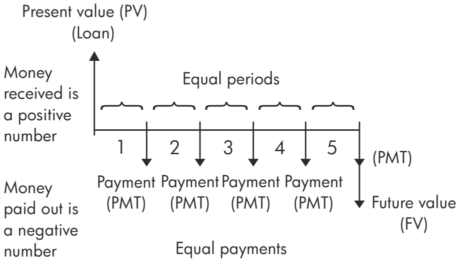
NOTE The payment is negative to indicate it is money owed.

7. To determine the maximum loan possible if the monthly payments are only $300, type the value –300 in the `PMT` field, highlight the PV field using ▲, and press SOLVE. The resulting value is PV = $15,705.85.

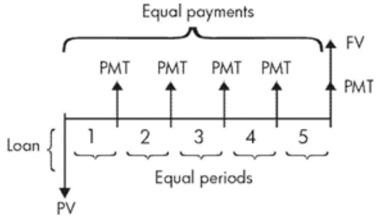


Cash flow diagrams

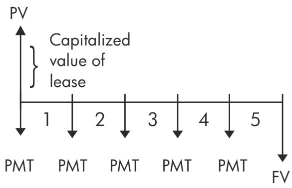
TVM transactions can be represented by using *cash flow diagrams*. A cash flow diagram is a time line divided into equal segments representing the compounding periods. Arrows represent the cash flows, which could be positive (upward arrows) or negative (downward arrows), depending on the point of view of the lender or borrower. The following cash flow diagram shows a loan from a *borrower's* point of view:



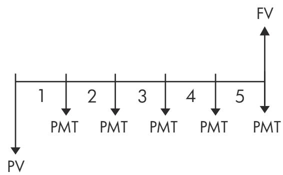
The following cash flow diagram shows a loan from the *lender's* point of view:



Cash flow diagrams also specify *when* payments occur relative to the compounding periods.The diagram to the right shows lease payments at the *beginning* of the period.



This diagram shows deposits (PMT) into an account at the end of each period.



Time value of money (TVM)

Time Value of Money (TVM) calculations, as the name implies, make use of the notion that a dollar today will be worth more than a dollar sometime in the future. A dollar today can be invested at a certain interest rate and generate a return that the same dollar in the future cannot. This TVM principal underlies the notion of interest rates, compound interest and rates of return. There are seven TVM variables:

Variable	**Description**
`N`	The total number of compounding periods or payments.
`I%YR`	The nominal annual interest rate (or investment rate). This rate is divided by the number of payments per year (P/YR) to compute the nominal interest rate *per compounding period* -- which is the interest rate actually used in TVM calculations.
`PV`	The present value of the initial cash flow. To a lender or borrower, PV is the amount of the loan; to an investor, PV is the initial investment. PV always occurs at the beginning of the first period.
`P/YR`	The number of payments made in a year.
`PMT`	The periodic payment amount. The payments are the same amount each period and the TVM calculation assumes that no payments are skipped. Payments can occur at the beginning or the end of each compounding period -- an option you control by un-checking or checking the `End` option.
`C/YR`	The number of compounding periods in a year.

FV	The future value of the transaction: the amount of the final cash flow or the compounded value of the series of previous cash flows. For a loan, this is the size of the final balloon payment (beyond any regular payment due). For an investment this is the cash value of an investment at the end of the investment period.

Performing TVM calculations

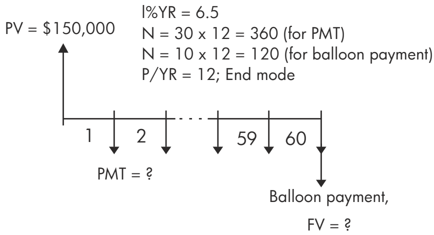
1. Launch the Finance App as indicated at the beginning of this section. It is recommended you reset the Finance app as shown before beginning a TVM problem.
2. With a variable highlighted, type in the known values starting with `N`, and press ENTER/ANS or OK to store the desired value. To manually navigate to a desired field, press the arrow keys.
3. Type in a different value for `P/YR` as required. The default value is 12, i.e., monthly payments.
4. With the `End` field highlighted, press the Check menu key CHK to uncheck this option for payments made at the beginning of each period or leave it checked for payments made at the end of each period.
5. Use the arrow keys to highlight the unknown variable and press SOLVE.

Example-mortgage with balloon payment

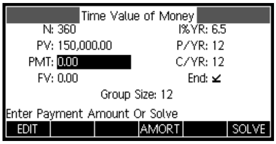
Suppose you have taken out a 30-year, $150,000 house mortgage at 6.5% annual interest. You expect to sell the house in 10 years, repaying the loan in a balloon payment. Find the size of the balloon payment, the value of the mortgage after 10 years of payment.

Solution

The following cash flow diagram illustrates the case of the mortgage with balloon payment:



1. Start the Finance App. Use the arrow keys to highlight `P/YR`. Verify that `P/YR = 12` and `End` is set for payments occurring at the end of the compounding period.
2. Enter the known TVM variables from the example as shown in the figure.



3. Highlight `PMT` and press SOLVE to obtain a payment of -$948.10.
4. To determine the balloon payment or future value (FV) for the mortgage after 10 years, enter 120 for `N`, highlight `FV`, and press SOLVE. This calculates the future value of the loan as -$127,164.19.

NOTE The negative values indicate payments from the homeowner.

Calculating Amortizations

Amortization calculations, which also use the TVM variables, determine the amounts applied towards principal and interest in a payment, or a series of payments.

To calculate amortizations:

1. Start the Finance Solver as indicated at the beginning of this section.
2. Set the following TVM variables:

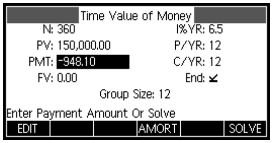
- Number of payments per year (`P/YR`)
- Payment at beginning or end of periods

3. Type and store values for the TVM variables, `I%YR`, `PV`, `PMT`, and `FV`, which define the payment schedule.
4. Enter the number of payments per amortization period in the `GSize` field. By default, the group size is 12 to reflect annual amortization.
5. Press **AMORT**. The calculator displays an amortization table. The table contains amounts applied to interest and principal, as well as the remaining balance of the loan, for each amortization period.

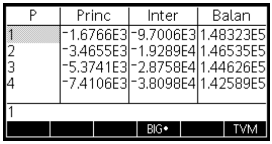
Example- Amortization for home mortgage

Using the data from the previous example of a home mortgage with balloon payment, calculate how much has been applied to the principal, how much has been applied to the interest, and the remaining balance of the loan after the first 10 years (12x10 = 120 payments).

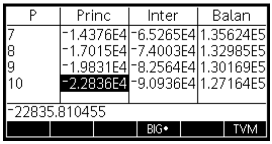
1. Verify and compare your data from the previous example with the figure to the right.



2. Press **AMORT**

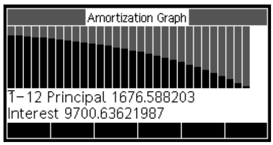


3. Scroll down the table to Group 10 to see the same results as shown previously. After 10 years, $22,835.81 has been paid on the principal, with an additional $90,936.43 paid in interest, leaving a balloon payment due of $127,164.19.



Amortization graph

Press the Plot key to see the amortization schedule presented graphically. The tracer shows the principal and interest paid in each payment group. Use the right- and left-cursor keys to trace among the payment groups.



12

Linear Solver app

About the Linear Solver app

The Linear Solver app allows you to solve a set of linear equations. The set can contain two or three linear equations.

In a two-equation set, each equation must be in the form $ax + by = k$. In a three-equation set, each equation must be in the form $ax + by + cz = k$.

You provide values for *a*, *b*, and *k* (and *c* in three-equation sets) for each equation, and the Linear Solver app will attempt to solve for *x* and *y* (and *z* in three-equation sets).

The HP 39gII will alert you if no solution can be found, or if there is an infinite number of solutions.

Getting started with the Linear Solver app

The following example defines a set of three equations and then solves for the unknown variables. In this example, we are going to solve the following equation set:

$$6x + 9y + 6z = 5$$

$$7x + 10y + 8z = 10$$

$$6x + 4y = 6$$

Hence we need the three-equation input form.

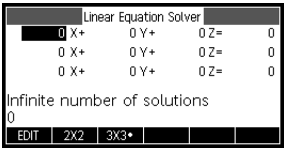
Open the Linear Solver app

1. Open the Linear Solver app.

   Apps Info *Select* `Linear Solver`

   RESET OK

   START



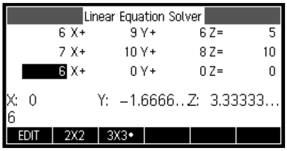
The Linear Equation Solver opens in the Numeric view.

NOTE If the last time you used the Linear Solver app you solved for two equations, the two-equation input form is displayed. To solve a three-equation set, press 3X3; now the input form displays three equations.

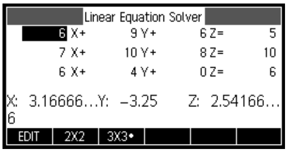
Define and solve the equations

2. You define the equations you want to solve by entering the coefficients of each variable in each equation and the constant term. Notice that the cursor is immediately positioned at the coefficient of *x* in the first equation. Enter that coefficient and press OK or ENTER.

3. The cursor moves to the next co-efficient. Enter that co-efficient, press OK or ENTER, and continue doing likewise until you have defined all the equations.



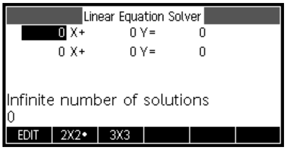
Once you have entered enough values for the solver to be able to generate solutions, those solutions appear on the display. In the example at the right, the solver was able to find solutions for *x*, *y*, and *z* as soon as the first co-efficient of the last equation was entered.



As you enter each of the remaining known values, the solution changes. The example at the right shows the final solution once all the coefficients and constants are entered for the set of equations we set out to solve.

Solve a two-by-two system

If the three-equation input form is displayed and you want to solve a two-equation set, press 2X2.



NOTE You can enter any expression that resolves to a numerical result, including variables; you can enter the name of a stored variable. For more information on storing variables, see the chapter titled *Using mathematical functions*.

13

Triangle Solver app

About the Triangle Solver app

The Triangle Solver app allows you to determine the length of a side of a triangle, or the measure of an angle of a triangle, from information you supply about the other lengths and/or angles.

You need to specify at least three of the six possible values—the lengths of the three sides and the measures of the three angles—before the solver can calculate the other values. Moreover, at least one value you specify must be a length. For example, you could specify the lengths of two sides and one of the angles; or you could specify two angles and one length; or all three lengths. In each case, the solver will calculate the remaining lengths or angle measures.

The HP 39gII will alert you if no solution can be found, or if you have provided insufficient data.

If you are determining the properties of a right-angled triangle, a simpler input form is available by pressing the RECT menu key.

Getting started with the Triangle Solver app

The following example solves for the unknown length of the side of a triangle whose two known sides—of lengths 4 and 6—meet at an angle of 30 degrees.

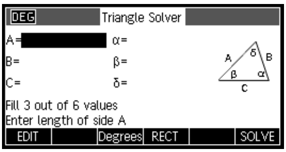
Open the Triangle Solver app

1. Open the Triangle Solver app.

Apps Info *Select*

`Triangle Solver`

RESET OK START



The Triangle Solver app opens in the Numeric view. This is the only view for this app.

Set angle measure

Make sure that your angle measure mode is appropriate. By default, the app starts in degree mode. If the angle information you have is in radians and your current angle measure mode is degrees, change the mode to degrees before running the solver. The `Degree` menu key is a toggle. Press it once to see it change to `Radians` for angles expressed in radians; press it again to return to degrees.

NOTE The lengths of the sides are labeled A, B, and C, and the angles are labeled α, β, and δ. It is important that you enter the known values in the appropriate fields. In our example, we know the length of two sides and the angle at which those sides meet. Hence if we specify the lengths of sides A and B, we must enter the angle as δ (since δ is the angle where A and B meet). If instead we entered the lengths as B and C, we would need to specify the angle as α. The illustration on the display will help you determine where to enter the known values.

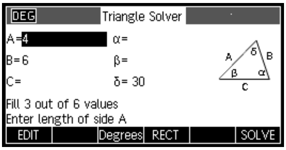
Specify the known values

2. Using the arrow keys, move to a field whose value you know, enter the value and press OK or ENTER ANS. Repeat for each known value.

4 ENTER ANS

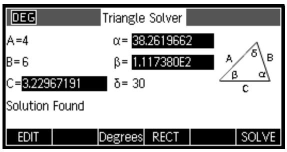
6 ENTER ANS ▶

30 ENTER ANS



Solve

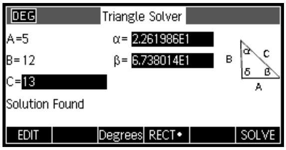
3. Press SOLVE. The solver calculates and displays the values of the unknown variables. As the illustration at the right shows, the length of the unknown side in our example is 3.22967. The other two angles have also been calculated.



Note: to clear all values and solve another problem, press SHIFT *CLEAR*.

Choose the triangle type

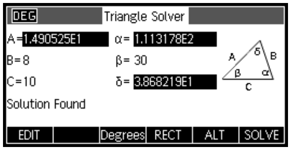
4. The Triangle Solver app offers you two input forms: a general input form and a more specialized form for right triangles. If the general input form is displayed, and you are investigating a right-angled triangle, press RECT to display the simpler input form. To return to the general input form, press RECT•. If the triangle you are investigating is not a right-angled triangle, or you are not sure what type it is, you should use the general input form.



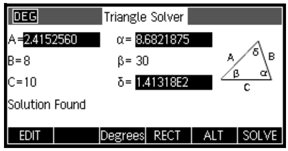
Special cases

The indeterminate case

If two sides and an adjacent acute angle are entered and there are two solutions, only one will be displayed initially.

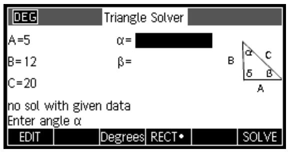


In this case, an ALT menu key is displayed (as in this example). You press ALT to display the second solution and ALT again to return to the first solution.



No solution with given data

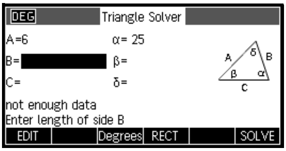
If you are using the general input form and you enter more than 3 values, the values might not be consistent, that is, no triangle could possibly have all the values you specified. In these cases, `No sol with given data` appears on the screen.



The situation is similar if you are using the simpler input form (for a right-angled triangle) and you enter more than two values.

Not enough data

If you are using the general input form, you need to specify at least three values for the Triangle Solver to be able to calculate the remaining attributes of the triangle. If you specify less than three, `Not enough data` appears on the screen.



If you are using the simplified input form (for a right-angled triangle), you must specify at least two values.

In addition, you cannot specify only angles and no lengths.

14

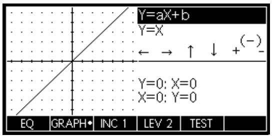
The Explorer Apps

Linear Explorer App

The Linear Explorer app is used to investigate the behavior of the graphs of $y = ax$ and $y = ax + b$ as the values of *a* and *b* change, both by manipulating the graph and seeing the change in the equation, and by manipulating the equation and seeing the change in the graph.

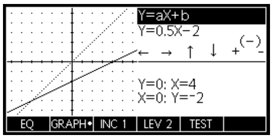
Open the app

Press Apps Info, select Linear Explorer, and press START. The app opens in Graph mode (note the dot in the GRAPH menu label).



Graph mode

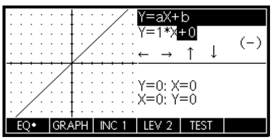
In Graph mode, ▲ and ▼ translate the graph vertically, effectively changing the y-intercept of the line. For vertical translations, press INC 1 (F3) to change the magnitude of the increment for the translation. The ◀ and ▶ keys (as well as - and +) increase and decrease the slope. Press (−) to change the sign of the slope.



The form of the linear function is shown at the top right of the display, with the current equation that matches the graph just below it. As you manipulate the graph of the line, the equation updates in real time to reflect the changes. Press LEV 2 (F4) to switch between direct variation and slope-intercept forms of linear functions.

Equation mode

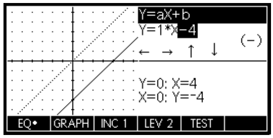
Press EQ (F1) to toggle to Equation mode. You will see the dot in the EQ menu key indicating the switch from Graph mode. You will also



see one of the parameters in the equation highlighted. In Equation mode, you change the values of one or more of the parameters in the equation and those changes are reflected in the graph. Press ▼ and ▲ to increase or decrease the value of the selected parameter, respectively. Press ▶ and ◀ to select another parameter. Press (−) to change the sign of *a*.

Test mode

Press TEST (F5) to enter Test mode. In Test mode, the app displays the graph of a randomly chosen linear function of the form dictated by your choice of level.

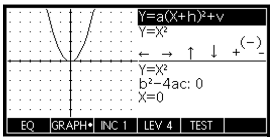


Press LEV 2 (F3) to choose between direct variation (LEV 1) and slope-intercept (LEV 2) forms of linear functions. Test mode then works like Equation mode. Use the arrow keys to select each parameter and set its value. When you are ready, press CHECK (F4) to see whether or not you have correctly matched your equation to the given graph. Press ANSW (F5) to see the correct answer. Press END (F6) to exit Test mode and return to Graph mode.

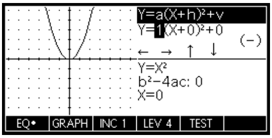
Quadratic Explorer app

The Quadratic Explorer app is used to investigate the behaviour of $y = a(x+h)^2 + v$ as the values of *a*, *h* and *v* change, both by manipulating the equation and seeing the change in the graph, *and* by manipulating the graph and seeing the change in the equation.

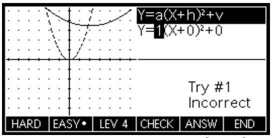
Press Apps, select `Quadratic Explorer`, and then press START. The Quadratic Explorer app opens in GRAPH• mode, in which the arrow keys, the + and - keys, and the (−) key are used to change the shape of the graph. This changing shape is reflected in the equation displayed at the top right corner of the screen, while the original graph is retained for comparison. In this mode the graph controls the equation.



It is also possible to have the equation control the graph. Press EQ to enter Equation mode. Press ▶ and ◀ to move between parameters and press ▲ and ▼ to change the value of a parameter. The graph of the equation will update in real time as you change the values of the parameters. Press LEV 2 to cycle through the various forms of quadratic functions available.



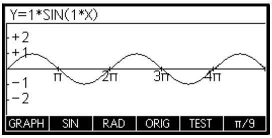
A TEST menu key is provided to evaluate the student's knowledge. Press TEST to display a target quadratic graph. The student must manipulate the equation's parameters to make the equation match the target graph. When a student feels that they have correctly chosen the parameters a CHECK menu key evaluates the answer and provide feedback. An ANSW menu key is provided for those who give up!



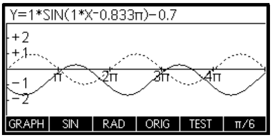
Trig Explorer app

The Trig Explorer app is used to investigate the behaviour of the graph of $y = a\sin(bx + c) + d$ as the values of *a*, *b*, *c* and *d* change, both by manipulating the equation and seeing the change in the graph, or by manipulating the graph and seeing the change in the equation.

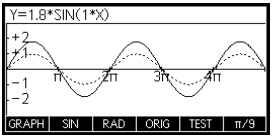
Press , select `Trig Explorer`, and then press START to display the screen shown right.



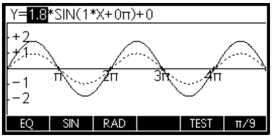
The app opens in Graph mode. Note that the first menu key (F1) is labeled `GRAPH`. In this mode, you can manipulate the graph and the changes are reflected in the equation. Press ▲, ▼, ◀ and ▶ to transform the graph, with the transformations reflected in the equation.



The button labelled ORIG is a toggle between ORIG and EXTR. When ORIG is chosen, ▲, ▼, ◄ and ► control vertical and horizontal translations. For horizontal translations, the F6 menu key controls the magnitude of the increment. By default, the increment is set at $\pi/9$. When EXTR is chosen ▲, ▼, ◄ and ► control vertical and horizontal dilations with respect to their respective axes. Thus, the arrow keys effectively change the amplitude and frequency of the graph. This is most easily seen by experimenting.



Press the F1 menu key to toggle from `GRAPH` to `EQ`. In this mode, the graph is controlled by the equation. In the equation displayed at the top of the screen, one of the parameters is highlighted. Press ▲ or ▼ to increase or decrease the value of the highlighted parameter. Press ► and ◄ to move from parameter to parameter.



The default angle setting for this app is radians. The angle setting can be changed to degrees by pressing RAD.

Like the Quadratic Explorer app, the Trig Explorer app also has a `TEST` view.

15

Extending your App Library

Apps are the application environments where you explore different classes of mathematical operations.

You can extend the capability of the HP 39gII by adding additional apps to the Apps Library. Adding new apps to the library can be done in a number of ways:

- Create new apps, based on existing apps, with specific configurations such as angle measure, graphical or tabular settings, and annotations.
- Transmit apps between HP 39gII calculators via micro-USB cable.
- Program new apps. See the chapter titled *Programming* for more details.

Creating new apps based on existing apps

You can create a new app based on an existing app. To create a new app, save an existing app under a new name, then modify the app to add the configurations and the functionality that you want.

Information that defines an app is saved automatically as it is entered into the calculator.

To keep as much memory available for storage as possible, delete any apps you no longer need.

Example

This example demonstrates how to create a new app by saving a copy of the built-in Solve app. The new app is saved under the name *TRIANGLES* and contains familiar formulas for solving problems involving triangles.

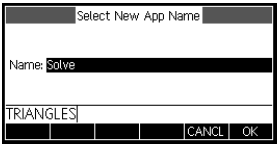
1. Open the Solve app and save it under the new name.

   Apps/Info `Solve`

   SAVE ALPHA ALPHA

   T R I A N G L E S

   ENTER/ANS START



2. Enter the formulas:

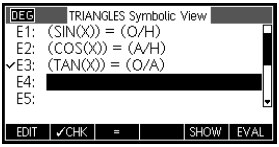
   SIN/ASIN E ALPHA X)/Paste M

   = ALPHA O x^{-1} ÷ N

   ALPHA H ENTER/ANS COS/ACOS F ALPHA X

)/Paste M = ALPHA A x^{-1} ÷ N ALPHA H ENTER/ANS TAN/ATAN G ALPHA

   X)/Paste M = ALPHA O x^{-1} ÷ N ALPHA A ENTER/ANS

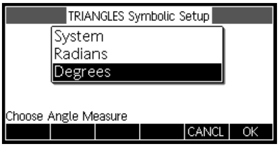


3. Decide whether you want the app to operate in `Degrees` or `Radians`.

   SHIFT Symb/Setup CHOOS

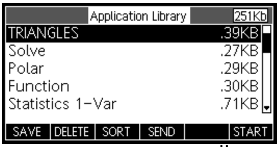
   `Degrees`

   OK



4. View the App Library. The `TRIANGLES` app is listed in the App Library.

   Apps/Info



The Solve app can now be reset and used for other problems. The advantage of storing an app is to allow you to keep a copy of a working environment for later use.

Resetting an app

Resetting an app clears all data and resets all default settings.

To reset an app, open the Library, select the app and press RESET.

You can only reset an app that is based on a built-in app if the programmer who created it has provided a Reset option.

Annotating an app with notes

The Info view (SHIFT Apps Info) attaches a note to the current app. See the chapter *Notes and Info* for more details.

Sending and receiving apps

A convenient way to distribute or share problems in class and to turn in homework is to transmit (copy) apps directly from one HP 39gII to another. Transfer of apps between calculators is done with the micro-USB cable that comes with each HP 39gII.

You can also send apps to, and receive apps from, a PC via the PC Connectivity Kit. A USB cable with a micro-USB connector is provided with the HP 39gII for connecting with a PC. It plugs into the micro-USB port on the calculator. The PC Connectivity Kit can be installed from the product CD included with the HP 39gII.

To transmit an app

1. Connect the two HP 39gII calculators with the micro-USB cable that came with each calculator.
2. On the sending calculator, open the Apps Library and select the app you wish to send.
3. Press the SEND menu key.
4. You may see the data transfer annunciator briefly.
5. On the receiving unit, open the Apps Library to see the new app.

To transmit an app from your PC to an HP 39gII, use the HP 39gII Connectivity Kit. This software application controls the transfer of all data from your PC to your HP 39gII.

Managing apps

The app library is where you go to manage your apps. Press Apps/Info. Highlight (using the cursor keys) the name of the app you want to act on.

To sort the app list

In the app library, press SORT. Select the sorting scheme and press ENTER/ANS.

- `Chronologically` produces a chronological order based on the date an app was last used. (The last-used app appears first, and so on.)
- `Alphabetically` produces an alphabetical order by app name.

To delete an app

To delete a customized app, open the app library, highlight the app to be deleted, and press ⬅/Clear. To delete all custom apps, press SHIFT *CLEAR*.

You cannot delete a built-in app. You can only clear its data and reset its default settings.

16

Using mathematical functions

Math functions

The HP 39gII contains many mathematical functions. To use a math function, you enter the function onto the command line, and include the function's argument(s) in parentheses after the function name. The most common math functions have their own key (or Shift of a key) on the keyboard. All the rest of the mathematical functions are found in the Math menu.

Keyboard functions

The most frequently used functions are available directly from the keyboard. Many of the keyboard functions also accept complex numbers as arguments.

+ - × ÷

Add, Subtract, Multiply, Divide. Also accepts complex numbers, lists and matrices.

value1+ value2, etc.

LN

Natural logarithm. Also accepts complex numbers.

`LN`(*value*)

Example:

`LN(1)` returns `0`

SHIFT e^x

Natural exponential. Also accepts complex numbers.

`e^`*value*

Example:

`e^5` returns `148.413159103`

LOG 10^x I

Common logarithm. Also accepts complex numbers.

LOG(*value*)

Example:

LOG(100) returns 2

SHIFT *10^x*

Common exponential (antilogarithm). Also accepts complex numbers.

10^*value*

Example:

10^3 returns 1000

SIN ASIN E   COS ACOS F   TAN ATAN G

Sine, cosine, tangent. Inputs and outputs depend on the current angle format (Degrees, Radians, or Grads).

SIN(*value*)
COS(*value*)
TAN(*value*)

Example:

TAN(45) returns 1 (Degrees mode).

SHIFT *ASIN*

Arc sine: $\sin^{-1}x$. Output range is from –90° to 90° or –π/2 to π/2. Inputs and outputs depend on the current angle format. Also accepts complex numbers.

ASIN(*value*)

Example:

ASIN(1) returns 90 (Degrees mode).

SHIFT *ACOS*

Arc cosine: $\cos^{-1}x$. Output range is from 0° to 180° or 0 to π. Inputs and outputs depend on the current angle format. Also accepts complex numbers. Output will be complex for values outside the normal cosine domain of $-1 \le x \le 1$.

ACOS(*value*)

Example:

ACOS(1) returns 0 (Degrees mode).

SHIFT *ATAN*

Arc tangent: $\tan^{-1}x$. Output range is from –90° to 90° or –π/2 to π/2. Inputs and outputs depend on the current angle format. Also accepts complex numbers.

`ATAN(`*value*`)`

Example:

`ATAN(1)` returns `45` (Degrees mode).

x^2

Square. Also accepts complex numbers.

value^2

Example:

`18`^{2} returns `324`

SHIFT √

Square root. Also accepts complex numbers.

√ *value* or √ *(expression)*

Example:

$\sqrt{324}$ returns `18`

x^y

Power (*x* raised to *y*). Also accepts complex numbers.

value^*power*

Example:

`2^8` returns `256`

SHIFT $\sqrt[n]{}$

Nth root ($\sqrt[n]{x}$). Takes the *n*th root of *x*.

root `NTHROOT` *value*

Example:

`3 NTHROOT 8` returns `2`

(−) ABS ;

Negation. Also accepts complex numbers.

–value

Example:

`-(1+2*i)` returns `-1-2*i`

SHIFT *ABS*

Absolute value. For a complex number, this is $\sqrt{x^2 + y^2}$.

`ABS`(*value*)
`ABS`((*x*+*y***i*))

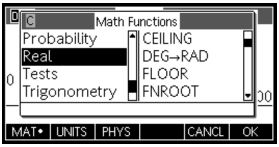
Example:

`ABS(–1)` returns `1`
`ABS((1,2))` returns `2.2360679775`

The Math menu

The Math menu provides access to math functions, units, and physical constants.

By default, pressing Math opens the Math Functions menu. Each of the three menus (Math Functions, Units, and SI Constants) has its own menu key. The Math menu is organized by *category*. For each category of functions on the left, there is a list of function names on the right. The highlighted category is the current category.



When you press Math, you see the menu list of Math categories in the left column and the corresponding functions of the highlighted category in the right column. The menu key MAT• indicates that the Math Functions menu list is active.

To select a function

1. Press [Math/Cmds] to display the Math menu. The categories appear in alphabetical order. Press ▼ or ▲ to scroll through the categories. To skip directly to a category, type the number (1-9) or letter (A-E) of the category.

2. The list of functions (on the right) applies to the currently highlighted category (on the left). Use ▶ and ◀ to switch between the category list and the function list.

3. Highlight the name of the function you want and press OK. This copies the function name (and an initial parenthesis, if appropriate) to the edit line.

Function categories

- Calculus
- Complex numbers
- Constant
- Distribution
- Hyperbolic trigonometry
- Integer
- List
- Loop
- Matrix
- Polynomial
- Probability
- Real numbers (Real)
- Tests
- Trigonometry

Math functions by category

Syntax

Each function's definition includes its syntax, that is, the exact order and spelling of a function's name, its delimiters (punctuation), and its arguments. Note that the syntax for a function does not require spaces.

Calculus functions

This category contains the numerical derivative and integral functions, as well as the Where function (|).

∂

Differentiates *expression* with respect to *variable* then substitutes value for variable and evaluates the result.

∂ (*expression, variable=value*)

Example:

∂ `(x`^{2}`-x,x=3)` returns `5`

∫

Integrates *expression* from *lower* to *upper* limits with respect to the *variable* of integration. To find the definite integral, both limits must have numeric values (that is, be numbers or real variables).

∫ (*expression, variable, lower, upper*)

Example:

∫ `(x`^{2}`-x,x,0,3)` returns `4.5`

|

Evaluates *expression* where each given variable is set to the given *value*. Defines numeric evaluation of a symbolic expression.

expression|(variable1=value1, variable2=value2,...)

Example:

`3*(X+1)|(X=3)` returns `12`

Complex number functions

These functions are for complex numbers only. You can also use complex numbers with all trigonometric and hyperbolic functions, and with some real-number and keyboard functions. Enter complex numbers in the form (*x*+*y***i*), where *x* is the real part and *y* is the imaginary part.

ARG

Argument. Finds the angle defined by a complex number. Inputs and outputs use the current angle format set in Modes.

`ARG((x+y*i))`

Example:

`ARG(3+3*i)` returns `45` (Degrees mode)

CONJ

Complex conjugate. Conjugation is the negation (sign reversal) of the imaginary part of a complex number.

`CONJ((x+y*i))`

Example:

`CONJ(3+4*i)` returns `(3-4*i)`

IM

Imaginary part, *y*, of a complex number, (*x*+*y***i*).

`IM ((x+y*i))`

Example:

`IM(3+4*i)` returns `4`

RE

Real part *x*, of a complex number, (*x*+*y***i*).

`RE((x+y*i))`

Example:

`RE(3+4*i)` returns `3`

Constants

The constants available from the Math Functions menu are mathematical constants. These are described in this section. The HP 39gII has two other menus of constants: program constants and physical constants. The physical constants are described further on in this chapter, while the program constants are described in the programming chapter.

e

Natural logarithm base. Internally represented as 2.71828182846.

e

i Imaginary value for $\sqrt{-1}$, the complex number (0,1).

i

MAXREAL Maximum real number. Internally represented as $9.99999999999 \times 10^{499}$.

MAXREAL

MINREAL Minimum real number. Internally represented as 1×10^{-499}.

MINREAL

π Internally represented as 3.14159265359.

π

Distribution

This category contains probability density functions, and both cumulative probability functions and their inverses, for the common probability distributions. These distributions include the Normal, Binomial, Chi-square, Fisher, Poisson, and Student's t distributions.

normald Normal probability density function. Computes the probability density at the value x, given the mean, μ and standard deviation, σ of a normal distribution. If only a single value (x) is supplied, assumes μ=0 and σ=1.

`normald([μ, σ,] x)`

Example:

`normald(0.5)` and `normald(0, 1, 0.5)` both return `0.352065326765`.

normald_cdf Cumulative normal distribution function. Returns the lower-tail probability of the normal probability density function for the value x, given the mean, μ and standard deviation, σ of a normal distribution. If only a single value (x) is supplied, assumes μ=0 and σ=1.

`normald_cdf([μ, σ,] x)`

Example:

`normald_cdf(0, 1, 2)` returns `0.97724986805`.

normald_icdf

Inverse cumulative normal distribution function. Returns the cumulative normal distribution value associated with the lower-tail probability, *p*, given the mean, μ and standard deviation, σ of a normal distribution. If only a single value (x) is supplied, assumes μ=0 and σ=1.

`normald_cdf`([μ, σ,] *p*)

Example:

`normald_icdf`(0, 1, 0.841344746069) returns `1`.

binomial

Binomial probability density function. Computes the probability of *k* successes out of *n* trials, each with a probability of success, *p*. Returns Comb(n,k) if there is no third argument. Note that *n* and *k* are integers with $k \le n$.

`binomial`(*n*, *k*, *p*)

Example:

`binomial`(4, 2, 0.5) returns `0.375`.

binomial_cdf

Cumulative binomial distribution function. Returns the probability of *k* or fewer successes out of *n* trials, with a probability of success, *p* for each trial. Note that *n* and *k* are integers with $k \le n$.

`binomial_cdf`(*n*, *p*, *k*)

Example:

`binomial_cdf`(4, 0.5, 2) returns `0.6875`.

binomial_icdf

Inverse cumulative binomial distribution function. Returns the number of successes, *k* out of *n* trials, each with a probability of *p,* such that the probability of *k* or fewer successes is *q*.

`binomial_icdf`(*n*, *p*, *q*)

Example:

`binomial_icdf`(4, 0.5, 0.6875) returns `2`.

chisquare

χ^2 probability density function. Computes the probability density of the χ^2 distribution at x, given *n* degrees of freedom.

`chisquare(`*n*, *x*`)`

Example:

`chisquare(2, 3.2)` returns `0.100948258997`.

chisquare_cdf

Cumulative χ^2 distribution function. Returns the lower-tail probability of the χ^2 probability density function for the value x, given *n* degrees of freedom.

`chisquare_cdf(`*n*, *k*`)`

Example:

`chisquare_cdf(2, 6.1)` returns `0.952641075609`.

chisquare_icdf

Inverse cumulative χ^2 distribution function. Returns the value x such that the χ^2 lower-tail probability of x, with *n* degrees of freedom, is *p*.

`chisquare_icdf(`*n*, *p*`)`

Example:

`chisquare_icdf(2, 0.952641075609)` returns `6.1`.

fisher

Fisher (or Fisher-Snedecor) probability density function. Computes the probability density at the value x, given numerator *n* and denominator *d* degrees of freedom.

`fisher(`*n*, *d*, *x*`)`

Example:

`fisher(5, 5, 2)` returns `0.158080231095`.

fisher_cdf

Cumulative Fisher distribution function. Returns the lower-tail probability of the Fisher probability density function for the value x, given numerator *n* and denominator *d* degrees of freedom.

`fisher_cdf(`*n*, *d*, *x*`)`

Example:

`fisher_cdf(5, 5, 2)` returns `0.76748868087`.

fisher_icdf

Inverse cumulative Fisher distribution function. Returns the value x such that the Fisher lower-tail probability of x, with numerator *n* and denominator *d* degrees of freedom, is *p*.

`fisher_icdf`(*n*, *d*, *p*)

Example:

`fisher_icdf`(5, 5, 0. .76748868087) returns `2`.

poisson

Poisson probability mass function. Computes the probability of *k* occurrences of an event in a time interval, given μ expected (or mean) occurrences of the event in that interval. For this function, *k* is a non-negative integer and μ is a real number.

`poisson`(μ , *k*)

Example:

`poisson`(4, 2) returns `0.14652511111`.

poisson_cdf

Cumulative poisson distribution function. Returns the probability x or fewer occurrences of an event in a given time interval, given μ expected occurrences.

`poisson_cdf`(μ , *x*)

Example:

`poisson_cdf`(4, 2) returns `0.238103305554`.

poisson_icdf

Inverse cumulative poisson distribution function. Returns the value x such that the probability of x or fewer occurrences of an event, with μ expected (or mean) occurrences of the event in the interval, is *p*.

`poisson_icdf`(μ , *p*)

Example:

`poisson_icdf`(4, 0.238103305554) returns `2`.

student

Student's t probability density function. Computes the probability density of the Student's-t distribution at x, given *n* degrees of freedom.

`student`(n, *x*)

Example:

`student`(3, 5.2) returns `0.00366574413491`.

student_cdf

Cumulative student's t distribution function. Returns the lower-tail probability of the student's t probability density function at x, given *n* degrees of freedom.

`student_cdf`(n, *x*)

Example:

`student_cdf`(3, –3.2) returns `0.0246659214813`.

student_icdf

Inverse cumulative student's t distribution function. Returns the value x such that the student's-t lower-tail probability of x, with *n* degrees of freedom, is *p*.

`student_icdf`(n, *p*)

Example:

`student_icdf`(3, 0.0246659214813) returns `3.2`.

Hyperbolic trigonometry

The hyperbolic trigonometry functions can also take complex numbers as arguments.

ACOSH

Inverse hyperbolic cosine : $\cosh^{-1}x$.

`ACOSH`(*value*)

ASINH

Inverse hyperbolic sine : $\sinh^{-1}x$.

`ASINH`(*value*)

ATANH

Inverse hyperbolic tangent : $\tanh^{-1}x$.

`ATANH`(*value*)

COSH

Hyperbolic cosine

`COSH`(*value*)

SINH

Hyperbolic sine.

`SINH`(*value*)

TANH

Hyperbolic tangent.

`TANH`(*value*)

ALOG

Antilogarithm (exponential). This is more accurate than `10^x` due to limitations of the power function.

`ALOG`(*value*)

EXP

Natural exponential. This is more accurate than e^x due to limitations of the power function.

`EXP(`*value*`)`

EXPM1

Exponent minus 1 : $e^x - 1$. This is more accurate than EXP when x is close to zero.

`EXPM1(`*value*`)`

LNP1

Natural log plus 1 : $\ln(x+1)$. This is more accurate than the natural logarithm function when x is close to zero.

`LNP1(`*value*`)`

Integer

ichinrem

Integer Chinese Remainder Theorem for two equations. Takes two lists [*a*, *p*] and [*b*, *q*] and returns a list of two integers, [*r*, *n*], such that $x = r \bmod n$. In this case, x is such that $x \equiv a \bmod p$ and $x \equiv b \bmod q$; also, $n = p \cdot q$

`ichinrem([`*a*, *p*`], [`*b*, *q*`])`

Example:

`ichinrem([2,7],[3,5])` returns `[-12,35]`.

idivis

Integer divisors. Returns a list of all the factors of the integer *a*.

`idivis(`*a*`)`

Example:

`idivis(12)` returns `[1,2,3,4,6,12]`.

iegcd

Integer extended greatest common divisor. For integers *a* and *b*, returns [*u*, *v*, i*gcd*] such that $u \cdot a + v \cdot b = igcd(a, b)$.

`iegcd(`*a*, *b*`)`

Example:

`iegcd(14, 21)` returns `[-1,1,7]`.

ifactor

Prime factorization. Returns the prime factorization of the integer *a* as a product.

`ifactor`(*a*)

Example:

`ifactor`(150) returns $2 \cdot 3 \cdot 5^2$.

ifactors

Prime factors. Similar to ifactor, but returns a list of the factors of the integer *a* with their multiplicities.

`ifactor`(*a*)

Example:

`ifactor`(150) returns `[2,1,3,1,5,2]`.

igcd

Greatest common divisor. Returns the integer that is the greatest common divisor of the integers *a* and *b*.

`igcd`(*a, b*)

Example:

`igcd`(24, 36) returns 12.

iquo

Euclidean quotient. Returns the integer quotient when the integer *a* is divided by the integer *b*.

`iquo`(*a, b*)

Example:

`iquo`(46, 21) returns 2.

iquorem

Euclidean quotient and remainder. Returns the integer quotient and remainder when the integer *a* is divided by the integer *b*.

`iquorem`(*a, b*)

Example:

`iquorem`(46, 21) returns [2, 4].

irem

Euclidean remainder. Returns the integer remainder when the integer *a* is divided by the integer *b*.

`irem`(*a, b*)

Example:

`irem`(46, 21) returns 4.

isprime

Prime integer test. Returns 1 if the integer *a* is prime; otherwise, returns 0.

`isprime`(*a*)

Example:

`isprime`(1999) returns 1.

ithprime

Nth prime. For the integer *n*, returns the nth prime number less than 10,000.

`ithprime`(*n*)

Example:

`ithprime`(5) returns 11.

nextprime

Next prime. Returns the next prime number after the integer a.

`nextprime`(*a*)

Example:

`nextprime`(11) returns 13.

powmod

Power and modulo. For the integers *a*, *n*, and *p*, returns a^n mod*p*.

`powmod`(*a, n, p*)

Example:

`powmod`(5, 2, 13) returns 12.

prevprime

Previous prime. Returns the previous prime number before the integer a.

`prevprime`(*a*)

Example:

`prevprime`(11) returns 7.

euler

Euler's phi (or totient) function. Takes a positive integer x and returns the number of positive integers less than or equal to x that are coprime to x.

`euler`(*x*)

Example:

`euler`(6) returns 2.

numer

Simplified Numerator. For the integers *a* and *b*, returns the numerator of the fraction a/b after simplification.

`numer`(a/b)

Example:

`numer`(10/12) returns 5.

denom

Simplified Denominator. For the integers *a* and *b*, returns the denominator of the fraction a/b after simplification.

`denom`(a/b)

Example:

`denom`(10/12) returns 6.

List functions

These functions work on list data. See the chapter *Lists* for details.

Loop functions

The loop functions display a result after evaluating an expression a given number of times.

ITERATE

Repeatedly for *#times* evaluates an *expression* in terms of *variable*. The value for *variable* is updated each time, starting with *initialvalue.*

`ITERATE(`*expression*, *variable*, *initialvalue*, *#times*`)`

Example:

`ITERATE(`X^2`,X,2,3)` returns `256`

Σ

Summation. Finds the sum of *expression* with respect to *variable* from *initialvalue* to *finalvalue.*

Σ`(`*expression, variable, initialvalue, finalvalue*`)`

Example:

`Σ(`x^2`,x,1,5)` returns 55.

Matrix functions

These functions are for matrix data stored in matrix variables. See the chapter *Matrices* for details.

Polynomial functions

Polynomials are products of constants (*coefficients*) and variables raised to powers (*terms*).

POLYCOEF

Polynomial coefficients. Returns the coefficients of the polynomial with the specified *roots*.

`POLYCOEF` ([*roots*])

Example:

To find the polynomial with roots 2, –3, 4, –5:
`POLYCOEF([2,-3,4,-5])` returns `[1,2,-25,-26,120]`, representing $x^4+2x^3-25x^2-26x+120$.

POLYEVAL

Polynomial evaluation. Evaluates a polynomial with the specified *coefficients* for the *value* of x.

`POLYEVAL([`*coefficients*`],` *value*`)`

Example:

For $x^4+2x^3-25x^2-26x+120$:
`POLYEVAL([1,2,-25,-26,120],8)` returns `3432`.

POLYROOT

Polynomial roots. Returns the roots for the *n*th-order polynomial with the specified *n*+1 *coefficients*.

`POLYROOT`([*coefficients*])

Example:

For $x^4+2x^3-25x^2-26x+120$:
`POLYROOT([1,2,-25,-26,120])` returns `[4,-5,-3,2]`.

HINT The results of `POLYROOT` will often not be easily seen in Home due to the number of decimal places, especially if they are complex numbers. It is better to store the results of `POLYROOT` to a matrix.

For example, `POLYROOT([1,0,0,-8]` **STO ▸** `M1` will store the three complex cube roots of 8 to matrix M1 as a complex vector. Then you can see them by going to the Matrix Catalog. You can also access them individually in calculations by referring to M1(1), M1(2) etc.

Probability functions

COMB

Number of combinations (without regard to order) of n things taken r at a time: $n!/(r!(n-r))$.

`COMB`(n, r)

Example:

`COMB(5,2)` returns `10`. That is, there are ten different ways that five things can be combined two at a time.

!

Factorial of a positive integer. For non-integers, $! = \Gamma(x + 1)$. This calculates the gamma function.

value!

Example:

5! Returns `120`

PERM

Number of permutations (with regard to order) of n things taken r at a time: $n!/(r!(n-r)!$

`PERM` (n, r)

Example:

`PERM(5,2)` returns `20`. That is, there are 20 different permutations of five things taken two at a time.

RANDOM

Random number. With no argument, this function returns a random number between zero and one. With one integer argument *a*, it returns a random integer between 0 and *a*. With three integer arguments, *n*, *a*, and *b*, returns n random integers between *a* and *b*.

`RANDOM`

`RANDOM(a)`

`RANDOM(n, a, b)`

UTPC

Upper-Tail Chi-Squared Probability given *degrees* of freedom, evaluated at *value*. Returns the probability that a χ^2 random variable is greater than *value.*

`UTPC`(*degrees, value*)

UTPF

Upper-Tail Snedecor's F Probability given *numerator* degrees of freedom and *denominator* degrees of freedom (of the F distribution), evaluated at *value*. Returns the probability that a Snedecor's F random variable is greater than *value.*

`UTPF`(*numerator, denominator, value*)

UTPN

Upper-Tail Normal Probability given *mean* and *variance*, evaluated at *value.* Returns the probability that a normal random variable is greater than *value* for a normal distribution. *Note: the variance is the square of the standard deviation*.

`UTPN`(*mean, variance, value*)

UTPT

Upper-Tail Student's t-Probability given *degrees of freedom*, evaluated at *value.* Returns the probability that the Student's t- random variable is greater than *value.*

`UTPT`(*degrees, value*)

Real-number functions

Some real-number functions can also take complex arguments.

CEILING

Smallest integer greater than or equal to *value*.

`CEILING`(*value*)

Examples:

`CEILING(3.2)` returns `4`
`CEILING(-3.2)` returns `-3`

DEG→RAD

Degrees to radians. Converts *value* from Degrees angle format to Radians angle format.

DEG→RAD(*value*)

Example:

`DEG→RAD(180)` returns `3.14159265359`, the value of π.

FLOOR

Greatest integer less than or equal to *value*.

FLOOR(*value*)

Example:

`FLOOR(-3.2)` returns `-4`

FNROOT

Function root-finder (like the Solve app). Finds the value for the given *variable* at which *expression* most nearly evaluates to zero. Uses *guess* as initial estimate.

FNROOT(*expression, variable, guess*)

Example:

`FNROOT(M*9.8/600-1,M,1)` returns `61.224489796.`

FRAC

Fractional part.

FRAC(*value*)

Example:

`FRAC (23.2)` returns `.2`

HMS→

Hours-minutes-seconds to decimal. Converts a number or expression in *H.MMSSs* format (time or angle that can include fractions of a second) to *x.x* format (number of hours or degrees with a decimal fraction).

HMS→(*H.MMSSs*)

Example:

`HMS→(8.30)` returns `8.5`

→HMS

Decimal to hours-minutes-seconds. Converts a number or expression in *x.x* format (number of hours or degrees with a decimal fraction) to *H.MMSSs* format (time or angle up to fractions of a second).

→HMS(*x.x*)

Example:

`→HMS(8.5)` returns `8.3`

INT

Integer part.

`INT(`*value*`)`

Example:

`INT(23.2)` returns `23`

MANT

Mantissa (significant digits) of *value*.

`MANT(`*value*`)`

Example:

`MANT(21.2E34)` returns `2.12`

MAX

Maximum. The greater of two values.

`MAX(`*value1*, *value2*`)`

Example:

`MAX(210,25)` returns `210`

MIN

Minimum. The lesser of two values.

`MIN(`*value1*, *value2*`)`

Example:

`MIN(210,25)` returns `25`

MOD

Modulo. The remainder of *value1*/*value2*.

value1 `MOD` *value2*

Example:

`9 MOD 4` returns `1`

%

x percent of y; that is, $x/100*y$.

`%(`x, y`)`

Example:

`%(20,50)` returns `10`

%CHANGE

Percent change from x to y, that is, $100(y–x)/x$.

`%CHANGE(`x, y`)`

Example:

`%CHANGE(20,50)` returns `150`

%TOTAL

Percent total : (100)y/x. What percentage of x, is y.

`%TOTAL`(x, y)

Example:

`%TOTAL(20,50)` returns `250`

RAD→DEG

Radians to degrees. Converts *value* from radians to degrees.

`RAD→DEG` (*value*)

Example:

`RAD→DEG(π)` returns `180`

ROUND

Rounds *value* to decimal *places*. Accepts complex numbers.

`ROUND`(*value*, *places*)

Round can also round to a number of significant digits as showed in the second example below.

Examples:

`ROUND(7.8676,2)` returns `7.87`

`ROUND(0.0036757,-3)` returns `0.00368`

SIGN

Sign of *value*. If positive, the result is 1. If negative, –1. If zero, result is zero. For a complex number, this is the unit vector in the direction of the number.

`SIGN`(*value*)
`SIGN`((x, y))

Example:

`SIGN (–2)` returns `–1`

`SIGN((3,4))` returns `(.6,.8)`

TRUNCATE

Truncates *value* to decimal *places*. Accepts complex numbers.

`TRUNCATE`(*value*, *places*)

Example:

`TRUNCATE(2.3678,2)` returns `2.36`

XPON

Exponent of *value*.

`XPON(`*value*`)`

Example:

`XPON(123.4)` returns `2`

Test functions

The test functions are logical operators that always return either 1 (*true*) or 0 (*false*).

<

Less than. Returns 1 if true, 0 if false.

value1<value2

≤

Less than or equal to. Returns 1 if true, 0 if false.

value1≤value2

==

Equals (logical test). Returns 1 if true, 0 if false.

value1==value2

≠

Not equal to. Returns 1 if true, 0 if false.

value1≠value2

>

Greater than. Returns 1 if true, 0 if false.

value1>value2

≥

Greater than or equal to. Returns 1 if true, 0 if false.

value1≥value2

AND

Compares *value1* and *value2*. Returns 1 if they are both non-zero, otherwise returns 0.

value1 `AND` *value2*

IFTE

If *expression* is true, do the *trueclause*; if not, do the *falseclause*.

`IFTE(`*expression*, *trueclause*, *falseclause*`)`

Example:

`IFTE(X>0,X`^{2}`,X`^{3}`)` with `x=-2` returns `-8`

NOT

Returns 1 if *value* is zero, otherwise returns 0.

`NOT` *value*

OR

Returns 1 if either *value1* or *value2* is non-zero, otherwise returns 0.

value1 `OR` *value2*

XOR

Exclusive OR. Returns 1 if either *value1* or *value2*—but not both of them—is non-zero, otherwise returns 0.

value1 `XOR` *value2*

Trigonometry functions

The trigonometry functions can also take complex numbers as arguments. For SIN, COS, TAN, ASIN, ACOS, and ATAN, see the Keyboard category.

ACOT

Arc cotangent.

`ACOT`(*value*)

ACSC

Arc cosecant.

`ACSC`(*value*)

ASEC

Arc secant.

`ASEC`(*value*)

COT

Cotangent: cos*x*/sin*x*.

`COT`(*value*)

CSC

Cosecant: 1/sin*x*

`CSC`(*value*)

SEC

Secant: 1/cos*x*.

`SEC`(*value*)

Units and physical constants

When you press Math Cmds, three menus become available:

- the Math Functions menu (which appears by default)
- the Units menu
- the Physical Constants menu

The math functions menu is described extensively earlier in this chapter.

Units

You can attach physical units to any numerical calculation or result. A numerical value with units attached is referred to as a measurement. You can operate on measurements just as you do on numbers without units attached, except that the units are carried along with the operations. The function usimplify (unit simplify) will simplify the results back to the simplest unit structure. The units are found in the Units menu. Like the Math menu, the Units menu is divided into a set of categories on the left and units in each category on the right. The categories are:

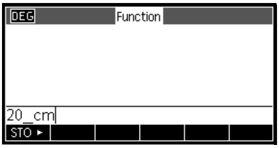
Unit categories

- Length
- Area
- Volume
- Time
- Speed
- Mass
- Acceleration
- Force
- Energy
- Power
- Pressure
- Temperature
- Electricity
- Light
- Angle
- Viscosity
- Radiation

Suppose you wish to add 20 centimeters and 5 inches.

1. If you want the result in cm, start by entering the 20 cm.

   20 Math Cmds UNITS

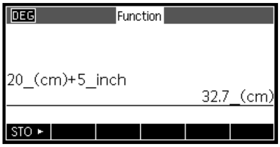


   ▼ ▼ (to select `Length`)

   ► ▼ (to select `_cm`)

   OK.

2. Now add 5 inches.

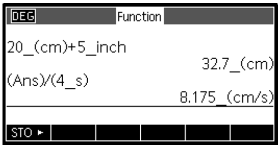


   [+] 5 [▶]

   [▲] (8 times for _inch)
   OK

   [ENTER].
   The result is shown as 32.7 cm. If you had wanted the result in inches, then you would have entered the 5 inches first.

3. To continue the example, we divide this result by 4 seconds and convert the result to kilometers per hour.



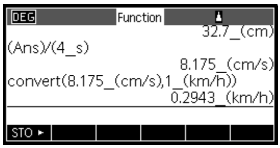
   [÷] 4 [Math]

   [▼] [▼] [▼] (to select Time)

   [▶] [▲] (to select _s)

   OK [ENTER].
   The result is shown as 8.175 cm/s.

4. Now convert the result to kilometers per hour.



   [Math] [▲] (5 times to select Functions)

   [▶] [▼] (to select convert) OK

   [Copy] [▲] (to select 8.175_(cm/s))

   COPY 1 [,] [Math]

   [▼] (6 times to select Speed) [▶]

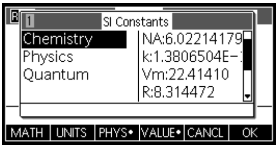
   [▼] (4 times to select _km/h) OK.
   The result is shown as 0.2943 kilometers per hour.

Physical constants

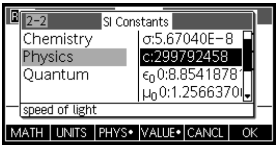
There are 29 physical constants you can use in calculations. These constants are grouped into the categories chemistry, physics and quantum mechanics. A list of all these constants can be found in *Physical Constants* in the *Reference Information* chapter.

To access the menu of physical constants:

1. Press Math Cmds.
2. Press PHYS.



3. Use the arrow keys to navigate through the options.
4. While in the Physical Constants menu, pressing VALUE toggles between showing the entire value of the constant and a description of the constant in the help line. To attach units to the constant when you paste it into the command line, keep VALUE• active when you press OK; to paste just the value without units, de-activate VALUE• before pressing OK.



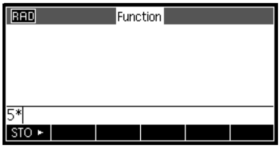
5. To use the selected constant in a calculation, press OK. The constant appears at the position of the cursor on the edit line.

Example:

Suppose you want to know the potential energy of a mass of 5 units according to the equation $E = mc^2$.

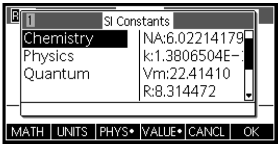
1. Enter the mass and multiplication.

   5 [×]



2. Go to the Physical Constants menu.

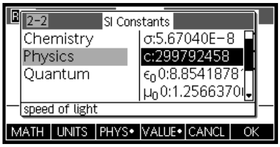
   [Math Cmds] PHYS



3. Select the speed of light.

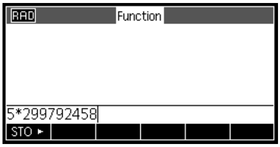
   [▼] (to select `Physics`)

   [►] [▼] (to select `c`)



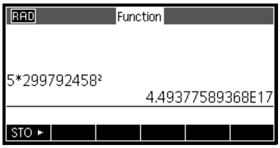
4. Enter the speed of light into the current expression.

   OK



5. Square the speed of light and evaluate the expression.

   [x^2] [ENTER]



17

Lists

Introduction

You can do list operations in Home and in programs. A list consists of comma-separated real or complex numbers, expressions, or matrices, all enclosed in braces. A list may, for example, contain a sequence of real numbers such as `{1,2,3}`. Lists represent a convenient way to group related objects.

There are ten list variables available, named `L0` to `L9`. You can use them in calculations or expressions in Home or in a program. Retrieve the list names from the Vars menu, or just type their names from the keyboard.

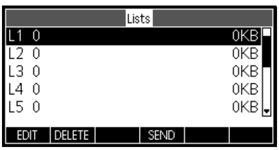
You can create, edit, delete, send, and receive named lists in the List catalog (SHIFT *LIST*). You can also create and store lists—named or unnamed—in Home.

List variables are identical in behavior to the columns `C1-C0` in the Statistics 2Var app and the columns `D1-D0` in the Statistics 1Var app. You can store a statistics column to a list (or vice versa) and use any of the list functions on the statistics columns, or the statistics functions on the list variables.

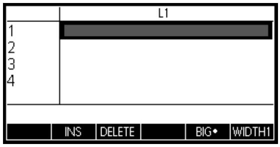
Create a list in the List Catalog

1. Open the List catalog.

   SHIFT *LIST*.

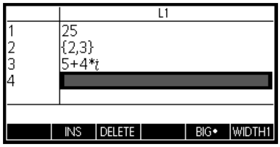


2. Highlight the list name you want to assign to the new list (`L1`, etc.) and press EDIT to display the List editor.



3. Enter the values you want in the list, pressing ENTER/ANS after each one.

   Values can be real or complex numbers (or an expression). If you enter a expression, it is evaluated and the result is inserted in the list.



4. When done, press SHIFT *LIST* to see the List catalog, or press Home/Modes to return to Home.

List Catalog keys

The list catalog keys are:

Key	**Meaning**
EDIT	Opens the highlighted list for editing.
DELETE or Clear	Deletes the contents of the selected list.
SEND	Transmits the highlighted list to another HP 39gII.
SHIFT *CLEAR*	Clears all lists.
SHIFT ▲ or ▼	Moves to the end or the beginning of the catalog.

The List Editor

Press EDIT to create or edit a list. Once you press this menu key, you enter the List Editor. The List Editor is a special environment for entering data into lists.

List edit keys

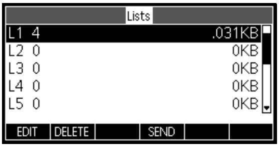
When you press EDIT to create or change a list, the following keys are available to you:

Key	Meaning
INS	Inserts a new value before the highlighted item.
EDIT	Copies the highlighted list item into the edit line.
BIG	Toggles between large and small fonts.
WIDTH1	Toggles between showing 1, 2, 3, or 4 lists at a time.
DELETE or Clear	Deletes the highlighted item from the list.
SHIFT *CLEAR*	Clears all elements from the list.
SHIFT ▲ or ▼	Moves to the end or the beginning of the list.

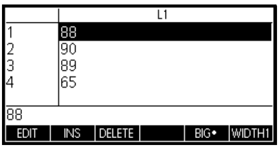
To edit a list

1. Open the List catalog.

   SHIFT *LIST*.



2. Press ▲ or ▼ to highlight the name of the list you want to edit (L1, etc.) and press EDIT to display the list contents.

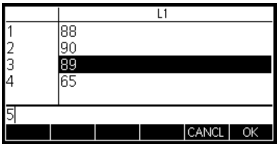


3. Press [▲] or [▼] to highlight the element you want to edit. In this example, edit the third element so that it has a value of 5.

   [▼] [▼] EDIT

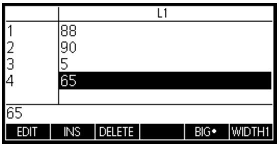
   [Clear] [Clear]

   5 OK.



To insert an element in a list

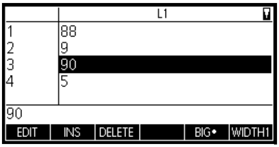
Suppose you wish to insert a new value, 9, in L1(2) in the list L1 shown to the right.



1. Move to the insertion point and insert the new value.

   [▲] [▲] INS,

   9 OK.



Deleting lists

To delete a list

In the List catalog, highlight the list name and press [Clear].

You are prompted to confirm that you want to delete the contents of the highlighted list variable. Press [ENTER ANS] to delete the contents, or [ON/C OFF] to cancel the deletion.

To delete all lists

In the List catalog, press [SHIFT] *CLEAR*.

Lists in the Home view

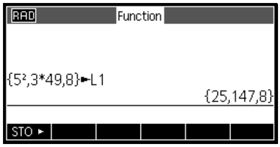
You can enter and operate on lists directly in the Home view. The lists you work on in the Home view can be named or not.

1. Enter the list on the edit line. Start and end the list with braces (the shifted [8] and [9] keys) and separate each element with a comma.

2. Press [ENTER] to evaluate and display the list.

   Immediately after typing in the list, you can store it in a variable by pressing [STO ►] *listname* [ENTER]. The list variable names are `L0` through `L9`.

   This example stores the list {25,147,8} in `L1`.



To display a list

To display a list in the Home view, type its name and press [ENTER].

To display one element

To display one element of a list in the Home view, enter *listname* (*element#*). For example, if L2 is {3,4,5,6}, then `L2(2)` [ENTER] returns `4`.

To store one element

To store a value in one element of a list in the Home view, enter *value* [STO ►] *listname* (*element#*). For example, to store 148 as the second element in L2, type `148` [STO ►] `L2(2)` [ENTER].

To transmit a list

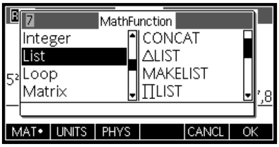
You can send lists to another calculator or a PC just as you can apps, programs, matrices, and notes. To send lists between two HP 39gII calculators:

1. Connect the two HP 39gII calculators with the micro-USB cable provided with the calculators and turn both calculators on.
2. Open the List catalog on the sending calculator.
3. Highlight the list to send.

4. Press SEND.
5. The transfer will occur immediately.
6. Open the List Catalog on the receiving calculator to see the new list.

List functions

List functions are found in the Math menu. You can use them in Home, as well as in programs.



You can type in the name of the function, or you can copy the name of the function from the List category of the MATH menu. Press Math 7 to highlight the List category in the left column of the Math menu (List is the seventh category in the Math menu). Press ▼ and ▲ to select the list function you want, select a function, and press OK.

List functions have the following syntax:

> Functions have arguments that are enclosed in parentheses and separated by commas. Example: `CONCAT(L1,L2)`. An argument can be either a list variable name (such as L1) *or* the actual list. For example, `REVERSE({1,2,3})`.

Common operators like +, –, ×, and / can take lists as arguments. If there are two arguments and both are lists, then the lists must have the same length, since the calculation pairs the elements. If there are two arguments and one is a real number, then the calculation pairs the number with each element of the list.

Example:

> `5*{1,2,3}` returns `{5,10,15}`.

Besides the common operators that can take numbers, matrices, or lists as arguments, there are commands that can only operate on lists.

CONCAT

Concatenates two lists into a new list.

> `CONCAT(`*list1*, *list2*`)`

Example:

CONCAT({1,2,3},{4}) returns {1,2,3,4}.

ΔLIST

Creates a new list composed of the first differences of a list, that is, the differences between the sequential elements in the list. The new list has one less element than the original list. The first differences for $\{x_1, x_2, x_3, \ldots x_{n-1}, x_n\}$ are $\{x_2-x_1, x_3-x_2, \ldots x_n-x_{n-1}\}$.

ΔLIST(*list1*)

Example:

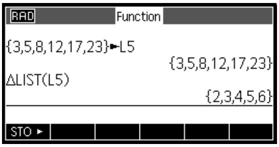
In Home, store {3,5,8,12,17,23} in L5 and find the first differences for the list.

Home Modes SHIFT

{3,5,8,12,17,23 SHIFT }

STO ► ALPHA L 5 ENTER ANS

Math Cmds B 7 2 ALPHA L 5 ENTER ANS



MAKELIST

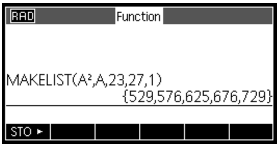
Calculates a sequence of elements for a new list. Evaluates *expression* with respect to *variable,* as *variable* takes on values from *begin* to *end* values, taken at *increment* steps.

MAKELIST(*expression*, *variable*, *begin*, *end*, *increment*)

The MAKELIST function generates a sequence by automatically producing a list from the repeated evaluation of an expression.

Example:

In Home, generate a series of squares from 23 to 27.



[Math Cmds] 7 3

[ALPHA] A [x²] [,]

[ALPHA] A [,] 23 [,]

27 [,] 1 [)] [ENTER]

ΠLIST

Calculates the product of all elements in list.

`ΠLIST(`*list*`)`

Example:

`ΠLIST({2,3,4})` returns `24`.

POS

Returns the position of an element within a list. The *element* can be a value, a variable, or an expression. If there is more than one instance of the element, the position of the first occurrence is returned. A value of 0 is returned if there is no occurrence of the specified element.

`POS(`*list, element*`)`

Example:

`POS ({3,7,12,19},12)` returns `3`

REVERSE

Creates a list by reversing the order of the elements in a list.

`REVERSE(`*list*`)`

Example:

`REVERSE({1,2,3})` returns `{3,2,1}`

SIZE

Calculates the number of elements in a list.

`SIZE(`*list*`)`

Also works with matrices.

Example:

`SIZE({1,2,3})` returns `3`

ΣLIST

Calculates the sum of all elements in a list.

`ΣLIST(`*list*`)`

Example:

`ΣLIST({2,3,4})` returns `9`.

SORT

Sorts the elements in a list in ascending order.

`SORT(`*list*`)`

Example:

`SORT({2,5,3})` returns `{2,3,5}`

Finding statistical values for lists

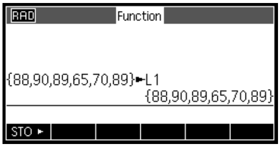
To find values such as the mean, median, maximum, and minimum of a list, use the Statistics 1Var app.

Example

In this example, use the Statistics 1Var app to find the mean, median, maximum, and minimum values of the elements in the list L1.

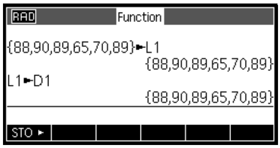
1. Create L1 with values 88, 90, 89, 65, 70, and 89.

   SHIFT {88 Mem , 90
   Mem , 89 Mem , 65
   Mem , 70 Mem , 89
   SHIFT } STO ▸ ALPHA L1 ENTER ANS



2. In Home, store L1 into D1. You will then be able to see the list data in the Numeric view of the Statistics 1Var app.

   ALPHA L1
   STO ▸ ALPHA D1 ENTER ANS

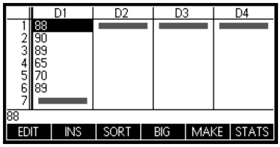


3. Start the Statistics 1Var app.

Apps Info *Select*

`Statistics 1Var`

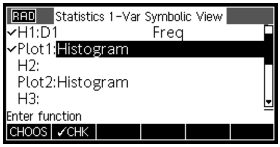
START



Note: your list values are now in column 1 (D1).

4. Select the column upon which to base the statistical calculations. This is done in the Symbolic view.

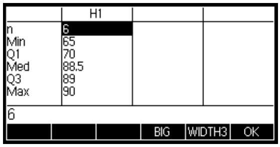
Symb Setup



By default, H1 is defined to use D1, so nothing further remains to be done in the Symbolic view; however, if the data were in D2 or any column other than D1, you would have to enter the desired data column here.

5. Calculate summary statistics.

Num Setup STATS



6. Press OK when you are done.

See the chapter titled, *Statistics 1Var* for the meaning of each computed statistic.

18

Matrices

Introduction

You can perform matrix calculations in Home and in programs. The matrix and each row of a matrix appear in brackets, and the elements and rows are separated by commas. For example, the following matrix:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

is displayed in the history as:
[[1,2,3],[4,5,6]]

You can enter matrices directly in the command line, or create them in the matrix editor.

Vectors

Vectors are one-dimensional arrays. They are composed of just one row. A vector is represented with single brackets; for example, [1,2,3]. A vector can be a real number vector or a complex number vector, for example [(1,2), (7,3)].

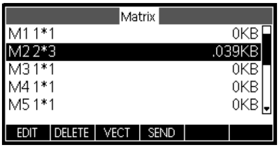
Matrices

Matrices are two-dimensional arrays. They are composed of more than one row and at least one column. Two-dimensional matrices are represented with nested brackets; for example, [[1,2,3],[4,5,6]]. You can create complex matrices, for example, [[(1,2), (3,4)], [(4,5), (6,7)]].

Matrix Variables

There are ten matrix variables available, named `M0` to `M9`. You can use them in calculations in Home or in a program. You can retrieve matrix names from the Vars menu, or just type their names from the keyboard.

Creating and storing matrices



The Matrix Catalog contains the matrix variables `M0-M9`. Once you select a matrix variable to use, you can create, edit, and delete matrices in the Matrix Editor. You may then return to the Matrix Catalog to send your matrix to another HP 39gII.

To open the Matrix catalog, press SHIFT *MATRIX*.

In the Matrix Catalog, a matrix is listed with two dimensions, even if it has only one row. A vector is listed with the number of its elements

You can also create and store matrices—named or unnamed—in Home. For example, the command:

```
POLYROOT([1,0,-1,0])▶M1
```

stores the roots of the complex vector of length 3 into the M1 variable. M1 now contains the three roots of $x^3 - x = 0$

Matrix Catalog keys

The table below lists the operations of keys in the Matrix Catalog.

Key	Meaning
EDIT	Opens the highlighted matrix for editing.
DELETE or Clear	Clears the selected matrix of all data
VECT	Changes the selected matrix into a one-dimensional vector
SEND	Transmits the highlighted matrix to another HP 39gII via USB.
SHIFT *CLEAR*	Clears all matrices.
SHIFT ▼ or ▲	Moves to the end or the beginning of the catalog.

Working with matrices

To start the Matrix Editor

To edit a matrix, go to the Matrix Catalog, highlight the matrix variable name you wish to use, and press the EDIT to enter the Matrix Editor.

Matrix Editor keys

The following table lists the matrix edit key operations.

Key	Meaning
EDIT	Copies the highlighted element to the edit line.
INS	Inserts a row of zeros above, or a column of zeros to the left, of the highlighted cell. You are prompted to choose row or column.
`WIDTHn`	Toggles between showing 1, 2, 3, or 4 columns at a time in the Matrix Editor.
BIG	Switches between larger and smaller font sizes.
GO	A three-way toggle for cursor advancement in the Matrix editor. GO→ advances to the right, GO↓, advances downward, and GO does not advance at all.
Clear	Deletes the highlighted cell, replacing it with a zero.
SHIFT *CLEAR*	Deletes the highlighted row, column, or the entire matrix (you are prompted to make a choice).
SHIFT ▲ ▼ ◀ ▶	Moves to the first row, last row, first column, or last column respectively.

To create a matrix in the Matrix Editor

1. Press SHIFT *MATRIX* to open the Matrix Catalog. The Matrix catalog lists the 10 matrix variables, `M0` to `M9`.
2. Highlight the matrix variable name you want to use and press EDIT or ENTER. Press VECT first if you want to create a vector.
3. For each element in the matrix, type a number or an expression, and press ENTER.

For complex numbers, enter each number in complex form, that is, *(a, b)*, where *a* is the real part and *b* is the imaginary part. You can also enter them in the form, *a+bi*.

4. Upon entry, the highlight moves to the next column in the same row by default. Use the cursor keys to move to a different row or column. You can change the direction of the highlight bar by pressing GO. The GO menu key toggles between the following options:
   – GO↓ specifies that the cursor moves to the cell below the current cell when you press ENTER.
   – GO→ specifies that the cursor moves to the cell to the right of the current cell when you press ENTER.
   – GO specifies that the cursor stays in the current cell when you press ENTER.
5. When done, press SHIFT *MATRIX* to see the Matrix catalog, or press Home to return to Home. The matrix entries are automatically saved.

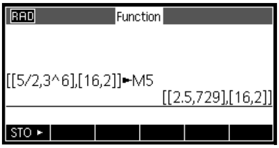
Matrices in the Home view

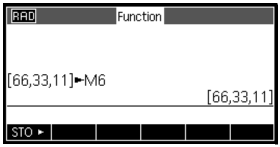
You can enter and operate on matrices directly in the Home view. The matrices you work on in the Home view can be named or not.

1. Enter the vector or matrix on the edit line. Start and end the vector or matrix with square brackets (the shifted 5 and 6 keys). Start each row of a matrix with square brackets as well.
2. Separate each element and each row with a comma.

3. Press ENTER to evaluate and display the vector or matrix. Immediately after entering the matrix, you can store it in a variable by pressing STO ▸ *matrixname*. The matrix variables are M0 through M9.

The left screen below shows the matrix `[[2.5,729],[16,2]]` being stored into M5. The screen on the right shows the vector `[66,33,11]` being stored into M6. Note that you can enter an expression (like 5/2) for an element of the matrix, and it will be evaluated.





To display a matrix

In Home, enter the name of the matrix variable and press ENTER .

To display one element

In Home, enter *matrixname* (*row,column*). For example, if `M2` is `[[3,4],[5,6]]`, then `M2(1,2)` ENTER returns `4`.

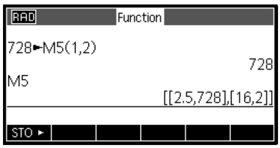
To store one element

In Home, enter *value* STO ▸ *matrixname* (*row, column*). For example, to change the element in the first row and second column of M5 to 728, then display the resulting matrix:

728 STO ▸ ALPHA

M5 (1 , 2)

ENTER ALPHA M5 ENTER



An attempt to store an element to a row or column beyond the size of the matrix results in re-sizing the matrix to allow the storage. Any intermediate cells will be filled with zeroes.

To transmit a matrix

You can send matrices between calculators just as you can send apps, programs, lists, and notes.

1. Connect the two HP 39gII calculators with the micro-USB cable provided with the calculators and turn both calculators on.
2. Open the Matrix catalog on the sending calculator.
3. Highlight the matrix or vector to send.
4. Press [SEND].
5. The transfer will occur immediately.
6. Open the List Catalog on the receiving calculator to see the new list.

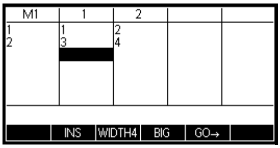
Matrix arithmetic

You can use the arithmetic functions (+, –, ×, / and powers) with matrix arguments. Division left-multiplies by the inverse of the divisor. You can enter the matrices themselves or enter the names of stored matrix variables. The matrices can be real or complex.

For the next examples, store [[1,2],[3,4]] into M1 and [[5,6],[7,8]] into M2.

Example

1. Create the first matrix.

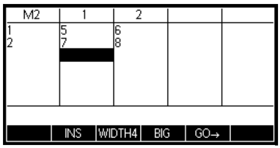


[SHIFT] *MATRIX* [EDIT] 1

[ENTER] 2 [ENTER] [▼] [◄]

[◄] 3 [ENTER] 4 [ENTER]

2. Create the second matrix.



[SHIFT] *MATRIX* [▼] [EDIT]

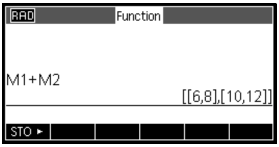
5 [ENTER] 6 [ENTER] [▼] [◄]

[◄] 7 [ENTER] 8 [ENTER]

3. Add the matrices that you created.

   Home/Modes ALPHA M1

   + ALPHA M2 ENTER/ANS

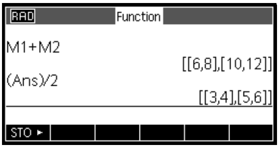


To multiply and divide by a scalar

For division by a scalar, enter the matrix first, then the operator, then the scalar. For multiplication, the order of the operands does not matter.

The matrix and the scalar can be real or complex. For example, to divide the result of the previous example by 2, press the following keys:

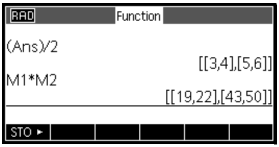
÷ 2 ENTER/ANS



To multiply two matrices

To multiply the two matrices M1 and M2 that you created for the previous example, press the following keys:

ALPHA M1 × ALPHA M2 ENTER/ANS

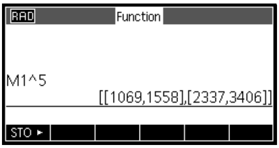


To multiply a matrix by a vector, enter the matrix first, then the vector. The number of elements in the vector must equal the number of columns in the matrix.

To raise a matrix to a power

You can raise a matrix to any power as long as the power is an integer. The following example shows the result of raising matrix M1, created earlier, to the power of 5.

ALPHA M1 x^y 5 ENTER/ANS

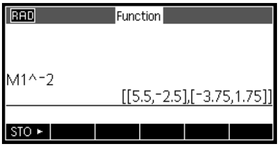


Note: you can also raise a matrix to a power without first storing it as a variable.

Matrices can be raised to negative powers. In this case, the result is equivalent to 1/[matrix]^ABS(power). In the following example, M1 is raised to the power of –2.

ALPHA M1 x^y (−)

2 ENTER



To divide by a square matrix

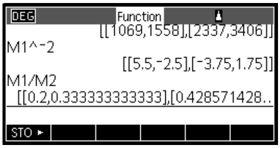
For division of a matrix or a vector by a square matrix, the number of rows of the dividend (or the number of elements, if it is a vector) must equal the number of rows in the divisor.

This operation is not a mathematical division: it is a left-multiplication by the inverse of the divisor. M1/M2 is equivalent to $M2^{-1}$ * M1.

To divide the two matrices M1 and M2 that you created for the previous example, press the following keys:

ALPHA M1 ÷ ALPHA M2

ENTER



To invert a matrix

You can invert a *square matrix* in Home by typing the matrix (or its variable name) and pressing SHIFT x^{-1} ENTER . Or you can use the matrix `INVERSE` command (-1) from the Matrix category of the Math menu.

To negate each element

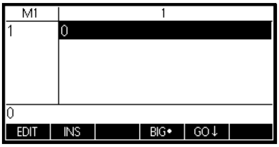
You can change the sign of each element in a matrix by pressing (−) before the matrix name.

Solving systems of linear equations

Solve the following linear system:

$$\begin{aligned} 2x + 3y + 4z &= 5 \\ x + y - z &= 7 \\ 4x - y + 2z &= 1 \end{aligned}$$

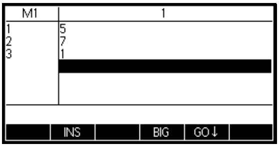
1. Open the Matrix catalog and create a vector.

   SHIFT *MATRIX* Clear

   OK VECT EDIT

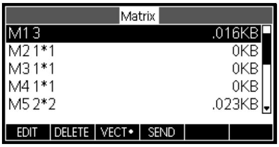
2. Create the vector of the constants in the linear system.

   5 ENTER 7 ENTER

   1 ENTER

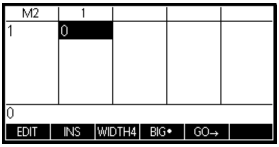
3. Return to the Matrix Catalog.

   SHIFT *MATRIX*

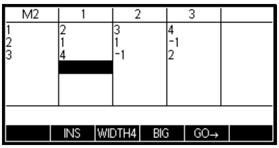
   In this example, the vector you created is listed as M1.

4. Create a new matrix.

   ▼ Clear OK EDIT

5. Enter the equation coefficients.

   2 ENTER 3 ENTER

   4 ENTER ▼ ◄ ◄ ◄

   1 ENTER 1 ENTER

   (−) 1 ENTER 4 ENTER

   (−) 1 ENTER 2 ENTER

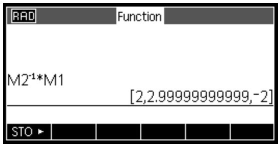
   In this example, the matrix you created is listed as M2.

6. Return to Home and enter the calculation to left-multiply the constants vector by the inverse of the coefficients matrix.

Home ALPHA M2

SHIFT x^{-1} ×

ALPHA M1 ENTER



The result is a vector of the solutions *x* = 2, *y* = 3 and *z* = –2.

An alternative method, is to use the RREF function.

Matrix functions and commands

About functions

- Functions can be used in any app or in Home. They are listed in the Math menu under the Matrix category. They can be used in mathematical expressions—primarily in Home—as well as in programs.
- Functions always produce and display a result. They do not change any stored variables, such as a matrix variable.
- Functions have arguments that are enclosed in parentheses and separated by commas; for example, `CROSS`(*vector1*,*vector2*). The matrix input can be either a matrix variable name (such as `M1`) or the actual matrix data inside brackets. For example, `CROSS(M1,[1,2])`.

About commands

Matrix commands are listed in the CMDS menu (SHIFT *CMDS*), in the matrix category.

See the chapter titled *Programming* for more information on matrix commands.

Functions differ from commands in that a function can be used in an expression. Commands cannot be used in an expression.

Argument conventions

- For *row#* or *column#*, supply the number of the row (counting from the top, starting with 1) or the number of the column (counting from the left, starting with 1).
- The argument *matrix* can refer to either a vector or a matrix.

Matrix functions

COLNORM

Column Norm. Finds the maximum value (over all columns) of the sums of the absolute values of all elements in a column.

`COLNORM`(*matrix*)

COND

Condition Number. Finds the 1-norm (column norm) of a square *matrix*.

`COND`(*matrix*)

CROSS

Cross Product of *vector1* with *vector2*.

`CROSS`(*vector1*, *vector2*)

DET

Determinant of a square *matrix*.

`DET`(*matrix*)

DOT

Dot Product of two arrays, matrix1 and matrix2.

`DOT`(matrix1, matrix2)

EIGENVAL

Displays the eigenvalues in vector form for *matrix*.

`EIGENVAL`(*matrix*)

EIGENVV

Eigenvectors and Eigenvalues for a square *matrix*. Displays a list of two arrays. The first contains the eigenvectors and the second contains the eigenvalues.

`EIGENVV`(*matrix*)

IDENMAT

Identity matrix. Creates a square matrix of dimension *size* × *size* whose diagonal elements are 1 and off-diagonal elements are zero.

`IDENMAT`(*size*)

INVERSE

Inverts a square matrix (real or complex).

`INVERSE`(*matrix*)

LQ

LQ Factorization. Factors an $m \times n$ *matrix* into three matrices:
{[[$m \times n$ *lowertrapezoidal*]],[[$n \times n$ *orthogonal*]],
[[$m \times m$ *permutation*]]}.

`LQ`(*matrix*)

LSQ

Least Squares. Displays the minimum norm least squares *matrix* (or *vector*).

`LSQ`(*matrix1, matrix2*)

LU

LU Decomposition. Factors a square *matrix* into three matrices:
{[[*lowertriangular*]],[[*uppertriangular*]],[[*permutation*]]}
The *uppertriangular* has ones on its diagonal.

`LU`(*matrix*)

MAKEMAT

Make Matrix. Creates a matrix of dimension *rows* × *columns*, using *expression* to calculate each element. If *expression* contains the variables I and J, then the calculation for each element substitutes the current row number for I and the current column number for J.

`MAKEMAT`(*expression, rows, columns*)

Example

`MAKEMAT(0,3,3)` returns a 3×3 zero matrix,
`[[0,0,0],[0,0,0],[0,0,0]]`.

QR

QR Factorization. Factors an *m*×*n matrix* into three matrices: {[[*m*×*m orthogonal*]],[[*m*×*n uppertrapezoidal*]],[[*n*×*n permutation*]]}.

`QR`(*matrix*)

RANK

Rank of a rectangular *matrix*.

`RANK`(*matrix*)

ROWNORM

Row Norm. Finds the maximum value (over all rows) for the sums of the absolute values of all elements in a row.

`ROWNORM`(*matrix*)

RREF

Reduced-Row Echelon Form. Changes a rectangular *matrix* to its reduced row-echelon form.

`RREF`(*matrix*)

SCHUR

Schur Decomposition. Factors a square *matrix* into two matrices. If *matrix* is real, then the result is {[[*orthogonal*]],[[*upper-quasi triangular*]]}.
If *matrix* is complex, then the result is {[[*unitary*]],[[*upper-triangular*]]}.

`SCHUR`(*matrix*)

SIZE

Dimensions of *matrix*. Returned as a list: {rows,columns}.

`SIZE`(*matrix*)

SPECNORM

Spectral Norm of *matrix*.

`SPECNORM`(*matrix*)

SPECRAD

Spectral Radius of a square *matrix*.

`SPECRAD`(*matrix*)

SVD

Singular Value Decomposition. Factors an $m \times n$ *matrix* into two matrices and a vector:
{[[$m \times m$ *square orthogonal*]],[[$n \times n$ *square orthogonal*]], [*real*]}.

`SVD`(*matrix*)

SVL

Singular Values. Returns a vector containing the singular values of *matrix*.

`SVL`(*matrix*)

TRACE

Finds the trace of a square *matrix*. The trace is equal to the sum of the diagonal elements. (It is also equal to the sum of the eigenvalues.)

`TRACE`(*matrix*)

TRN

Transposes *matrix*. For a complex matrix, TRN finds the conjugate transpose.

`TRN`(*matrix*)

Examples

Identity Matrix

You can create an identity matrix with the IDENMAT function. For example, IDENMAT(2) creates the 2×2 identity matrix [[1,0],[0,1]].

You can also create an identity matrix using the MAKEMAT (*make matrix*) function. For example, entering MAKEMAT(I ≠ J,4,4) creates a 4 × 4 matrix showing the numeral 1 for all elements except zeros on the diagonal. The logical operator (≠) returns 0 when I (the row number) and J (the column number) are equal, and returns 1 when they are not equal.

Transposing a Matrix

The TRN function swaps the row-column and column-row elements of a matrix. For instance, element 1,2 (row 1, column 2) is swapped with element 2,1; element 2,3 is swapped with element 3,2; and so on.

For example, `TRN([[1,2],[3,4]])` creates the matrix `[[1,3],[2,4]]`.

Reduced-Row Echelon Form

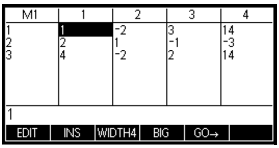
The following set of equations

$$\begin{aligned} x - 2y + 3z &= 14 \\ 2x + y - z &= -3 \\ 4x - 2y + 2z &= 14 \end{aligned}$$

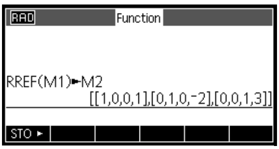
can be written as the augmented matrix

$$\left[\begin{array}{ccc|c} 1 & -2 & 3 & 14 \\ 2 & 1 & -1 & -3 \\ 4 & -2 & 2 & 14 \end{array}\right]$$

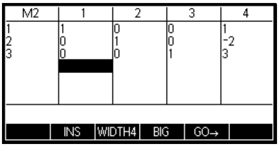
which can then be stored as a 3×4 real matrix in any matrix variable. M1 is used in this example.



You can use the RREF function to change this to reduced row echelon form, storing it in any matrix variable. M2 is used in this example.



The reduced row echelon matrix gives the solution to the linear equation in the fourth column.

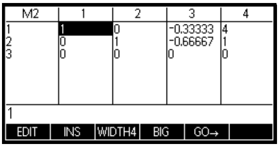


An advantage of using the `RREF` function is that it will also work with inconsistent matrices resulting from systems of equations which have no solution or infinite solutions.

For example, the following set of equations has an infinite number of solutions:

$$x + y - z = 5$$
$$2x - y = 7$$
$$x - 2y + z = 2$$

The final row of zeros in the reduced-row echelon form of the augmented matrix indicates an inconsistent system with infinite solutions.



19

Notes and Info

The HP 39gII has text editors for entering notes. There are two text editors:

- The Notes Editor runs from within the Notes Catalog, a collection of notes independent of apps. These notes can be sent to another calculator from the Note Catalog.
- The Info Editor runs from the Info view of an app. A note created in the Info view is associated with the app. When you save the app or send it to another calculator, this note is saved or sent as well.

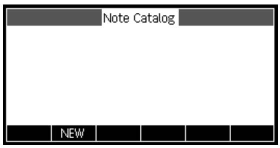
The Notes Catalog

Subject to available memory, you can store as many notes as you want in the Note Catalog. These notes are independent of any app. The Note Catalog lists the existing entries by name. The list does not include notes that were created in any apps' Info view, but these can be copied and pasted using the clipboard. From the Note Catalog, you create or edit individual notes in the Note Editor.

To create a note in the Note Editor

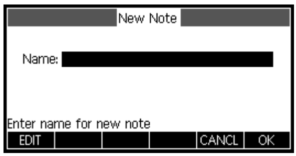
1. Open the Note Catalog.

   SHIFT *NOTES*



2. Create a new note.

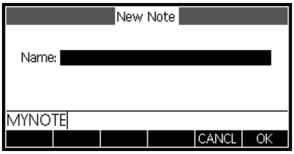
   NEW



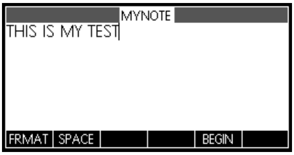
3. Enter a name for your note.

   ALPHA ALPHA MYNOTE

   OK OK



4. Write your note, using the note editing keys and formatting options shown in the following sections.



   Press Home/Modes when you are finished, or press an app key to exit the Note Editor. Your work is automatically saved. To access your new note, return to the Note Catalog.

   While you are in the Note Catalog, you can use the following keys.

Notes Catalog keys

Key	Meaning
EDIT	Opens the selected note for editing.
NEW	Begins a new note, and asks for a name.
SEND	Transmits the selected note to another HP 39gII or PC.
DELETE or Clear	Deletes the selected note.
SHIFT *CLEAR*	Deletes all notes in the catalog.

To create a note in the Info view

1. In an app, press SHIFT Info for the Info view and EDIT to start your note.
2. Use the note editing keys and formatting options. These are identical to those found in the Note Editor (see previous section).Your work is automatically saved. To exit Info view, press any view key or Home Modes.

Note Editor keys

While you are in the Note or Info Editors, you can use the following keys:

Key	Meaning
FRMAT	Opens the text formatting menu. See *Formatting options* later in this chapter.
SPACE	Space key for text entry.
▲ PAGE PAGE ▼	Moves from page to page in a multi-page note.
BEGIN	Starts text selection. Use the arrow keys to select existing text for formatting.
← Clear	Backspaces cursor and deletes character.
ENTER ANS	Starts a new line.
SHIFT *CLEAR*	Erases the entire note.
Vars Chars A	Menu for entering variable names, and contents of variables.
Math Cmds B	Menu for entering math operations, and constants.
SHIFT *CMDS*	Menu for entering program commands.

Key	Meaning (Continued)
SHIFT *CHARS*	Displays special characters. To type one, highlight it and press OK. To copy a character *without* closing the Chars menu, press ECHO.

Entering alphanumeric characters

While in the Note or Info editors, you will want to enter upper-case and lower-case alphabetical characters. The table below describes the various options available for entering these characters.

Purpose	Keystroke
Upper-case alpha shift (one character)	ALPHA
Upper-case alpha lock	ALPHA ALPHA
Lower-case alpha shift	ALPHA SHIFT
Lower-case alpha lock	ALPHA SHIFT ALPHA

To release upper-case or lower-case alpha lock, just press ALPHA one more time. While in an alpha lock, you can switch cases for one keystroke by pressing SHIFT ; to switch cases and lock, press SHIFT ALPHA .

Text formatting

You can format text in any Note or Info. To format existing text, follow these steps:

1. Open the Note or Info view.
2. Move the cursor to the beginning of the text you wish to format.
3. Press BEGIN .

4. Move the cursor to the end of the text you wish to format.
5. Press FRMAT to open the Format menu.

   The Format menu is a two-column menu. The left column contains a list of categories and the right column lists the formatting options within each category. Select a category on the left and then a formatting option on the right.
6. Use ▲ and ▼ to select the formatting category. Use ▶ to switch to the right column and then use ▲ and ▼ again to select the formatting option you wish to apply to the selected text.
7. Press OK to apply or CANCL to cancel.

Formatting Options

The formatting options are listed in the table below.

Category	**Options**
Align Text alignment	• Left • Center • Right
Font Font size	• Small • Large
FG Color Foreground color	• Black • Dark Gray • Light Gray • White
BG Color Background color	• Black • Dark Gray • Light Gray • White

Category	Options
Bullets	• Level 1 • Level 2 • Level 3
Style Font style	• Underline • Strikethrough • Superscript • Subscript • Normal

To import a note

You can import a note from the Note Catalog into an app's Info view and vice versa.

Suppose you want to copy a note named *Assignments* from the Note Catalog into the Function Info view:

1. Open the note `Assignment`.

   SHIFT *NOTES*

2. Move the cursor to the beginning of the text you wish to copy and begin text selection.

   BEGIN

3. Move the cursor to the end of the text you wish to format.
4. Copy the selected text to the clipboard.

   SHIFT *COPY*

5. Open the app's Info view

   Apps Info *Select* `Function` OK

   SHIFT *INFO*

6. Move the cursor to the location where you want the copied text to be pasted and open the clipboard.

   SHIFT *PASTE*

7. Select the text from the clipboard and press OK.

To import a graphics variable

You can copy the contents of a graphics variable into a note or the Info view of an app.

1. Open the note or the Info view of the app. Place the insert cursor where you want the graphic to appear. The graphic will be copied here.
2. Press Vars Chars A.
3. Highlight `Graphic`, then press ▶ and highlight the name of the variable (`G1`, etc.).
4. Press VALUE to recall the contents of the graphic variable and then press OK.

To transmit a note

You can send notes between calculators just as you can send apps, programs, matrices, and lists.

1. Connect the two HP 39gII calculators with the micro-USB cable provided with the calculators and turn both calculators on.
2. Open the Notes Catalog on the sending calculator.
3. Highlight the name of the note to send.
4. Press SEND.
5. The transfer will occur immediately.
6. Open the Notes Catalog on the receiving calculator to see the new list.

20

Variables and memory management

Introduction

The HP 39gII has approximately 250Kb of user memory, as well as 80Mb of flash memory.You use the calculator's memory to store the following objects:

- copies of apps with specific configurations
- new apps that you download
- home variables
- app variables
- user-defined variables
- variables created through a catalog or editor, for example a matrix or a text note
- programs that you create.

A variable is an object that you create in memory to hold data. The HP39gII has three types of variables: Home variables, App variables, and User variables.

- Home variables are available in all apps. For example, you can store real numbers in variables *A* to *Z* and complex numbers in variables `Z0` to `Z9`. These can be numbers you have entered, or the results of calculations. These variables are available within all apps and within any programs.
- App variables apply only to a single app. Apps have specific variables allocated to them which vary from app to app.
- User variables are added to the Vars menu via programs. These variables can be either local to the program or global. See *Programming* for more details.

You can use the Memory Manager (SHIFT *MEMORY*) to view the amount of memory available. The catalog views, which are accessible via the Memory Manager, can be used to transfer variables such as lists or matrices between calculators.

Storing and recalling variables

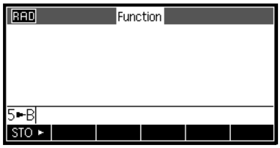
You can store numbers or expressions from a previous input or result into variables.

Numeric Precision

A number stored in a variable is always stored as a 12-digit mantissa with a 3-digit exponent. Numeric precision in the display, however, depends on the display mode (Standard, Fixed, Scientific, or Engineering). A displayed number has only the precision that is displayed. If you copy it from the Home view display history, you obtain only the precision displayed, not the full internal precision. On the other hand, the variable *Ans* always contains the most recent result to full precision.

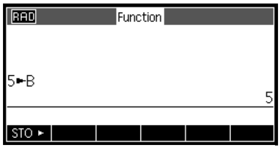
To store a value

1. In the Home view, enter a value, expression or object, followed by the Store command.



   Home/Modes 5 STO ►

2. Enter a name for the variable suitable for the object.



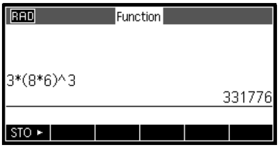
   ALPHA B ENTER/ANS

To store the results of a calculation

If the value you want to store is the last result just calculated, then just press STO ►, followed by the variable name and press ENTER/ANS . If the value you want to store is further up in the Home view display history, then use ▲ to highlight the value, COPY to copy it to the command line, and then proceed to store it.

The following example illustrates the procedure.

1. Perform the calculation for the result you want to store.



3 [×] [(] 8 [×] 6 [)] [x^y] 3 [ENTER]

2. Highlight the result you wish to store

[▲]

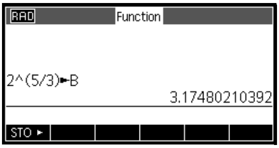
3. Copy the result to the edit line

COPY

4. Store the result

STO ► [ALPHA] A [ENTER]

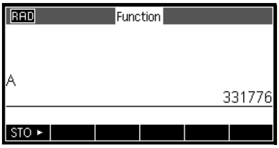
The results of a calculation can also be stored directly to a variable. For example:



2 [x^y] [(] 5 [÷] 3 [)] STO ► [ALPHA] B [ENTER]

To recall a value

To recall a variable's value, type the name of the variable and press [ENTER] .

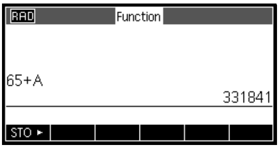


[ALPHA] A [ENTER]

To use variables in calculations

You can use variables in calculations. The calculator substitutes the variable's value in the calculation:

65 [+] [ALPHA] A [ENTER]

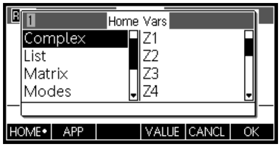


The Vars menu

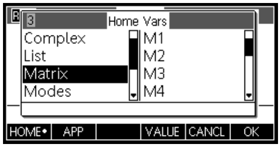
You use the Vars menu to access all variables in the calculator. There are menu keys for Home, App, and User variables. When you press Vars Chars A, the Vars menu opens with the Home variables menu open by default. The Vars menu is organised by category. For each variable category in the left column, there is a list of variables in the right column. You select a category and then select a variable in the category.

1. Open the Vars menu and press HOME

   Vars Chars A HOME



2. Use the cursor keys or press the number of the category (1-5) to select a variable category. In the figure to the right, the Matrix category has been selected.

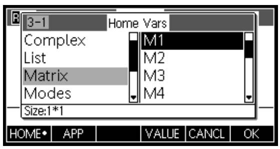


3. Move the highlight to the variables column.

   ▶

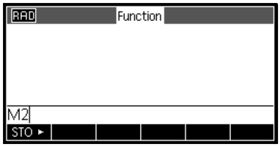
4. Use the cursor keys to select the variable that you want. For example, to select M2, press ▼.

   ▼



5. Choose whether to place the variable name or the variable contents on the command line.
   - Press VALUE to indicate that you want the variable's contents to appear on the command line.
   - Press OK to indicate that you want the variable's name to appear on the command line.
6. Press OK to place the contents or name on the command line. The selected object appears on the command line.

   OK



   Note: the Vars menu can also be used to enter the names or values of variables into programs.

Example

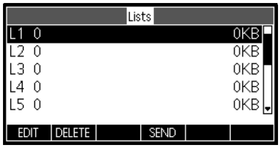
This example demonstrates how to use the Vars menu to add the contents of two list variables, and to store the result in another list variable.

1. Display the List Catalog.

   SHIFT *LIST*

   to select `L1`

   EDIT

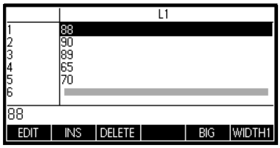


2. Enter the data for `L1`.

   88 OK 90 OK

   89 OK 65 OK

   70 OK



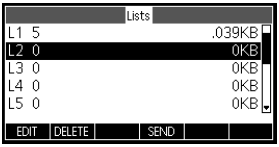
   Note: you can press BIG• *for the smaller font. Press*

   ▲ *to scroll up and view the data you entered.*

3. Return to the List Catalog to create L2.

SHIFT *LIST*

▼ *to select* L2 EDIT

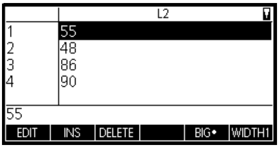


4. Enter data for L2.

55 OK 48 OK

86 OK 90 OK

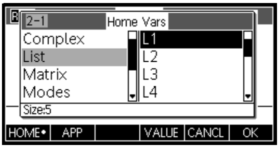
77 OK



5. Press Home/Modes to access Home.

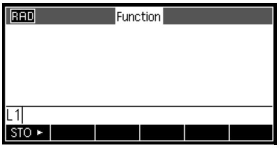
6. Open the variable menu and select L1.

Vars/Chars A ▲ ▶



7. Copy it to the command line.

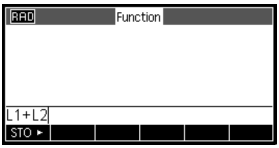
OK



8. Insert the + operator and select the L2 variable from the List variables.

+ Vars/Chars A ▶ ▼

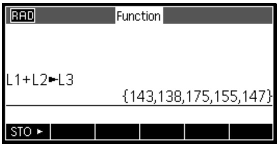
OK



9. Store the answer in the List catalog L3 variable.

STO ▸ ALPHA L3 ENTER

Note: you can also type list names directly from the keyboard.



Home variables

The following table lists the categories of Home variables and the available variable names in each category.

It is not possible to store data of one type in a variable of another type. For example, you use the Matrix catalog to create matrices. You can create up to ten matrices, and you can store these in variables `M0` to `M9`. You cannot store matrices in variables other than `M0` to `M9`.

Category	**Available names**
Complex numbers	Z0 to Z9 To store a complex number, enter it in the form $a + b*i$. For example, $2 + 3*1$ STO ▸ Z1.
Lists	L0 to L9 For example, {1,2,3} STO ▸ L1.
Matrices	M0 to M9 Store matrices and vectors in these variables. See the chapter *Matrices* for more information on matrices and vectors. For example, [[1,2],[3,4]] STO ▸ M1.
Mode settings	Modes variables store the modes settings in SHIFT *MODES*.
Programs	Program variables store programs.
Real numbers	A to Z and θ For example, 7.45 STO ▸ A.

App variables

Most app variables store values that are unique to a particular app. These include symbolic expressions and equations, settings for the Plot and Numeric views, and the results of some calculations such as roots and intersections.
See *Reference Information* for a complete listing of app variables and *Programming* for more information about using app variables in programs.

To access an app variable

1. Open the app that contains the variable you want.

   Apps/Info *Select Function*

2. Go to where you want to paste the variable.

   Home/Modes

3. Open the Vars menu and switch to the App Vars menu.

   Views/Help

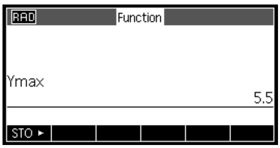
   APP (to select App Vars)

4. Use the cursor keys to select the view and then the variable you want.

   ► ▼ ▼ (to select Plot)

   ► ▼ ▼ ▼ (to select Ymax)

5. To copy the variable name to the edit line, press OK; to copy the variable contents, press VALUE and OK.



You can qualify the name of any app variable so that it can be accessed from anywhere on the HP 39gII. For example, both the Function app and the Parametric app have an app variable named Xmin. If you are in the Parametric app and enter Xmin in the Home view, you will see the value of Xmin from the Parametric app. To access the value of Xmin in the Function app, you must either start the Function app (as above) or qualify the name by entering Function::Xmin. For more information on qualifying variable names, see the chapter *Programming*.

User variables

The HP 39gII supports both user-defined functions and user-defined variables. Both of these object types can be local (within an app or a program) or global (visible and accessible anywhere on the calculator). For more information about creating and using user-defined variables and functions (as well as declaring them local or global), see the chapter *Programming*.

Memory Manager

Use the Memory Manager to view the amount of available memory and organize it. If the available memory is low, use the Memory Manager to determine which variables you might delete to free up memory. You can also use the Memory manager to send sets of variables to another HP 39gII or to clone your entire memory to another HP 39gII.

Memory manager keys

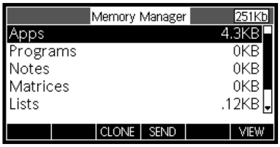
Start the Memory Manager by pressing SHIFT *MEMORY*. When the Memory Manager is open the keys listed in the table on the following page are available to you:

Key	**Meaning**
CLONE	Replace the memory of a connected HP 39gII with the current memory of the cloning 39gII.
SEND	Sends all variables of the selected type (lists, matrices, etc.) to another HP 39gII.
VIEW	Opens the catalog or library of the selected variable type.

Key	Meaning
Clear	Deletes the contents of all variables of the selected type.
SHIFT Clear	Deletes all memory.

Example

1. Start the Memory Manager. A list of variable categories is displayed.



SHIFT *MEMORY*

Free memory is displayed in the top right corner and the body of the screen lists each category of variable and the total memory used by the variables of that type.

2. Select a category and press VIEW. Memory Manager opens the selected catalog or library so you can edit, delete, or clear variables of a selected type. To delete variables in a category:

   – Press Clear to delete the selected variable.

   – Press SHIFT *CLEAR* to delete all variables in the selected category.

To send all variables of a single type

You can send all the variables of a single type (all lists, matrices, programs, notes, etc.) from your HP 39gII to another HP 39gII or a PC. To send variables of a single type between two HP 39gII calculators:

1. Connect the two HP 39gII calculators with the micro-USB cable provided with the calculators and turn both calculators on.
2. Open the Memory Manager on the sending calculator.
3. Use ▼ and ▲ to highlight the variable type to send.
4. Press SEND.
5. The transfer will occur immediately.
6. Open the Memory Manager on the receiving calculator to see the new variables.

To clone your HP 39gII

You can clone the entire memory of your HP 39gII to another HP 39gII calculator, effectively copying your HP 39gII to another HP 39gII. This is helpful if you want to backup your calculator's memory, or in settings where calculators in a classroom or in a group require similar configuration. To clone your HP 39gII:

1. Connect the two HP 39gII calculators with the micro-USB cable provided with the calculators and turn both calculators on.
2. Open the Memory Manager on the sending calculator.
3. Press CLONE.
4. You will see the transfer annunciator flash briefly.
5. The cloned HP 39gII is now ready for use.

21

Programming

Introduction

This chapter describes how to program the HP 39gII. In this chapter you'll learn about:

- programming commands
- writing functions in programs
- using variables in programs
- executing programs
- debugging programs
- creating programs for building custom apps
- sending a program to another HP39gII

HP 39gII Programs

An HP 39gII program contains a sequence of commands that execute automatically to perform a task.

Command Structure

Commands are separated by a semicolon (;). Commands that take multiple arguments have those arguments enclosed in parentheses and separated by a comma(,). For example,

`PIXON` (*xposition*, *yposition*);

Sometimes, arguments to a command are optional. If an argument is omitted, a default value is used in its place. In the case of the PIXON command, a third argument could be used that specifies the color of the pixel:

`PIXON` (*xposition*, *yposition* [, *color*]);

The last argument indicates which of four colors to use when lighting up the pixel. Here, the default value is 0 (black). In this manual, optional arguments to commands appear inside square brackets, as shown above. In the `PIXON` example, a graphic variable (G) could be specified as the first argument. The default is G0, which always contains the currently displayed screen. Thus, the full syntax for the PIXON command is:

`PIXON`([*G*,] *xposition, yposition* [,*color*]);

Some built-in commands employ an alternate syntax, whereby function arguments do not appears in parentheses. Examples include `RETURN` and `RANDOM`.

Program Structure

Programs can contain any number of subroutines (each of which is a function or procedure). Subroutines start with a heading consisting of the name, followed by parentheses that contain a list of parameters or arguments, separated by commas. The body of a subroutine is a sequence of statements enclosed within a BEGIN END; pair. For example, the body of a simple program, called `MYPROGRAM`, could look like this:

```
EXPORT MYPROGAM()

BEGIN

PIXON(1,1);

END;
```

Comments

When a line of a program begins with two slashes, //, the rest of the line will be ignored. This allows the programmer to insert comments in the program:

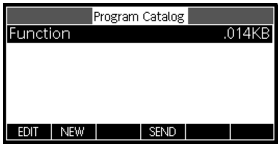
```
EXPORT MYPROGAM()

BEGIN

PIXON(1,1);

//This line is just a comment.

END;
```

The Program Catalog

The Program catalog is where you name, delete, send, or run programs. This is also where you start the Program Editor, where you create and edit programs. Programs can also be run from the Home view or other programs.

Open the Program Catalog

Press SHIFT *PRGM* to open the Program Catalog.



The Program Catalog displays a list of program names. The first item in the Program Catalog is a built-in entry that has the same name as the active app. This entry is the app program for the active app, if such a program exists. See the section on *App Programming*.

Before starting to work with programs, you should take a few minutes to become familiar with the Program Catalog menu keys. You can use any of the following keys (both menu and keyboard) to perform tasks in the Program Catalog.

Program catalog keys

The program catalog keys are:

Key	Meaning
EDIT	Opens the highlighted program for editing.
NEW	Prompts for a new program name, then opens an empty program. Not displayed if the app program is selected
SEND	Transmits the highlighted program to another HP 39gII or to a PC.
RUN	Runs the highlighted program.

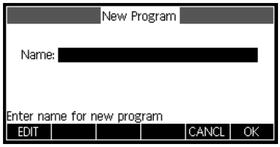
Key	Meaning
SHIFT ▲ or SHIFT ▼	Moves to the beginning or end of the Program catalog.
Clear	Deletes the highlighted program.
SHIFT Clear	Deletes all programs.

Creating a New Home Program

1. Open the Program catalog and start a new program.

   SHIFT *PRGM*

   NEW

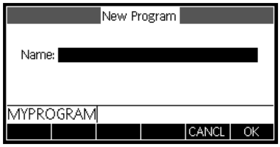


2. The HP 39gII prompts you for a name.

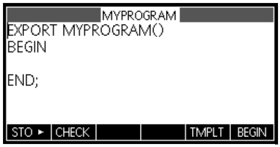
   ALPHA ALPHA for alpha lock

   MYPROGRAM

   OK OK.



   A template for your program is automatically created. The template consists of a heading for a function with the same name as the program, `EXPORT MYPROGRAM()`, and a `BEGIN...END` pair that blocks off the statements for the function.



HINT A program name can contain only alphanumeric characters (letters and numbers) and the underscore character. The first character must be a letter. For example, `GOOD_NAME` and `Spin2` are valid program names, while `HOT STUFF` (no space allowed) and `2Cool!` (starts with number and no !) are not valid.

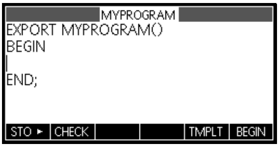
The Program Editor

Until you become familiar with the HP 39gII commands, the easiest way to enter commands is to select them from menus.

Entering a program

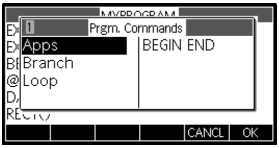
1. Position the cursor where you want the command to go using the navigation keys.

   ▼ ▼



2. Press **TMPLT** to open the Program Templates menu.

   TMPLT



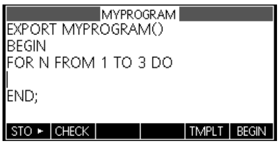
The Program Templates menu contains structures that control execution flow, such as `IF THEN` statements and `FOR` loops. Use the cursor keys to highlight a command and press **OK** to paste the command into the program at the cursor position.

3. Insert a `FOR` loop.

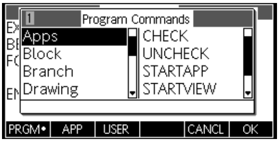
   ▼ ▼ *Select* `Loop`

   ▶ *Select* `FOR` **OK**

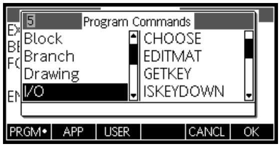
   Again, a template is inserted. Use the keyboard to fill in the missing parts of the command, then position the cursor on the blank line after the `FOR` command. In this case, complete the statement "`FOR N FROM` 1 `TO` 3 `DO`".



Press SHIFT *CMDS* to bring up the complete menu of Program Commands. On the left, use ▼ or ▲ to highlight a command category, then press ▶ to access the commands in the category. Select the command that you want and press OK to paste the command into the program. You can also use keyboard shortcuts indicated in the menu title bar in the Program Commands menu to quickly select a command.

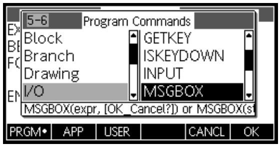


4. Insert the `MSGBOX` (Message Box) command.



SHIFT *CMDS*

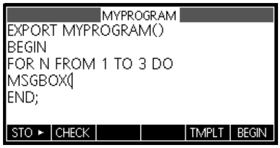
▼ ▼ ▼ ▼



(or enter `5`)

Select `I/O`

▶ (switch columns)

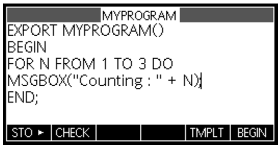


▼ ▼ ▼ ▼ ▼

(or enter `6`)

Select *`MSGBX`* OK

5. Fill in the arguments to the `MSGBOX` command, and type a semicolon at the end of the command.



HINT Use the Characters menu to enter the quote, ("). Press SHIFT *CHARS*, highlight the quote character, and press ENTER.

HINT For lower-case alpha lock, press: ALPHA ALPHA SHIFT ALPHA.

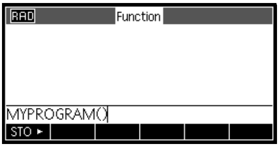
When you are done, press SHIFT *PRGM* to return to the Program Catalog or Home to go to the Home view. You can also press any of the app-control keys to enter the current app's views. You are ready now to execute the program.

Run a Program

From Home, type the name of the program, with open and closing parentheses after it. If the program takes any arguments, insert these in the parentheses, separated by commas. Press ENTER.

From the Program Catalog, highlight the program you want to run and press RUN. When a program is executed from the catalog, the system looks for a function named `START()` (no parameters). If it finds one, that function is executed. Otherwise, it looks for a function with the same name as the program. If it finds that, it executes. Otherwise, nothing happens when RUN is pressed.

1. Run MYPROGRAM.



   Home SHIFT *CMDS*

   USER

   Select MYPROGRAM

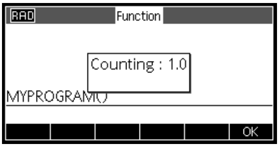
   ▶ (switch columns) Select MYPROGRAM

   OK ( ) ENTER

The program executes, displaying a message box.

2. Press OK three times to see the `FOR` loop finish.

OK OK OK



3. After the program terminates, you can resume any other activity with the HP39gII.

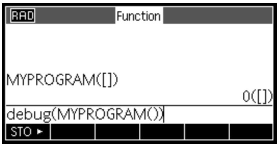
Regardless of where you start the program, all programs run in Home. What you see will differ slightly depending on where you started the program. If you start the program from Home, the HP 39gII displays the contents of `Ans` (Home variable containing the last result), when the program has finished. If you start the program from the Program catalog using the RUN key, the HP 39gII returns you to the Program catalog when the program ends.

Debug a Program

You cannot run a program that contains syntax errors. You must first correct all the syntax errors before executing the program.

If there is an error detected at run-time, such as division by zero, the program will stop and you will see an error message. If the program does not do what you expect it to do, or if there is a run-time error detected by the system, you can execute the program step by step, and look at the values of local variables. To do that, type debug(`MYPROGRAM`()) on the edit line.

1. Start the debug tool for the program you just wrote.



Home Modes ALPHA ALPHA

SHIFT ALPHA (for alpha lock)

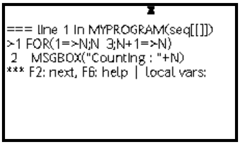
debug (Copy L SHIFT *CMDS* USER

Select MYPROGRAM

(Right arrow key) (switch columns)

Select MYPROGRAM

OK Copy Paste



Paste ENTER

While debugging a program, the **>** indicator at the left of the screen points to the current command. The current value of each local variable is visible at the bottom of the screen. Since there are no local variables in our program, nothing is shown. While in the debugger, the menu keys perform the following actions:

- F2—Execute the next command
- F3—Step in
- F4—Continue
- F5—Stop program execution
- F6—Display Help

2. Execute the first command.

F2

The FOR loop starts and the pointer moves to Line 2.

3. Execute Line 2.

F2 ENTER

The message box appears. Note that when each message box is displayed, you still have to dismiss the message box by pressing ENTER. Press F2 and ENTER repeatedly to execute the program step-by-step.

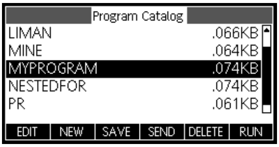
When the next instruction to be executed is a user function or subroutine, pressing F2 will execute that function or subroutine all at once, while pressing F3 will enter in the subroutine and execute it step by step (Step In).

Edit an existing program

To edit an existing program, use the Program Catalog.

1. Open the Program catalog.

   [SHIFT] *PRGM*



2. Use the arrow keys to highlight the program you want to edit, and press [EDIT]. The HP 39gII opens the Program Editor. The name of your program appears in the title bar of the display. You can use the following keys to edit your program.

Editing Keys

Keys	**Meaning**
[STO ▶]	Inserts the STORE character (▶) at the cursor location.
[CHECK]	Checks the current program for errors.
[▲ PAGE]	Displays previous page of the program.
[PAGE ▼]	Displays next page of the program.
[TMPLT]	Displays the catalog of program commands. Select a command and press [ENTER] to insert the command into your program.
▼ ▲	Moves up or down one line.

Keys	**Meaning**
SHIFT ▲ SHIFT ▼	Moves up or down one page
◀ ▶ direction keys	Moves left or right one character.
SHIFT ◀ or SHIFT ▶	Moves to beginning or end of line
ENTER ANS	Starts a new line.
← Clear	Deletes the character to the left of the cursor (Backspace)
SHIFT *CLEAR*	Erases the entire program.
Vars Chars A	Displays menus for selecting variable names, contents of variables, functions names, and constants.
Math Cmds B	Displays menus for selecting mathematical functions, units, and constants.
SHIFT *CMDS*	Displays menus for selecting program conmmands.
SHIFT *CHARS*	Displays all characters. To type one, highlight it and press OK. To enter several characters in a row, ECHO while in the Chars menu.

Copy a program or part of a program

You can use the global `Copy` and `Paste` commands to copy part or all of a program. The following steps illustrate the process:

1. Press SHIFT *PRGM* to open the Program catalog.
2. Highlight the program containing the commands you wish to copy and press EDIT.
3. Move the cursor to the beginning of the commands you wish to copy.
4. Press BEGIN.
5. Move the cursor to the end of the commands you wish to copy. The selected commands will be highlighted as you move the cursor.
6. When all the commands you want are highlighted, press SHIFT *COPY* to copy the selected commands to the clipboard.
7. Return to the Program Catalog and open the target program.
8. Move the cursor to the line where you wish to insert the copied commands.
9. Press SHIFT *PASTE* and the clipboard will open. Your commands will be first in the list and highlighted already, so just press OK. The commands will be pasted into the program, beginning at the cursor location.

Delete a program

To delete a program:

1. Press SHIFT *PRGM* to open the Program catalog.
2. Highlight a program to delete, then Clear or DELETE.
3. Press OK.

Delete all programs

You can delete all programs at once.

1. In the Program catalog, press SHIFT *CLEAR*.

2. Press OK.

Delete the contents of a program

You can clear the contents of a program without deleting the program name.

1. Press SHIFT *PRGM* to open the Program catalog.

2. Highlight a program, then press EDIT.

3. Press SHIFT *CLEAR*, then press OK.

4. The contents of the program are deleted, but the program name remains.

To transmit a program

You can send programs between calculators just as you can send apps, notes, matrices, and lists.

1. Connect the two HP 39gII calculators with the micro-USB cable provided with the calculators and turn both calculators on.
2. Open the Program Catalog on the sending calculator.
3. Highlight the name of the program to send.
4. Press SEND.
5. The transfer will occur immediately.
6. Open the Program Catalog on the receiving calculator to see the new list.

The HP 39gII Programming Language

Variables and visibility

Variables in an HP 39gII program can be used to store numbers, lists, matrices, graphics objects, and strings. The name of a variable must be a sequence of alphanumeric characters (letters and numbers), starting with a letter. Names are case-sensitive, so the variables named MaxTemp and maxTemp would be different.

The HP39gII has built-in variables of various types, visible globally. The following table illustrates many of these, with an example showing how to store a value into the variable:

Type	Names	Store Example
Real number	A-Z and θ	2.7 ▶ R
Complex numbers	Z0-Z9	(2,3) ▶ Z1
Lists	L0-L9 C0-C9 D0-D9	{ 1, 2, 3 ,4} ▶ L1
Matrices	M0-M9	[[1,2],[3,4],[5,6]] ▶ M1
Graphics	G0-G9	See graphics section
Functions	F0-F9	COS(X) ▶ F1

These names are reserved by the system. These (and all other) system variables are visible everywhere, and users may not use the names for other data. That is, you may not name a program L1, for example, nor store a real number into a variable named G1. A full list of system variables appears in the chapter titled, *Reference Information*. Besides these reserved variables, each HP app has its own reserved variables. For more information on these variables, see the section in this chapter *Variables and programs*.

Within a program, you can declare variables for use only within a particular function. This is done using a `LOCAL` declaration. The use of `LOCAL` variables allows the programmer to declare and use variables that will not affect the rest of the calculator. `LOCAL` variables declared by the programmer are not bound to a particular type. That is, you can store floating-point numbers, integers, lists, matrices, and symbolic expressions into a variable with any local name. Although the system will allow you to store different types into the same local variable, this is poor programming practice and should be avoided.

Qualifying the name of a variable

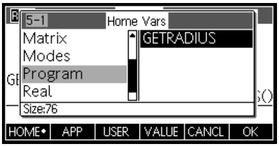
The HP39gII system has many system variables with names that are apparently the same. For example, the Function app has a variable named `Xmin`, but so, too, do the Polar, Parametric, Sequence, and Solve apps. In a program, or in the Home view, you can reference different versions of these by fully "qualifying" the name of the variable. This is done by inserting the name of the app (or program) that the variable belongs to, followed by two colons, and then the actual variable name. For example, the qualified variables `Function::Xmin` and `Parametric::Xmin` refer to the value of `Xmin` within each app, and could contain different values. Similarly, if you declare a local variable inside a program, you could reference that variable using the name of the program, followed by the two colons and the variable name.

Variables declared in a program should have descriptive names. For example, a variable used to store the radius of a circle can be named `RADIUS`. If such a variable is needed after the program executes, it can be exported from the program using the `EXPORT` command. To do this, the first command in the program (located before the program heading) would be `EXPORT RADIUS`. Then, if a value is assigned to `RADIUS`, the name would appear on the Vars menu and be visible globally. This feature allows for extensive and powerful interactivity among different environments in the HP39gII. Note that if more than one program exports a variable with the same name, the most recently exported version will be active, unless the name is fully qualified.

This program prompts the user for the value of `RADIUS`, and exports the variable for use outside the program.

```
EXPORT RADIUS;

EXPORT GETRADIUS()

BEGIN

INPUT(RADIUS);

END;
```

The EXPORT command for the variable RADIUS must appear before the heading of the function where RADIUS is assigned. After you execute this program, a new variable named RADIUS appears on the USER GETRADIUS section of the Vars menu.



Functions, their arguments, and parameters

The programming environment for the HP39gII is highly structured. You can define your own functions in a program, and data can be passed to a function using parameters. Functions can return a value (using the RETURN statement) or not. When a program is executed from Home, the program will return the value returned by the last statement that was executed.

Furthermore, functions can be defined in a program and exported for use by other programs in the same way that this is done for variables. This feature makes the HP39gII an incredibly powerful programming platform.

In this section, we will create a small set of programs, each illustrating some aspect of programming on the HP 39gII. Each of these programs will be used as a building block for a custom app described in the next section, *App Programs*.

Here is a program that defines a function called ROLLDIE that simulates the rolling of a single die, returning a random integer between 1 and whatever number is passed into the function:

First, create a new program named ROLLDIE. Then enter the program.

Program ROLLDIE

```
EXPORT ROLLDIE(N)

BEGIN

RETURN 1 + FLOOR(N*RANDOM);

END;
```

The first line is the heading of the function. Execution of the RETURN statement causes a random integer from 1 to N to be calculated and returned as the result of the function. Note that execution of a RETURN command causes execution of the function to terminate.

Any statements between the end of the RETURN statement and END are ignored.

On the Home screen (or in fact, anywhere in the calculator where a number can be used), you can enter ROLLDIE(6) and a random integer between 1 and 6, inclusive will be returned.

Another program could use the ROLLDIE function, and generate *n* rolls of a die with any number of sides. In the following program, the ROLLDIE function is used to generate *n* rolls of 2 dice, each with the number of sides given by the local variable sides. The results are stored into the list L2, so that L2(1) shows the number of times the dies came up with a 1, L2(2) shows the frequency of 2's, etc. L2(1) should be 0 as a result.

Program ROLLMANY

```
EXPORT ROLLMANY(n,sides)
BEGIN
LOCAL k,roll;
// initialize list of frequencies
MAKELIST(0,X,1,2*sides,1) ▶ L2;
FOR k FROM 1 TO n DO
ROLLDIE(sides) + ROLLDIE(sides) ▶ roll;
L2(roll)+1▶ L2(roll);
END;
END;
```

This program uses a FOR loop, explained in the section on loops.

A function's visibility can be restricted to within the program where it is defined by omitting the EXPORT command when the function is declared. For example, you could define the ROLLDIE function inside the ROLLMANY program like this:

```
EXPORT ROLLMANY(n,sides)
BEGIN
  LOCAL k,roll;
  // initialize list of frequencies
```

```
  MAKELIST(0,X,1,2*sides,1)▶ L2;
  FOR k FROM 1 TO n DO
ROLLDIE(sides)+ROLLDIE(sides)▶ roll;
          L2(roll)+1▶ L2(roll);
  END;
END;
ROLLDIE(n)
BEGIN
    RETURN 1 + FLOOR(n*RANDOM);
END;
```

In this scenario, assume there is no ROLLDIE function exported from another program. Instead, ROLLDIE is visible only in the context of ROLLMANY.

Finally, the list of results could be returned as the result of calling ROLLMANY instead of being stored directly into the global list variable, L2. This way, if the user wanted to store the results elsewhere, it could be done easily.

```
EXPORT ROLLMANY(n,sides)
BEGIN
  LOCAL k,roll,results;
  MAKELIST(0,X,1,2*sides,1)▶ results;
  FOR k FROM 1 TO n DO
ROLLDIE(sides)+ROLLDIE(sides)▶ roll;
          results(roll)+1▶ results(roll);
  END;
RETURN results;
END;
```

On the Home screen, you would enter ROLLMANY(100,6)▶ L5 and the results of the simulation of 100 rolls of two six-sided dice would be stored into list L5.

App programs

Apps are a unified collection of views, programs, notes, and associated data. Creating an app program allows you to redefine the app's views and how a user will interact with those views. This is done through two mechanisms: dedicated program functions with special names and redefining the views in the Views menu.

Using dedicated program functions

There is a set of special program names which run the named programs if they exist. These programs are run on the keyboard events shown in the table below. These program functions are designed to be used in the context of an app.

Program	**Name**	**Keystrokes**
Symb	Symbolic view	Symb Setup
SymbSetup	Symbolic Setup	SHIFT Symb Setup
Plot	Plot view	Plot Setup
PlotSetup	Plot Setup	SHIFT Plot Setup
Num	Numeric view	Num Setup
NumSetup	Numeric Setup	SHIFT Num Setup
Info	Info view	SHIFT Apps Info
START	Starts an app	START
RESET	Resets or initializes an app	RESET

Redefining the Views menu

The Views menu allows any app to define views in addition to the standard seven views shown in the table above. By default, each HP app has its own set of additional views contained in this menu. The `VIEWS` command allows you to redefine these views to run programs you have created for an app. The syntax for the `VIEWS` command is:

`VIEWS` "*text*"

By adding `VIEWS` "*text*", before the declaration of a function, you will override the list of views for the app. For example, if your app program defines 3 views "*SetSides*", "*RollDice*" and "*PlotResults*", when the user presses the Views key, he will see *SetSides*, *RollDice*, and *PlotResults* instead of the app's default view list.

Customizing an app

When an app is active, its associated program appears as the first item in the Program Catalog. It is within this program that you put functions to create a custom app. A useful procedure for customizing an app is illustrated below:

1. Decide on the HP app that you want to customize. For example, you could customize the Function app or the Statistics 1Var app. The customized app inherits all the properties of the HP app. Go to the apps Catalog and save the customized app with a unique name.
2. Customize the new app if you need to by configuring the settings, for example by setting axes or angle measures.
3. Develop the functions to work with your customized app. When you develop the app's functions, use the app naming conventions described above.
4. Put the `VIEWS` command into your program to modify the app's Views menu.
5. Decide if your app will create new global variables. If such variables are appropriate, you should `EXPORT` them from a separate user program that is called from the `Start()` function in the app program, so they will not have their values lost.
6. Test the customized app and debug the associated programs.

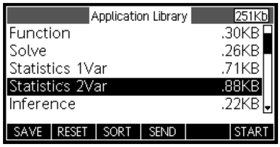
It is possible to link more than one app via programs. For example, a program associated with the Function app could execute a command to start the Statistics 1Var app, and a program associated with the Statistics 1Var app could return to the Function app (or launch any other app).

Example:

The following example illustrates the process of creating a custom app. This app creates an environment to simulate the rolling of a pair of dice, each with a number of sides specified by the user. The results are tabulated, and can be viewed either in a table or graphically. The app is based on the Statistics 1Var app.

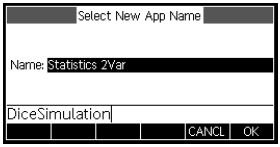
1. Save the Statistics 1Var app with a unique name.

Apps Info select Statistics 1Var
SAVE



2. Name the app `DiceSimulation` and press the OK menu key.

ALPHA ALPHA `D`
SHIFT ALPHA `ice`
ALPHA ALPHA ALPHA `S`
SHIFT ALPHA `imulation` OK OK



3. Start the new app.

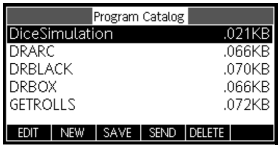
START

4. Open the Program Catalog.

SHIFT *PRGM*

Each app has one program attached to it. Initially, this program is empty. You customize the app by entering functions into that program.

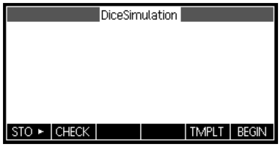
5. Edit the program `DiceSimulation`.



Select `DiceSimulation`

EDIT

It is here that you enter functions to customize the app. At this point, you decide how you expect the user to interact with the app. In this case, we will create views to do the following:



- `START`: start the app
- `SETSIDES`: specify the number of sides (faces) on each die
- `SETNUMROLLS`: specify number of times to roll the dice
- `RESET`: start over

The `START` option will initialize the app and display a note embedded in the app containing instructions for the user. The user will also interact with the app through the Numeric view and the Plot view. These views will be activated by pressing [Num Setup] and [Plot Setup], but the functions `Num` and `Plot` in our app program will actually launch those views after doing some configuration.

Recall the program to get the number of sides for a die, presented earlier in this chapter. It is expanded here, so that the possible sums of two such dice are stored in the list D1. Enter the following sub-routines into the app program for the `DiceSimulation` app.

The program DiceSimulation

```
START()
BEGIN
DICESIMVARS();
// Empty data columns D1 and D2
{}▶D1;
{}▶D2;
SETSAMPLE(H1,D1);
SETFREQ(H1,D2);
0▶H1Type;
END;
VIEWS “Roll Dice”,ROLLMANY()
BEGIN
LOCAL k,roll;
MAKELIST(X+1,X,1,2*SIDES-1,1)▶D1;
MAKELIST(0,X,1,2*SIDES-1,1)▶D2;
FOR k FROM 1 TO ROLLS DO
Roll:=ROLLDIE(SIDES)+ROLLDIES(SIDES);
D2(roll-1)+1▶D2(roll-1);
END;
-1▶Xmin;
MAX(D1)+1▶Xmax;
0▶Ymin;
MAX(D2)+1▶Ymax;
STARTVIEW(1,1);
END;
VIEWS "Set Sides", SETSIDES()
BEGIN
REPEAT
INPUT(SIDES,"Die Sides","N = ","Enter num
sides",2);
FLOOR(SIDES)▶SIDES;
```

```
IF SIDES<2 THEN
MSGBOX("Must be >= 2");
END;
UNTIL SIDES>=2;
END;
// specify num times to roll the dice.
VIEWS "Set Rolls",SETROLLS()
BEGIN
REPEAT
INPUT(ROLLS,"Num of Rolls","N = ","Enter
num rolls",10);
FLOOR(ROLLS)▶ROLLS;
IF ROLLS<1 THEN
MSGBOX("You must enter a number >= 1");
END;
UNTIL ROLLS>=1;
END;
Plot()
BEGIN
-1▶Xmin;
MAX(D1)+1▶Xmax;
0▶Ymin;
MAX(D2)+1▶Ymax;
STARTVIEW(1,1);
END;
```

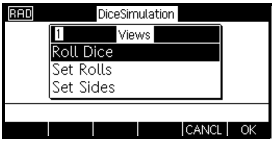
The ROLLMANY() routine is another adaptation from a program presented previously in this chapter. Since you cannot pass parameters into a program called through a selection from a custom Views menu, the exported variables SIDES and ROLLS are used in place of the parameters that were used in the previous versions.

The program above calls two other user programs: `ROLLDIE()` and `DICESIMVARS()`. `ROLLDIE()` appears earlier in this chapter. Here's `DICESIMVARS`. Store it into a new user program.

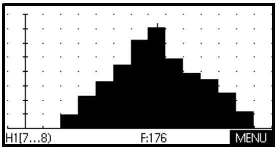
The program DICESIMVARS

```
EXPORT ROLLS,SIDES;
EXPORT DICESIMVARS()
BEGIN
10 ▶ ROLLS;
6 ▶ SIDES;
END;
```

Press [Views/Help] to see the custom app menu. Here you can set the number of sides of the dice, the number of rolls, and execute a simulation.



After running a simulation, press [Plot/Setup] to see a histogram of your simulation results.



Program commands

This section contains details on each of the individual commands grouped by category.

App commands

These commands allow you to launch any HP app, bring up any view of the current app, and change the options in the Views menu.

STARTAPP

Syntax: `STARTAPP`(*"name"*)

Starts the app with *name*. This will cause the app's program's `START` function to be run if present. The app's default view will be started. Note that the `START` function is always executed when the user presses START in the app library. Also works for user apps.

Example: `STARTAPP`(*"Function"*) launches the Function app.

STARTVIEW

Syntax: `STARTVIEW`(*n* [,*draw?*])

Starts the nth view of the current app. If *draw?* is true (non 0), it will force an immediate redrawing of the screen for that view.

The view numbers are as follows:

```
Symbolic:0
Plot:1
Numeric:2
Symbolic Setup:3
Plot Setup:4
Numeric Setup:5
App Info: 6
Views Menu:7
First special view(Split Screen Plot Detail):8
Second special view(Split Screen Plot Table):9
Third special view (Autoscale):10
Fourth special view (Decimal):11
Fifth special view (Integer):12
Sixth special view (Trig):13
```

The special views in parentheses refer to the Function app, and may differ for other apps. The numbers of special views for other apps correspond to their position in the Views menu for that app. The first special view is launched by `STARTVIEW(6)`, the second with `STARTVIEW(7)`, and so on.

Note that if $n < 0$, this allows starting global views:

```
HomeScreen:-1
Home Modes:-2
Memory Manager:-3
Apps Library:-4
Matrix Catalog:-5
List Catalog:-6
Program Catalog:-7
Notes Catalog:-8
```

VIEWS

Syntax: `VIEWS` ("*string*")

Adds a view to the Views menu.

These commands are used to select or deselect particular functions for graphing or display in the numeric view

DEBUG

Syntax: `DEBUG` (*"program name"*)

Starts the debugger for the program name you choose.

Block commands

The block commands determine the beginning and end of a sub-routine or function. There is also a `Return` command to recall results from sub-routines or functions.

BEGIN...END

Syntax: `BEGIN` *stmt1;stm2;...stmtN;* `END;`

A command list is a list of single statements enclosed within a `BEGIN - END` pair. Command lists appear as the body of a function.

RETURN

Syntax: `RETURN` *expression;*

Returns the current value of *expression*.

Assignment Statements

:=

Syntax: `var :=` *expression;*

▶

Syntax: *expression* ▶ `var;`

In each case, the expression is evaluated first, then the result stored into the variable var. ▶ and `:=` cannot be used with the graphics variables G0..G9. Instead, see the command `BLIT`.

When assigning a value to a cell in a list, vector, or matrix, use the ▶ command rather than `:=`. For example, the command `73` ▶ *L1*(*5*) will put the number 73 into the 5th position of list L1. If you are entering a program using a calculator emulator running on a computer, then => can be used as a synonym for ▶.

Branch Commands

IF...THEN...END

Syntax: `IF` *test* `THEN` *command(s)* `END`;

Evaluate *test*. If *test* is true (non 0), execute *command(s)*. Otherwise, nothing happens.

Example:

IF...THEN...ELSE...END

Syntax: `IF` *test* `THEN` *command(s)*1 `ELSE` *command(s)2* `END`;

Evaluate *test*. If *test* is true (non 0), execute *command(s)1*, otherwise, execute *command(s)2*

IFTE

Syntax: `IFTE`(*test,true_xpr,false_xpr*)

Evaluates *test*. If *test* is true (non 0), return *true_xpr*, otherwise return *false_xpr*

IFERR...THEN...END

`IFERR` *commands1* `THEN` *commands2* [`ELSE` *commands3*] `END`;

Executes sequence of *commands1*. If an error occurs during execution of *commands1*, execute sequence of *commands2*. Otherwise, execute sequence of *commands3*.

CASE...END

Syntax:

```
CASE
    IF test1 THEN commands1 END
    IF test2 THEN commands2 END
    …
    [DEFAULT commands]
END;
```

Evaluates *test1*. If true, execute *commands1* and end the `CASE`. Otherwise, *evaluate test2*. If true, execute *commands2*. Continue evaluating tests until a true is found. If no true test is found, execute *commandsD*, if provided.

Example:

```
CASE
IF x < 0 THEN RETURN "negative"; END
IF x < 1 THEN RETURN "small"; END
DEFAULT RETURN "large";
END;
```

Drawing Commands

There are 10 graphic variables in the HP39gII, called *G0* to *G9*. *G0* is always the current screen graphic.

G1 to *G9* can be used to store temporary graphic objects (called *GROBs* for short) when programming applications that use graphics. Variables *G1* to *G9* are temporary and are cleared when the calculator turns OFF.

They are twenty-six functions that can be used to modify graphic variables. Thirteen of them work based on Cartesian coordinates using the Cartesian plane defined in the current app by the variables *Xmin, Xmax, Ymin,* and *Ymax* in the plot setup menu.

Thirteen of them work on pixel coordinates where the pixel `0,0` is the top left pixel of the *GROB*, and `255,126` is the bottom right. This second set of function has a *_P* suffix on the function name.

PIXON and PIXON_P

Syntax: `PIXON`([*G*], *xposition, yposition* [*,color*])

`PIXON_P`([*G*], *xposition, yposition* [*,color*])

Sets the color of the pixel of *G* with coordinates *x,y* to color. *G* can be any of the graphic variables and is optional. The default is *G0*, the current graphic. Color can be `0` to `3` (`0`=black, `1`= dark gray, `2`= light gray, `3`= white) and is optional. The default is `0`.

PIXOFF and PIXOFF_P

Syntax: `PIXOFF`([*G*], *xposition, yposition*)

`PIXOFF_P`([*G*], *xposition, yposition*)

Sets the color of the pixel of *G* with coordinates *x,y* to white. *G* can be any of the graphic variables and is optional. The default is *G0*, the current graphic

GETPIX and GETPIX_P

Syntax: `GETPIX`([*G*], *xposition, yposition*)

`GETPIX_P`([*G*], *xposition, yposition*)

Returns the color of the pixel of *G* with coordinates *x,y*.

G can be any of the graphic variables and is optional. The default is *G0*, the current graphic.

RECT and RECT_P

Syntax: `RECT`([*G, x1, y1, x2, y2, edgecolor, fillcolor*])

`RECT_P`([*G, x1, y1, x2, y2, edgecolor, fillcolor*])

Draws a rectangle on G between points *x1,y1* and *x2,y2* using edgecolor for the perimeter and fillcolor for the inside.

G can be any of the graphic variables and is optional. The default is *G0*, the current graphic

x1, y1 are optional. The default values represent the top left of the graphic.

x2, y2 are optional. The default values represent the bottom right of the graphic.

edgecolor and *fillcolor* can be `-1` to `3` (`-1`= transparent, `0`=black, `1`= dark gray, `2`= light gray, `3`= white).

edgecolor is optional. The default is white.

fillcolor is optional. The default is edgecolor.

Note: To erase a `GROB`, execute `RECT(`*G*`)`. To clear the screen execute `RECT()`.

When optional arguments are provided in a command like `RECT`, with multiple optional parameters, provided arguments correspond to the leftmost parameters first. For example, in the program below, the arguments `40` and `90` in the `RECT_P` command correspond to *x1* and *y1*. The argument `0` corresponds to *edgecolor*, since there is only the one additional argument. If there had been two additional arguments, they would have referred to *x2* and *y2* rather than *edgecolor* and *fillcolor*. The program produces the figure below to the right.

```
EXPORT BOX()
BEGIN
RECT();
RECT_P(40,90,0);
FREEZE;
END
```

The program below also uses the RECT_P command. In this case, the pair of arguments *0* and *3* correspond to *x2* and *y2*. The program produces the figure below to the right.

```
EXPORT BOX()
BEGIN
RECT();INVERT(G0);
RECT_P(40,90,0,3);
FREEZE;
END
```

INVERT and INVERT_P

Syntax: INVERT([*G, x1, y1, x2, y2*])

INVERT_P([*G, x1, y1, x2, y2*])

Inverts a rectangle on *G* between points *x1,y1* and *x2,y2*. This means that every black pixel becomes white and vice-versa. In the same way Light gray and dark gray are inverted. *G* can be any of the graphic variables and is optional. The default is *G0*.

x2, y2 are optional and if not specified will be the bottom right of the graphic.

x1, y1 are optional and if not specified will be the top left of the graphic. If only one x,y pair is specified, it refers to the top left.

ARC and ARC_P

Syntax; ARC(*G, x, y, r* [*,c, a1, a2*])

ARC_P(*G, x, y, r* [*,c, a1, a2*])

Draws an arc or circle on *G*, centered on point *x,y*, with radius *r* and color *c* starting at angle *a1* and ending on angle *a2*.

G can be any of the graphic variables and is optional. The default is *G0*

r is given in pixels.

c is optional and if not specified black is used.

a1 and *a2* follow the current angle mode and are optional. The default is a full circle.

LINE and LINE_P

Syntax: `LINE`(*G, x1, y1, x2, y2, c*)

`LINE_P`(*G, x1, y1, x2, y2, c*)

Draws a line of color *c* on *G* between points *x1,y1* and *x2,y2*.

G can be any of the graphic variables and is optional. The default is *G0*.

c can be `0` to `3` (`0`=black, `1`= dark gray, `2`= light gray, `3`= white). *c* is optional. The default is black.

TEXTOUT and TEXTOUT_P

Syntax: `TEXTOUT`(*text* [,*G*], *x, y* [,*font, c1, width, c2*])

`TEXTOUT_P`(*text* [,*G*], *x, y* [,*font, c1, width, c2*])

Draws text using color *c1* on graphic *G* at position *x, y* using font. Do not draw text more than width pixels wide and erase the background before drawing the text using color *c2*. *G* can be any of the graphic variables and is optional. The default is *G0*

Font can be:

0: current font selected in mode screen, *1*: small font *2*: large font. Font is optional and if not specified is the current font selected in mode screen.

c1 can be `0` to `3` (`0`=black, `1`= dark gray, `2`= light gray, `3`= white). *c1* is optional. The default is black.

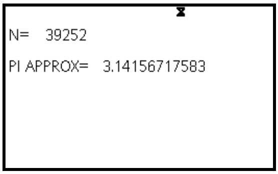
width is optional and if not specified, no clipping is performed.

c2 can be `0` to `3` (`0`=black, `1`= dark gray, `2`= light gray, `3`= white). *c2* is optional. If not specified the background is not erased.

Example:

This program displays the successive approximations for using the series for the arctangent(1).

```
EXPORT RUNPISERIES()

BEGIN

LOCAL sign;

2 ▶ K;4 ▶A;

-1 ▶ sign;

RECT();

TEXTOUT_P("N=",0,0);

TEXTOUT_P("PI APPROX=",0,30);

REPEAT

A+sign*4/(2*K-1) ▶ A;

TEXTOUT_P(K
,35,0,2,0,100,3);

TEXTOUT_P(A
,90,30,2,0,100,3);

sign*-1 ▶ sign;

K+1▶ K;

UNTIL 0;

END;
```



The program executes until the user presses ON/C OFF to terminate. The spaces after `K` (the number of the term) and `A` (the current approximation) in the `TEXTOUT_P` commands are there to overwrite the previously displayed value.

BLIT and BLIT_P

Syntax: `BLIT`([*trgtGRB, dx1, dy1, dx2, dy2*], *srcGRB* [*,sx1, sy1, sx2, sy2, c*])

`BLIT_P` ([*trgtGRB, dx1, dy1, dx2, dy2*], *srcGRB* [*,sx1, sy1, sx2, sy2, c*])

Copies the region of *srcGRB* between point *sx1, sy1* and *sx2, sy2* into the region of *trgtGRB* between points *dx1, dy1* and *dx2, dy2*. Do not copy pixels from *srcGRB* that are color *c*.

trgtGRB can be any of the graphic variables. *trgtGRB* can be any of the graphic variables and is optional. The default is G0.

srcGRB can be any of the graphic variables.

dx2, dy2 are optional and if not specified will be calculated so that the destination area is the same size as the source area.

sx2, sy2 are optional and if not specified will be the bottom right of the *srcGRB*.

sx1, sy1 are optional and if not specified will be the top left of *srcGRB*.

dx1, dy1 are optional and if not specified will be the top left of *trgtGRB*.

c can be `0` to `3` (`0`=black, `1`= dark gray, `2`= light gray, `3`= white). *c* is optional. If not specified all pixels from *G2* will be copied.

NOTE Using the same variable for *trgtGRB* and *srcGRB* can be unpredictable when the source and destination overlap.

DIMGROB and DIMBROB_P

Syntax: `DIMGROB`(*G, w, h* [,*c*]) or `DIMGROB`(*G* [,*line_1, line_2,…,line_h*])

`DIMGROB`(*G, w, h* [,*c*]) or `DIMGROB`(*G* [,*line_1, line_2,…,line_h*])

Sets the dimensions of *GROB G* to w*h. initializes the graphic *G* with color *c* or with the graphic data provided in the list. *G* can be any graphic variable except G0. *c* can be `0` to `3` (`0`=black, `1`= dark gray, `2`= light gray, `3`= white). *c* is optional. The default is white.

If the graphic is initialized using graphic data, the list must have as many numbers as the height of the *GROB*. Each number, as seen in base 16 describes a line. Two bits are used for each pixel (`00`=black, `01`=dark gray, `10`=light gray, `11`=white). Hence, each hex digit describes two pixels.

You can enter hexadecimal number using the `0xdigits` syntax.

The first pixel of the line is defined by the 2 lest significant bit of the number. The 2nd pixel by the 2 lest significant bit, etc.

SUBGROB and SUBGROB_P

Syntax: `SUBGROB`(*srcGRB* [,*x1, y1, x2, y2*], *trgtGRB*)

`SUBGROB_P`(*srcGRB* [,*x1, y1, x2, y2*], *trgtGRB*)

Sets *trgtGRB* to be a copy of the area of *srcGRB* between points *x1,y1* and *x2,y2*.

srcGRB can be any of the graphic variables and is optional. The default is *G0*.

trgtGRB can be any of the graphic variables except *G0*.

x2, y2 are optional and if not specified will be the bottom right of *srcGRB*.

x1, y1 are optional and if not specified will be the top left of *srcGRB*.

NOTE `SUBGROB`(*G1, G4*) will copy G1 in G4.

GROBH and GROBH_P

Syntax: `GROBH`(*G*)

`GROBH_P`(*G*)

Returns the height of *G*.

G can be any of the graphic variables and is optional. The default is *G0*.

GROBW and GROBW_P

Syntax: `GROBW`(*G*)

`GROBW_P`(*G*)

Returns the width of *G*.

G can be any of the graphic variables and is optional. The default is *G0*.

FREEZE

Syntax: `FREEZE`

Pauses program execution until a key is pressed. This prevents the screen from being redrawn after the end of the program execution, leaving the modified display on the screen for the user to see.

I/O Commands

This section describes commands for inputting data into a program, and for outputting data from a program. These commands allow users to interact with programs.

These commands start the Matrix and List editors.

EDITLIST

Syntax: `EDITLIST`(*listvar*)

Starts the list editor, loading *listvar*.

Example: `EDITLIST`(*L1*) edits list L1.

EDITMAT

Syntax: `EDITMAT`(*matrixvar*)

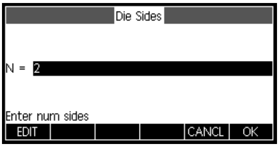
Starts the Matrix Editor and displays the specified matrix. If used in programming, returns to the program when user presses **OK**.

Example: `EDITMAT`(*M1*) edits matrix M1.

INPUT

Syntax: `INPUT`(*var* [,*"title", "label", "help", default*]);

Starts a dialog box with the title text, *title*, with one field named *label*, displaying *help* at the bottom and using the *default value*. Updates the variable *var* if the user presses **OK** and returns 1. If the user presses **CANCL**, it does not update the variable, and returns 0.



Example:

```
EXPORT SIDES;

EXPORT GETSIDES()

BEGIN

INPUT(SIDES,"Die Sides","N = ","Enter num sides",2);

END;
```

PRINT

Syntax: `PRINT`(*expression or string*);

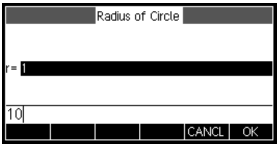
Prints the result of expression or string to the terminal.

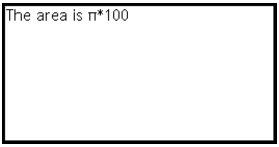
The terminal is a program text output viewing mechanism which is displayed only when `PRINT` commands are executed. When visible, you can use ▼ and ▲ to view the text, Clear to erase the text and any other key to hide the terminal. You can show the terminal at anytime using the [ON/C OFF] [4 Matrix T] combination (press and hold [ON/C OFF], then press [4 Matrix T], then release both keys). Pressing [ON/C OFF] stops the interaction with the terminal.

There are also commands for outputting data in the Graphics section. In particular, the commands `TEXTOUT` and `TEXTOUT_P` can be used for test output.

This example prompts the user to enter a value for the radius of a circle, and prints the area of the circle on the terminal.

```
EXPORT AREACALC()

BEGIN

LOCAL radius;

INPUT(radius, "Radius of Circle",:"r =
","Enter radius",1);

PRINT("The area is "π+ *radius^2);

END;
```





Notice the use of the `LOCAL` variable for the radius, and the naming convention that uses lower case letters for the local variable. Adhering to such a convention will improve the readability of your programs.

GETKEY

Syntax: `GETKEY`

Returns the ID of the first key in the keyboard buffer, or -1 if no key was pressed since the last call to `GETKEY`. Key IDs are integers from 0 to 50, numbered from top left (key 0) to bottom right (key 50) as shown on the following page.

ISKEYDOWN

Syntax: `ISKEYDOWN`(*key_id*);

Returns true (non-zero) if the key whose *key_id* is provided is currently pressed, and false (0) if it is not.

MSGBOX

Syntax: `MSGBOX`(*expression or string* [,*ok_cancel?*]);

Displays a message box with the value of the given expression or string.

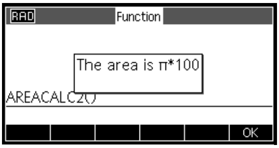
If *ok_cancel?* is true, displays OK and CANCL menu keys, otherwise only displays the OK key. Default value for *ok_cancel* is false.

Returns true (non-zero) if the user presses OK, false (0) if the user presses CANCL.

Replace the `PRINT` command in the previous example with the `MSGBOX` command to:

```
EXPORT AREACALC()

BEGIN

LOCAL radius;

INPUT(radius, "Radius of Circle",:"r =
","Enter radius",1);

MSGBOX("The area is "π+ *radius^2);

END;
```

If the user enters 10 for the radius, the message box shows this:



CHOOSE

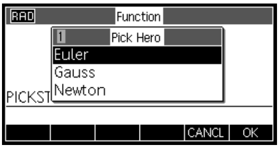
Syntax:`CHOOSE`(*var, "title", "item1", "item2",…,"itemn"*)

Displays a choose box with the given title and containing the choose items. If the user selects an object, the variable whose name is provided will be updated to contain the number of the selected object (an integer, 1, 2, 3, …) or 0 if the user presses CANCL.

Returns true (non zero) if the user selects an object, otherwise return false (0).

Example:

```
CHOOSE(N,"Pick
Hero","Euler","Gauss","Newton");

IF N==1 PRINT("You picked Euler");END

ELSE IF N==2
PRINT("You picked
Gauss");END

ELSE IF N==3
PRINT("You picked
Newton");END

END;
```



After execution of `CHOOSE`, the value of *n* will be updated to contain 0, 1, 2, or 3. The `IF THEN ELSE` command causes the name of the selected person will be printed to the terminal.

Loop commands

FOR…FROM…TO… DO…END

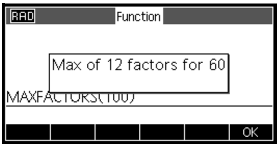
Syntax: `FOR` *var* `FROM` *start* `TO` *finish* [*STEP increment*] `DO` *commands* `END;`

Sets variable *var* to start, and for as long as this variable's value is less than or equal to *finish*, executes the sequence of *commands*, and then adds 1 (*increment*) to *var*.

Example 1: This program determines which integer from 2 to N has the greatest number of factors.

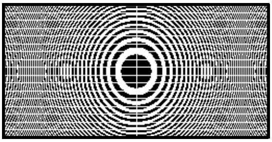
```
EXPORT MAXFACTORS(N)
BEGIN
LOCAL cur, max,k,result;
1▶ max;1▶ result;
FOR k FROM 2 TO N DO
  SIZE(idivis(k)) ▶cur;
  IF cur > max THEN
    cur ▶max;
    k ▶result;
  END;
END;
MSGBOX("Max of "+ max +" factors  for "+result);
```

In Home, enter `MAXFACTORS`(100).



Example 2: This program draws an interesting pattern on the screen.

```
EXPORT DRAWPATTERN()
BEGIN
  LOCAL xinc,yinc,color;
  STARTAPP("Function");
  RECT();
 xincr := (Xmax - Xmin)/254;
 yincr := (Ymax - Ymin)/110;
 FOR X FROM Xmin TO Xmax STEP xincr DO
  FOR Y FROM Ymin TO Ymax STEP yincr DO
    color := FLOOR(X^2+Y^2) MOD 4;
   PIXON(X,Y,color);
    END;
  END;
FREEZE;
END;
```



REPEAT…UNTIL…

Syntax: `REPEAT` *commands* `UNTIL` *test*;

Repeats the sequence of *commands* until *test* is true (non 0).

This code prompts for a positive value for SIDES, modifying an earlier program in this chapter.

Example:

```
EXPORT SIDES;
EXPORT GETSIDES()
BEGIN
  REPEAT
    INPUT(SIDES,"Die Sides","N = ","Enter num sides",2);
 UNTIL SIDES>0;
END;
```

WHILE…DO…END

Syntax: `WHILE` *test* `DO` *commands* `END`;

Evaluate test. If result is true (non 0), executes the *commands*, and repeat.

Example: A perfect number is one that is equal to the sum of all its proper divisors. For example, 6 is a perfect number because 6 = 1+2+3. This function returns true when its argument is a perfect number.

Example:

```
EXPORT ISPERFECT(n)
BEGIN
  LOCAL d, sum;
  2 ▶ d;
  1 ▶ sum;
  WHILE sum < = n AND d < n DO
    IF irem(n,d)==0 THEN
      sum+d ▶ sum;
    END;
    d+1▶ d;
  END;
  RETURN sum==n;
END;
```

This program displays all the perfect numbers up to 1000:

```
EXPORT PERFECTNUMS()
BEGIN
LOCAL k;
FOR k FROM 2 TO 1000 DO
  IF ISPERFECT(k) THEN
    MSGBOX(k+" is perfect, press OK");
  END;
END;
END;
```

BREAK

Syntax: `BREAK`

Exits from a loop. Execution picks up with the first statement after the loop.

CONTINUE

Syntax: `CONTINUE`

Transfer execution to the start of the next iteration of a loop.

Matrix Commands

Some matrix commands take as argument the matrix variable name on which the command is applied. Valid names are the global variables M0..M9 or a local variable that contains a matrix.

ADDCOL

Syntax: `ADDCOL`

(*name* [,*value1*,...,*valuen*],*column_number*)

Add Column. Inserts values into a column before *column_number* in the specified matrix. You enter the values as a vector (these are not optional arguments!). The values must be separated by commas and the number of values must be the same as the number of rows in the matrix name.

ADDROW

Syntax: `ADDROW`

(*name* [,*value1*,...,*valuen*],*row_number*)

Add Row. Inserts values into a row before *row_number* in the specified matrix. You enter the values as a vector (these are not optional arguments!). The values must be separated by commas and the number of values must be the same as the number of columns in the matrix name.

DELCOL

Syntax: `DELCOL`(*name* ,*column_number*)

Delete Column. Deletes *column column_number* from matrix name.

DELROW

Syntax: `DELROW`(*name* ,*row_number*)

Delete Row. Deletes row *row_number* from matrix name.

EDITMAT

Syntax: `EDITMAT`(*name*)

Starts the Matrix Editor and displays the specified matrix. If used in programming, returns to the program when user presses OK. Even though this command returns the matrix that was edited, `EDITMAT` cannot be used as an argument to other matrix commands.

RANDMAT

Syntax: `RANDMAT` (*name, rows, columns*)

Creates random matrix with a specified number of rows and columns and stores the result in *name* (*name* must be M0...M9). The entries will be integers ranging from -99 to 99.

REDIM

Syntax: `REDIM`(*name, size*)

Redimensions the specified matrix (*name*) or vector to *size*. For a matrix, size is a list of two integers (*n1,n2*). For a vector, size is a list containing one integer (*n*). Existing values in the matrix are preserved. Fill values will be 0.

REPLACE

Syntax: `REPLACE`(*name, start, object*)

Replaces portion of a matrix or vector stored in *name* with an *object* starting at position, *start*. *Start* for a matrix is a list containing two numbers; for a vector, it is a single number. `REPLACE` also works with lists and graphics.

SCALE

Syntax: `SCALE`(*name, value, rownumber*)

Multiplies the specified *row_number* of the specified matrix by *value*.

SCALEADD

Syntax: `SCALEADD`(*name, value, row1, row2*)

Multiplies the specified *row1* of the matrix (*name)* by *value*, then adds this result to the second specified *row2* of the matrix (*name)*.

SUB

Syntax: `SUB`(*name, object, start, end*)

Extracts a sub-object, a portion of a list, matrix, or graphic from *object,* and stores it into *name*. *Start* and *end* are each specified using a list with two numbers for a matrix, a number for vector or lists, or an ordered pair, (*X,Y*), for graphics.

SWAPCOL

Syntax: `SWAPCOL`(*name, column1, column2*)

Swaps columns. Exchanges *column1* and *column2* of the specified matrix (*name*).

SWAPROW

Syntax: `SWAPROW`(*name, row1, row2*)

Swap Rows. Exchanges *row1* and *row2* in the specified matrix (*name*).

String commands

A string is a sequence of characters enclosed in double quotes (""). To put a double quote in a string, use two consecutive double quotes. The \ character starts an "escape" sequence, and the character(s) immediately following are interpreted specially. *n* inserts a new line, two backslashes insert a single backslash.

Example: `PRINT`(*"Hello**nWorld!"*) displays

```
Hello
World!
```

on the terminal.

+

Syntax: *str1 + str2 or str1 + expression*

Adds two strings together.

Example 1: *"QUICK"+"DRAW"* returns "QUICKDRAW"

Example 2: *32* ▶ *X;"X = "+X* returns "X = 32"

asc

Syntax: `asc`(*str*)

Returns a vector containing the ASCII codes of string *str*.

Example: `asc`(*"AB"*) returns [65,66]

char

Syntax: `char`(*vector or int*)

Returns the string corresponding to the character codes in *vector*, or the single code *int*.

Examples: `char`(*65*) returns "A"; `char`([82,77,72]) returns "RMH"

dim

Syntax: `dim`(*str*)

Returns the number of characters in string *str*.

Example: `dim`(*"12345"*) is 5, `dim`(*""""*) and `dim`(*"**n"*) are both 1 (notice the use of the two double quotes and the escape sequence).

expr

Syntax: `expr`(*str*)

Parses the string *str* into a number or expression.

Examples: `expr`(*"2+3"*) returns 5. If the variable *X* has the value 90, then expr(*"X+10"*) returns 100. In a program, if variables *a* and *b* are not declared and *X* is 90, then `expr`(*"2X+a+b"*) returns 180+a+b.

When used in tandem with other functionality built-in to the HP39gII, the *expr* command can be used in powerful ways. For example, you could build functions up out of strings and export these functions so they can be used throughout the calculator.

string

Syntax: `string`(*object*);

Returns a string representation of the *object*. The result varies depending on the type of *object*. For example,

`string`(*2/3*); results in `string`(*"2/3"*)

Examples:

String	**Result**
`string`(2/3)	"2/3"
`string`(2.0/3)	"0.666666666667"
`string`(F1), when F1(X) = COS(X)	"(X)->COS(X)"
`string`(L1) when L1 = {1,2,3}	"1,2,3"
`string`(M1) when M1 = $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	"[1,2,3],[4,5,6]"

inString

Syntax: `inString`(*str1*,*str2*)

Returns the index of the first occurrence of *str2* in *str1*. Returns 0 if *str2* is not present in *str1*. Note that the first character in a string is a position *1*.

Examples:

`inString`(*"vanilla"*,*"van"*) returns 1.

`inString` (*"banana"*,*"na"*) returns 3

`inString`(*"ab"*,*"abc"*) returns 0

left

Syntax: `left`(*str,n*)

Return the first *n* characters of string *str*. If $n \geq dim(str)$ or $n < 0$, returns *str*. If *n* == 0 returns the empty string.

Example: `left`(*"MOMOGUMBO",3*) returns "MOM"

right

Syntax: right(*str,n*)

Returns the last n characters of string *str*. If *n* <= 0, returns empty string. If *n* > - *dim*(*str*), returns str

Example: `right`("MOMOGUMBO",5) returns "GUMBO"

mid

Syntax: `mid`(*str,pos,* [*n*])

Extracts *n* characters from string *str* starting at index pos. *n* is optional, if not specified, extracts all the remainder of the string.

Example: `mid`(*"MOMOGUMBO",3,5*) returns "MOGUM", `mid`(*"PUDGE",4*) returns "GE"

rotate

Syntax: `rotate`(*str,n*)

Permutation of characters in string *str*. If 0<=*n*<*dim*(*str*), shifts *n* places to left. If *-dim*(*str*)<*n*<=-1, shifts *n* spaces to right. If *n* > *dim*(*str*) or *n* < *-dim*(*str*), returns *str*.

Examples:

`rotate`(*"12345",2*) returns "34512"
`rotate`(*"12345",-1*) returns "51234"
`rotate`(*"12345",6*) returns "12345"

Test Commands

The Test commands include both Boolean and relational operations. Boolean and relational expressions evaluate to true or false. A non-zero number is equivalent to true, and a number equal to 0 is equivalent to false. Note that in addition to real numbers, complex numbers, strings, lists, and matrices can be compared using the relational operators ==, NOT (or !=), and (or <> or !=). These commands are not in the CMDS menu. They appear in the Math menu but are listed here for convenience.

Relational expressions

==

Equality.

Syntax: *object1* == *object2*

Example: *3+1== 4* returns 1.

<

Less than.

Syntax: *object1* < *object2*

Example: *3+1* < *4* returns 0.

≤ (or < =)

Less than or equal to.

Syntax: *object1* ≤ *object2*

Example: *3+1* ≤ *4* returns 1.

>

Greater than.

Syntax: *object1* > *object2*

Example: *3+1* > *4* returns 0.

≥ (or > =)

Greater than or equal to.

Syntax: *object1* ≥ *object2*

Example: *3+1* ≥ *4* returns 1.

≠ (or <> or !=)

Not equal to.

Syntax: *object1* ≠ *object2*

Example: *3+1* ≠ *4* returns 0.

Boolean expressions

AND (or &&)

Logical And.

Syntax: *expr1* `AND` *expr2*

Example: *3+1==4* `AND` *4* < *5* returns 1.

OR (or ||)

Logical Or.

Syntax: *expr1* `OR` *expr2*

Example: *3+1==4* `OR` *8* < *5* returns 1.

XOR

Exclusive Or.

Syntax: *expr1* `XOR` *expr2*

Example: *3+1==2* `OR` *8 < 5* returns 0.

NOT (or !)

Logical Negation.

Syntax: `NOT`(*expr1*)

Example: `NOT`(*3+1==4*) returns 0.

Variable commands

These commands allow you to control the visibility of a user-defined variable or function.

Export

Export.

Syntax: `EXPORT`(*FunctionName*)

Exports the function `FunctionName` so that it is globally available and appears in the Program Commands menu (SHIFT *CMDS*) when USER is pressed.

Local

Local.

Syntax: `LOCAL`*var1,var2,…varn;*

Makes the variables var1, var2, etc. local to the program in which they are found.

Variables and Programs

The HP 39gII has three types of variables: Home variables, App variables, and User variables. You use the Variable menu (Vars Chars A) to retrieve Home, app, or User variables.

Home variables are used for real numbers, complex numbers, graphics, lists, and matrices among other things. Home variables keep the same value in Home and in apps.

App variables are those whose values depend on the current app. The app variables are used in programming to represent the definitions and settings you make when working with apps interactively.

User variables are variables exported from a user program. They provide one of several mechanisms to allow programs to communicate with the rest of the calculator, or with other programs. Once a variable has been exported from a program, it will appear among the User variables in the Vars menu, next to the program that exported it.

This chapter deals with App variables and User variables. For information on Home variables, see *Variables and memory management*.

App variables

Not all app variables are used in every app. S1fit, for example, is only used in the Statistics 2Var app. However, most of the variables are used in common by the Function, Parametric, Polar, Sequence, Solve, Statistics 1Var, and Statistics 2Var apps. If a variable is not available in all of these apps, or is available only in some other apps, then a list of the apps where the variable can be used appears under the variable name.

The following sections list the app variables by the view in which they are used.

Plot view variables

Axes

Turns axes on or off. From Plot Setup, check (or uncheck) `AXES`.

Or, in a program, type:

`1 ▶ Axes`—to turn axes on (default).

`0 ▶ Axes`—to turn axes off.

Cursor

Sets crosshairs type. (Inverted or blinking is useful if the background is solid).

From Plot Setup, choose `Cursor`.

Or, in a program, type:

`2 ▶ CrossType`—for blinking crosshairs.

`1 ▶ CrossType`—to invert the crosshairs.

`0 ▶ CrossType`—for solid crosshairs (default).

GridDots

Turns the background dot grid in Plot view on or off.

From Plot setup, check (or uncheck) `GRID DOTS`.

Or, in a program, type:

`1` ▶ `GridDots`—to turn the grid dots on (default).

`0` ▶ `GridDots`—to turn the grid dots off.

GridLines

Turns the background line grid in Plot view on or off.

From Plot setup, check (or uncheck) `GRID LINES`.

Or, in a program, type:

`1` ▶ `GridLines`—to turn the grid lines on (default).

`0` ▶ `GridLines`—to turn the grid lines off.

Hmin/Hmax
Statistics 1Var

Defines minimum and maximum values for histogram bars.

From Plot Setup for one-variable statistics, set values for `HRNG`.

Or, in a program, type:

n_1 ▶ `Hmin`

n_2 ▶ `Hmax`

where $n_1 < n_2$

Hwidth
Statistics 1Var

Sets the width of histogram bars.

From Plot Setup for one-variable statistics, set a value for `Hwidth`.

Or, in a program, type:

n ▶ `Hwidth`

Indep

Defines the value of the independent variable used in tracing mode.

Or, in a program, type:

n ▶ `Indep`

Labels

Draws labels in Plot view showing X and Y ranges.

From Plot Setup, check (or uncheck) `Labels`

Or, in a program, type:

1 ▶ `Labels`—to turn labels on.

0 ▶ `Labels`—to turn labels off (default).

Nmin/Nmax
Sequence

Defines the minimum and maximum independent variable values.

Appears as the `NRNG` fields in the Plot Setup input form. From Plot Setup, enter values for `NRNG`.

Or, in a program, type:

n_1 ▶ `Nmin`

n_2 ▶ `Nmax`

where $n_1 < n_2$

Recenter

Recenters at the cursor location when zooming.

From Plot-Zoom-Set Factors, check (or uncheck) `Recenter`.

Or, in a program, type:

1 ▶ `Recenter`— to turn recenter on (default).

0 ▶ `Recenter`— to turn recenter off.

S1mark-S5mark
Statistics 2Var

Sets the mark to use for scatter plots.

From Plot Setup for two-variable statistics, highlight one of `S1mark-S5mark` and choose a mark.

Or, in a program, type:

n ▶ `S1mark`

where n is 1,2,3,...5

SeqPlot
Sequence

Enables you to choose types of sequence plots: Stairstep or Cobweb.

From Plot Setup, select `SeqPlot`, then choose `Stairstep` or `Cobweb`.

Or, in a program, type:

1 ▶ `SeqPlot`—for Stairstep.

2 ▶ `SeqPlot`—for Cobweb.

θmin/θmax
Polar

Sets the minimum and maximum independent values.

Appears as the RNG field in the Plot Setup input form. From Plot Setup, enter values for RNG.

Or, in a program, type:

n_1 ▶ θmin

n_2 ▶ θmax

where $n_1 < n_2$

θstep
Polar

Sets the step size for the independent variable.

From Plot Setup, enter a value for STEP.

Or, in a program, type:

n ▶ θstep

where $n > 0$

Tmin/Tmax
Parametric

Sets the minimum and maximum independent variable values.

Appears as the TRNG field in the Plot Setup input form. From Plot Setup, enter values for TRNG.

Or, in a program, type:

n_1 ▶ Tmin

n_2 ▶ Tmax

where $n_1 < n_2$

Tracing

Turns tracing on or off in the Plot view.

From the Plot view, press MENU then TRACE to toggle tracing off or on.

Or, in a program, type:

1 ▶ Tracing—to turn Tracing mode on (default).

0 ▶ Tracing—to turn Tracing mode off.

Tstep
Parametric

Sets the step size for the independent variable.

From Plot Setup, enter a value for TSTEP.

Or, in a program, type

n ▶ `Tstep`

where $n > 0$

Xcross Sets the horizontal coordinate of the crosshairs.

Only works with TRACE off (TRACE). From the Plot view, use the cursor keys to move to the desired *x*-value.

Or, in a program, type:

n ▶ `Xcross`

Ycross Sets the vertical coordinate of the crosshairs.

Only works with TRACE off (TRACE). From the Plot view, use the cursor keys to move to the desired *y*-value.

Or, in a program, type:

n ▶ `Ycross`

Xtick Sets the distance between tick marks for the horizontal axis.

From Plot Setup input, enter a value for `Xtick`.

Or, in a program, type:

n ▶ `Xtick` where $n > 0$

Ytick Sets the distance between tick marks for the vertical axis.

From Plot Setup, enter a value for `Ytick`.

Or, in a program, type:

n ▶ `Ytick` where $n > 0$

Xmin/Xmax Sets the minimum and maximum horizontal values of the plot screen.

Appears as the `XRNG` fields (horizontal range) in the Plot Setup input form. From Plot Setup, enter values for `XRNG`.

Or, in a program, type:

n_1 ▶ `Xmin`

n_2 ▶ `Xmax`

where $n_1 < n_2$

Ymin/Ymax

Sets the minimum and maximum vertical values of the plot screen.

Appears as the `YRNG` fields (vertical range) in the Plot Setup input form. From Plot Setup, enter the values for `YRNG`.

Or, in a program, type:

n_1 ▶ `Ymin`

n_2 ▶ `Ymax`

where $n_1 < n_2$

Xzoom

Sets the horizontal zoom factor.

From Plot setup (Plot Setup), press MENU then ZOOM. Scroll to `Set Factors`, select it and press OK. Enter the value for `XZOOM` OK.

Or, in a program, type:

n ▶ `XZOOM`

where $n > 0$

The default value is 4.

Yzoom

From Plot setup (Plot Setup), press MENU then ZOOM. Scroll to `Set Factors`, select it and press OK. Enter the value for `YZOOM` and press OK.

Or, in a program, type:

n ▶ `YZOOM`

The default value is 4.

Symbolic view variables

AltHyp
Inference

Determines the alternative hypothesis used for a hypothesis testing. Choose an option from the Symbolic view.

Or, in a program, type:

0 ▶ AltHyp—for $\mu < \mu_0$

1 ▶ AltHyp—for $\mu > \mu_0$

2 ▶ AltHyp—for $\mu \neq \mu_0$

E0...E9
Solve

Can contain any equation or expression. Independent variable is selected by highlighting it in Numeric View.

Example:

```
X+Y*X-2=Y▶ E1
```

F0...F9
Function

Can contain any expression. Independent variable is X.

Example:

```
SIN(X)▶ F1
```

H1...H5

Contains the data values for a 1-variable statistical analysis. For example, H1(n) returns the nth value in the data set for the H1 analysis.

H1Type...H5Type

Sets the type of plot used to graphically represent the statistical analyses H1 through H5. From the Symbolic setup, specify the type of plot in the field for Type1, Type 2, etc.

Or in a program, store one of the following constant integers or names into the variables H1Type, H2Type, etc.

0 Histogram (default)

1 Box and Whisker

2 Normal Probability

3 Line

4 Bar

5 Pareto

Example:

```
2▶H3Type
```

Method
Inference

Determines whether the Inference app is set to calculate hypothesis test results or confidence intervals.

Or, in a program, type:

`0 ▶ Method`—for `Hypothesis Test`

`1 ▶ Method`—for `Confidence Interval`

R0...R9
Polar

Can contain any expression. Independent variable is θ.

Example:

```
2*SIN(2*θ)▶ R1
```

S1Type...S5Type
Statistics 2Var

Sets the type of fit to be used by the `FIT` operation in drawing the regression line. From Symbolic Setup view, specify the fit in the field for `Type1,Type2`, etc.

Or, in a program, store one of the following constant integers or names into a variable `S1Type,S2Type`, etc.

```
1 Linear
2 Logarithmic
3 Exponential
4 Power
5 Exponent
6 Inverse
7 Logistic
8 Quadratic
9 Cubic
10 Quartic
11 User Defined
```

Example:

```
Cubic ▶ S2fit
```

or

```
9 ▶ S2fit
```

Type
Inference

Determines the type of hypothesis test or confidence interval. Depends upon the value of the variable `Method`. Make a selection from the Symbolic view.

Or, in a program, store the constant number from the list below into the variable Type. With `Method=0`, the constant values and their meanings are as follows:

```
0 Z-Test:1 μ
1 Z-Test:μ1 – μ2
2 Z-Test:1 π
3 Z-Test:π1 – π2
4 T-Test:1 μ
5 T-Test:μ1 – μ2
```

With `Method=1`, the constant values and their meanings are as follows:

```
0 Z-Int:1 μ
1 Z-Int:μ1 – μ2
2 Z-Int:1 π
3 Z-Int:π1 – π2
4 T-Int:1 μ
5 T-Int:μ1 – μ2
```

X0, Y0...X9,Y9

Parametric

Can contain any expression. Independent variable is T.

Example:

```
SIN(4*T)▶ Y1;2*SIN(6*T)▶ X1
```

U0...U9

Sequence

Can contain any expression. Independent variable is N.

Example:

```
RECURSE (U,U(N-1)*N,1,2) ▶U1
```

Numeric view variables

C0...C9

Statistics 2Var

C0 through C9, for columns of data. Can contain lists.

Enter data in the Numeric view

Or, in a program, type:

`LIST ▶ Cn`

where n = 0, 1, 2, 3 ... 9 and `LIST` is either a list or the name of a list.

D0...D9

Statistics 1Var

D0 through D9, for columns of data. Can contain lists.

Enter data in the Numeric view

Or, in a program, type:

`LIST ▶ Dn`

where n = 0, 1, 2, 3 ... 9 and `LIST` is either a list or the name of a list.

NumCol

Function
Parametric
Polar
Sequence

Sets the column to be highlighted in the Numeric view.

Or, in a program, type:

n `▶ NumCol`

where n can be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

NumFont

Enables you to choose the font size in the Numeric view. Does not appear in the Num Setup input form. Corresponds to BIG• in the Numeric view.

Or, in a program, type:

`0 ▶ NumFont`—for small.

`1 ▶ NumFont`—for big (default).

NumIndep

Function
Parametric
Polar
Sequence

Specifies the list of independent values to be used by `Build Your Own Table`. Enter your values one-by-one in the Numeric view.

Or, in a program, type:

```
LIST ▶ NumIndep
```

`List` can be either a list itself or the name of a list.

NumRow
Function
Parametric
Polar
Sequence

Sets the row to be highlighted in the Numeric view. Use the cursor keys to select a row in the Numeric view.

Or, in a program, type:

n ▶ `NumRow`

where $n > 0$

NumStart
Function
Parametric
Polar
Sequence

Sets the starting value for a table in Numeric view.

From Num Setup, enter a value for `NUMSTART`.

Or, in a program, type:

n ▶ `NumStart`

NumStep
Function
Parametric
Polar
Sequence

Sets the step size (increment value) for an independent variable in Numeric view.

From Num Setup, enter a value for `NUMSTEP`.

Or, in a program, type:

n ▶ `NumStep`

where $n > 0$

NumType
Function
Parametric
Polar
Sequence

Sets the table format.

From Num Setup, enter `0` or `1`.

Or, in a program, type:

`0 ▶ NumType`—for `BuildYourOwn`.

`1 ▶ NumType`—for `Automatic` (default).

NumZoom
Function
Parametric
Polar
Sequence

Sets the zoom factor in the Numeric view.

From Num Setup, type in a value for `NUMZOOM`.

Or, in a program, type:

n ▶ `NumZoom`

where $n > 0$

Inference app variables

The following variables are used by the Inference app. They correspond to fields in the Inference app Numeric view. The set of variables shown in this view depends on the hypothesis test or the confidence interval selected in the Symbolic view.

Alpha

Sets the alpha level for the hypothesis test. From the Numeric view, set the value of `Alpha`.

Or, in a program, type:

n ▶ `Alpha`

where $0 < n < 1$

Conf

Sets the confidence level for the confidence interval. From the Numeric view, set the value of `Conf`.

Or, in a program, type:

n ▶ `Conf`

where $0 < n < 1$

Mean1

Sets the value of the mean of a sample for a 1-mean hypothesis test or confidence interval. For a 2-mean test or interval, sets the value of the mean of the first sample. From the Numeric view, set the value of `Mean1`.

Or, in a program, type:

n ▶ `Mean1`

Mean2

For a 2-mean test or interval, sets the value of the mean of the second sample. From the Numeric view, set the value of `Mean2`.

Or, in a program, type:

n ▶ `Mean2`

The following variables are used to set up hypothesis test or confidence interval calculations in the Inference app.

$\mu 0$

Sets the assumed value of the population mean for a hypothesis test. From the Numeric view, set the value of $\mu 0$.

Or, in a program, type:

n ▶ $\mu0$

where $0 < \mu0 < 1$

n1

Sets the size of the sample for a hypothesis test or confidence interval. For a test or interval involving the difference of two means or two proportions, sets the size of the first sample. From the Numeric view, set the value of `n1`.

Or, in a program, type:

n ▶ `n1`

n2

For a test or interval involving the difference of two means or two proportions, sets the size of the second sample. From the Numeric view, set the value of `n2`.

Or, in a program, type:

n ▶ `n2`

$\pi0$

Sets the assumed proportion of successes for the One-proportion Z-test. From the Numeric view, set the value of $\pi0$.

Or, in a program, type:

n ▶ $\pi0$

where $0 < \pi0 < 1$

Pooled

Determine whether or not the samples are pooled for tests or intervals using the Student's T-distribution involving two means. From the Numeric view, set the value of `Pooled`.

Or, in a program, type:

`0` ▶ `Pooled`—for not pooled (default).

`1` ▶ `Pooled`—for pooled.

s1

Sets the sample standard deviation for a hypothesis test or confidence interval. For a test or interval involving the difference of two means or two proportions, sets the sample standard deviation of the first sample. From the Numeric view, set the value of `s1`.

Or, in a program, type:

n ▶ s1

s2

For a test or interval involving the difference of two means or two proportions, sets the sample standard deviation of the second sample. From the Numeric view, set the value of s2.

Or, in a program, type:

n ▶ s2

σ1

Sets the population standard deviation for a hypothesis test or confidence interval. For a test or interval involving the difference of two means or two proportions, sets the population standard deviation of the first sample. From the Numeric view, set the value of σ1.

Or, in a program, type:

n ▶ σ1

σ2

For a test or interval involving the difference of two means or two proportions, sets the population standard deviation of the second sample. From the Numeric view, set the value of σ2.

Or, in a program, type:

n ▶ σ2

x1

Sets the number of successes for a one-proportion hypothesis test or confidence interval. For a test or interval involving the difference of two proportions, sets the number of successes of the first sample. From the Numeric view, set the value of x1.

Or, in a program, type:

n ▶ x1

x2

For a test or interval involving the difference of two proportions, sets the number of successes of the second sample. From the Numeric view, set the value of x2.

Or, in a program, type:

n ▶ x2

Finance app variables

The following variables are used by the Finance app. They correspond to the fields in the Finance app Numeric view.

CPYR

Compounding periods per year. Sets the number of compounding periods per year for a cash flow calculation. From the Numeric view of the Finance app, enter a value for C/YR.

Or, in a program, type:

n ▶CPYR

where $n > 0$

END

Determines whether interest is compounded at the beginning or end of the compounding period. From the Numeric view of the Finance app. Check or uncheck END.

Or, in a program, type:

1▶END—for compounding at the end of the period (Default)

0▶END—for compounding at the beginning of the period

FV

Future value. Sets the future value of an investment. From the Numeric view of the Finance app, enter a value for FV.

Or, in a program, type:

n ▶FV

Note: positive values represent return on an investment or loan.

IPYR

Interest per year. Sets the annual interest rate for a cash flow. From the Numeric view of the Finance app, enter a value for I%YR.

Or, in a program, type:

n ▶IPYR

where $n > 0$

NbPmt

Number of payments. Sets the number of payments for a cash flow. From the Numeric view of the Finance app, enter a value for N.

Or, in a program, type:

n ▶`NbPmt`

where $n > 0$

PMT

Payment value. Sets the value of each payment in a cash flow. From the Numeric view of the Finance app, enter a value for `PMT`.

Or, in a program, type:

n ▶`PMT`

Note: payment values are negative if you are making the payment and positive if you are receiving the payment.

PPYR

Payments per year. Sets the number of payments made per year for a cash flow calculation. From the Numeric view of the Finance app, enter a value for `P/YR`.

Or, in a program, type:

n ▶`PPYR`

where $n > 0$

PV

Present value. Sets the present value of an investment. From the Numeric view of the Finance app, enter a value for `PV`.

Or, in a program, type:

n ▶`PV`

Note: negative values represent an investment or loan.

GSize

Group size. Sets the size of each group for the amortization table. From the Numeric view of the Finance app, enter a value for `Group Size`.

Or, in a program, type:

n ▶`GSize`

Linear Solver app variables

The following variables are used by the Linear Solver app. They correspond to the fields in the app's Numeric view.

LSystem

Contains a 2x3 or 3x4 matrix which represents a 2x2 or 3x3 linear system. From the Numeric view of the Linear

Solver app, enter the coefficients and constants of the linear system.

Or, in a program, type:

`matrix▶LSystem`

where `matrix` is either a matrix or the name of one of the matrix variables M0-M9.

Size

Contains the size of the linear system. From the Numeric view of the Linear Solver app, press 2X2 or 3X3.

Or, from a program, type:

`2▶Size`—for a 2x2 linear system

`3▶Size`—for a 3x3 linear system

Triangle Solver app variables

The following variables are used by the Triangle Solver app. They correspond to the fields in the app's Numeric view.

SideA

The length of Side A. Sets the length of the side opposite the angle A. From the Triangle Solver Numeric view, enter a positive value for A.

Or, in a program, type:

n▶`SideA`

where $n > 0$

SideB

The length of Side B. Sets the length of the side opposite the angle B. From the Triangle Solver Numeric view, enter a positive value for B.

Or, in a program, type:

n▶`SideB`

where $n > 0$

SideC

The length of Side C. Sets the length of the side opposite the angle C. From the Triangle Solver Numeric view, enter a positive value for C.

Or, in a program, type:

n▶`SideC`

where $n > 0$

AngleA

The measure of angle A. Sets the measure of angle A. The value of this variable will be interpreted according to the angle mode setting (`Degrees` or `Radians`). From the Triangle Solver Numeric view, enter a positive value for A.

Or, in a program, type:

n ▶`AngleA`

where $n > 0$

AngleB

The measure of angle B. Sets the measure of angle B. The value of this variable will be interpreted according to the angle mode setting (`Degrees` or `Radians`). From the Triangle Solver Numeric view, enter a positive value for B.

Or, in a program, type:

n ▶`AngleB`

where $n > 0$

AngleC

The measure of angle C. Sets the measure of angle C. The value of this variable will be interpreted according to the angle mode setting (`Degrees` or `Radians`). From the Triangle Solver Numeric view, enter a positive value for C.

Or, in a program, type:

n ▶`AngleC`

where $n > 0$

RECT

Corresponds to the status of RECT in the Numeric view of the Triangle Solver app. Determines whether a general triangle solver or a right triangle solver is used. From the Triangle Solver view, press RECT.

Or, in a program, type:

`0`▶`RECT`—for the general Triangle Solver

`1`▶`RECT`—for the right Triangle Solver

Modes variables

These variables are found in the Symbolic setup of an app. They can be used to overwrite the value of the corresponding variable in Home Modes.

AAngle

Sets the angle mode.

From Modes view, choose `System`, `Degrees`, or `Radians` for angle measure. `System` (default) will force the angle measure to agree with that in Modes. In the Statistics app, you can set this from Symbolic Setup as well.

Or, in a program, type:

`1 ▶ AAngle`—for System.

`2 ▶ AAngle`—for Degrees.

`3 ▶ AAngle`—for Radians.

ADigits

Number of decimal places to use for `Number format` in the Home view and for labelling axes in the Plot view.

From the Modes view, enter a value in the second field of `Number Format`.

Or, in a program, type:

n ▶ `ADigits`

where $0 < n < 11$

AFormat

Defines the number display format used for number display in the Home view and to label axes in the Plot view.

From the Modes view, choose `Standard`, `Fixed`, `Scientific`, or `Engineering` in the Number Format field.

Or, in a program, store the constant number (or its name) into the variable Format.

```
1 Standard
2 Fixed
3 Sci
4 Eng
```

Example:

```
Scientific ▶ AFormat
```

or

```
3 ▶ AFormat
```

Results variables

These variables are found in various views. They capture the results of calculations such as those performed when the STATS menu key is pressed in the Statistics 1Var Numeric view.

The following results variables store calculations from the Function app. They store results from the commands in the Plot view FCN menu.

Area

Contains the last value found by the `Signed area` function in the Plot-FCN menu.

Extremum

Contains the last value found by the `Extremum` operation in the Plot-FCN menu.

Isect

Contains the last value found by the `Intersection` function in the Plot-FCN menu.

Root

Contains the last value found by the `Root` function in the Plot-FCN menu.

Slope

Contains the last value found by the `Slope` function in the Plot-FCN menu.

The following Results variable stores calculations from the Linear Solver app. These calculations correspond to the solution to a 2x2 or 3x3 linear system.

LSolution

Contains a vector with the last solution found by either the Linear Solver app or the `LSolve` app function.

The following Results variables store calculations from the Statistics 1Var app. These calculations are performed when STATS is pressed in the Numeric view or the `Do1VarStats` command is executed.

NbItem

Contains the number of data points in the current 1-variable analysis (`H1-H5`).

Min

Contains the minimum value of the data set in the current 1-variable analysis (`H1-H5`).

Q1

Contains the value of the first quartile in the current 1-variable analysis (`H1-H5`).

Med

Contains the median in the current 1-variable analysis (`H1-H5`).

Q3 Contains the value of the third quartile in the current 1-variable analysis (`H1-H5`).

Max Contains the maximum value in the current 1-variable analysis (`H1-H5`).

ΣX Contains the sum of the data set in the current 1-variable analysis (`H1-H5`).

ΣX² Contains the sum of the squares of the data set in the current 1-variable analysis (`H1-H5`).

MeanX Contains the mean of the data set in the current 1-variable analysis (`H1-H5`).

sX Contains the sample standard deviation of the data set in the current 1-variable analysis (`H1-H5`).

σX Contains the population standard deviation of the data set in the current 1-variable analysis (`H1-H5`).

serrX Contains the standard error of the data set in the current 1-variable analysis (`H1-H5`).

The following Results variables store calculations from the Statistics 2Var app. These calculations are performed when STATS is pressed in the Numeric view or the `Do2VarStats` command is executed.

Corr Contains the correlation coefficient from the latest calculation of summary statistics. This value is based on the linear fit only, regardless of the fit type chosen.

CoefDet Contains the coefficient of determination from the latest calculation of summary statistics. This value is based on the fit type chosen.

sCOV Contains the sample covariance of the current 2-variable statistical analysis (`S1-S5`).

σCOV Contains the population covariance of the current 2-variable statistical analysis (`S1-S5`).

ΣXY Contains the sum of the X·Y products for the current 2-variable statistical analysis (`S1-S5`).

MeanX Contains the mean of the independent values (X) of the current 2-variable statistical analysis (`S1-S5`).

ΣX Contains the sum of the independent values (X) of the current 2-variable statistical analysis (`S1-S5`).

ΣX^2 Contains the sum of the squares of the independent values (X) of the current 2-variable statistical analysis (`S1-S5`).

sX Contains the sample standard deviation of the independent values (X) of the current 2-variable statistical analysis (`S1-S5`).

σX Contains the population standard deviation of the independent values (X) of the current 2-variable statistical analysis (`S1-S5`).

serrX Contains the standard error of the independent values (X) of the current 2-variable statistical analysis (`S1-S5`).

MeanY Contains the mean of the dependent values (Y) of the current 2-variable statistical analysis (`S1-S5`).

ΣY Contains the sum of the dependent values (Y) of the current 2-variable statistical analysis (`S1-S5`).

ΣY^2 Contains the sum of the squares of the dependent values (Y) of the current 2-variable statistical analysis (`S1-S5`).

sY Contains the sample standard deviation of the dependent values (Y) of the current 2-variable statistical analysis (`S1-S5`).

σY Contains the population standard deviation of the dependent values (Y) of the current 2-variable statistical analysis (`S1-S5`).

serrY Contains the standard error of the dependent values (Y) of the current 2-variable statistical analysis (`S1-S5`).

The following Results variables store calculations from the Inference app. These calculations are performed when CALC is pressed in the Numeric view.

CritScore Contains the value of the Z- or t-distribution associated with the input α-value

CritVal1 Contains the lower critical value of the experimental variable associated with the negative TestScore value which was calculated from the input α-level.

CritVal2 Contains the upper critical value of the experimental variable associated with the positive TestScore value which was calculated from the input α-level.

DF Contains the degrees of freedom for the t-tests.

Prob Contains the probability associated with the TestScore value.

Result For hypothesis tests, contains 0 or 1 to indicate rejection or failure to reject the null hypothesis.

TestScore Contains the Z- or t-distribution value calculated from the hypothesis test or confidence interval inputs.

TestValue Contains the value of the experimental variable associated with the TestScore.

App Functions

App functions are used by several of the HP Apps to perform common calculations. For example, in the Function app, the Plot view FCN menu has a function called SLOPE that calculates the slope of a given function at a given point. The SLOPE function can be used, from the Home view or a program, etc. to give the same results as if you were in the Function app Plot view. App functions can be used to get the same results in a program or the Home view or anywhere else- just as if you were in the app. The App functions described in this section are grouped by app.

Function app functions

The Function app functions provide the same functionality found in the Function app's Plot view under the FCN menu. All of these operations work on functions. The functions may be expressions in X or the names of the Function app variable F0 through F9.

AREA Area under a curve or between curves. Finds the signed area under a function or between two functions. Finds the area under the function Fn or below Fn and above the function Fm, from lower X-value to upper X-value.

AREA(*Fn*, [*Fm*,] *lower, upper*)

Example:

AREA(-X, X^2-2, -2, 1) returns 4.5

EXTREMUM

Extremum of a function. Finds the extremum (if one exists) of the function Fn that is closest to the X-value guess.

EXTREMUM(*Fn, guess*)

Example:

EXTREMUM(X^2-X-2, 0) returns 0.5

ISECT

Intersection of two functions. Finds the intersection (if one exists) of the two functions Fn and Fm that is closest to the X-value guess.

ISECT(*Fn, Fm, guess*)

Example:

ISECT(X, 3-X,2) returns 1.5

ROOT

Root of a function. Finds the root of the function Fn (if one exists) that is closest to the X-value guess.

ROOT(*Fn, guess*)

Example:

ROOT(3-X^2, 2) returns 1.732…

SLOPE

Slope of a function. Returns the slope of the function Fn at the X-value (if value exists).

SLOPE(*Fn, value*)

Example:

SLOPE(3-X^2, 2) returns -4

Solve app functions

The Solve app has a single function that solves a given equation or expression for one of its variables. *En* may be an equation or expression, or it may be the name of one of the Solve Symbolic variables E0-E9.

SOLVE

Solve. Solves an equation for one of its variables. Solves the equation *En* for the variable *var*, using the value of *guess* as the initial value for the value of the variable *var*. If *En* is an expression, then the value of the variable *var* that makes the expression equal to zero is returned.

`SOLVE`(*En, var, guess*)

Example:

`SOLVE`(X^2-X-2, `X`, 3) returns 2

This function also returns an integer that is indicative of the type of solution found, as follows:

`0`—an exact solution was found

`1`—an approximate solution was found

`2`—an extremum was found that is as close to a solution as possible

`3`—neither a solution, an approximation, nor an extremum was found

See the Chapter *Solve app* for more details on the types of solutions returned by this function.

Statistics 1Var app functions

The Statistics 1Var app has a 3 functions designed to work together to calculate summary statistics based on one of the statistical analyses (`H1-H5`) defined in the Symbolic view of the Statistics 1Var app.

Do1VStats

Performs the same calculations as pressing STATS in the Statistics 1Var app's Numeric view and stores the results in the appropriate Statistics 1Var app results variables. *Hn* must be one of the Statistics 1Var app Symbolic view variables `H1-H5`.

`Do1VStats`*(Hn)*

SETFREQ

Set frequency. Sets the frequency for one of the statistical analyses (`H1-H5`) defined in the Symbolic view of the Statistics 1Var app. The frequency can be either one of the column variables `D0-D9`, or any positive integer. *Hn* must be one of the Statistics 1Var app Symbolic view variables `H1-H5`. If used, *Dn* must be one of the column variables `D0-D9`; otherwise, *value* must be a positive integer.

`SETFREQ`(*Hn, Dn*)

or

`SETFREQ`(*Hn, value*)

SETSAMPLE

Set sample data. Sets the sample data for one of the statistical analyses (`H1-H5`) defined in the Symbolic view of the Statistics 1Var app. Sets the data column to one of the column variables `D0-D9` for one of the statistical analyses `H1-H5`.

`SETSAMPLE(`*Hn, Dn*`)`

Statistics 2Var app functions

The Statistics 2Var app has a number of functions. Some are designed to calculate summary statistics based on one of the statistical analyses (`S1-S5`) defined in the Symbolic view of the Statistics 2Var app. Others predict X- and Y-values based on the fit specified in one of the analyses.

Do2VStats

Performs the same calculations as pressing STATS in the Statistics 2Var app's Numeric view and stores the results in the appropriate Statistics 2Var app results variables. Sn must be one of the Statistics 2Var app Symbolic view variables `S1-S5`.

`Do2VStats(`*Sn*`)`

PredX

Predict X. Uses the fit from the first active analysis (`S1-S5`) found to predict an x-value given the y-value value.

`PredX(`*value*`)`

PredY

Predict Y. Uses the fit from the first active analysis (`S1-S5`) found to predict a y-value given the x-value value.

`PredY(`*value*`)`

SetDepend

Sets the dependent column for one of the statistical analyses `S1-S5` to one of the column variables `C0-C9`.

`SetDepend(`*Sn, Cn*`)`

SetIndep

Sets the independent column for one of the statistical analyses S1-S5 to one of the column variables `C0-C9`.

`SetIndep(`*Sn, Cn*`)`

Inference app functions

The Inference app has a single function that returns the same results as pressing CALC in the Inference app's Numeric view. The results depend on the contents of the Inference app variables `Method`, `Type`, and `AltHyp`.

DoInference

Calculate confidence interval or test hypothesis. Performs the same calculations as pressing CALC in the Inference app's Numeric view and stores the results in the appropriate Inference app results variables.

`DoInference()`

Sequence app functions

The Sequence app has a single function for defining a sequence and storing it into one of the Sequence app Symbolic variables `U0-U9`.

RECURSE

Recursion. Provides a method for defining a sequence without using the Symbolic view of the Sequence app. *Un* is one of the Sequence app Symbolic view variables `U0-U9`, *nthterm* is an expression in N, *Un(N-1)*, and/or *Un(N-2)*, and term1 and term2 are the first two terms of the sequence. If *nthterm* is an expression only in *N*, then *term1* and *term2* are optional.

`RECURSE(`*Un, nthterm* [, *term1, term2*])

Examples:

`RECURSE(U1, N)` defines the sequence U1 to be the sequence of counting numbers.

`RECURSE(U2, U2(N-1)*N, 1)` defines U2 to create a factorial-calculating sequence. U2(5), for example, will return 120.

`RECURSE(U3, U3(N-2)+U3(N-1), 1, 1)` defines U3 to be the Fibonacci sequence.

Finance app functions

The Finance app has a single function that returns the same results as pressing SOLVE in the Finance app's Numeric view.

DoFinance

Calculate TVM results. Solves a TVM problem for the variable *TVMVar*. The variable must be one of the Finance app's Numeric view variables. Performs the same calculation as pressing SOLVE in the Finance app Numeric view with *TVMVar* highlighted.

`DoFinance(`*TVMVar*)

Example:

`DoFinance(FV)` returns the future value of an investment in the same way as pressing SOLVE in the Finance app Numeric view with `FV` highlighted.

Linear Solver app functions

The Linear Solver app has a single function that solves a 2x2 or 3x3 linear system, based on a matrix of coefficients and constants.

LinSolve

Solve linear system. Solves the 2x2 or 3x3 linear system represented by matrix.

`LinSolve`(*matrix*)

Examples:

`LinSolve([[A, B, C], [D, E,F]])` solves the linear system:

$$\begin{cases} ax + by = c \\ dx + ey = f \end{cases}$$

Triangle Solver app functions

The Triangle Solver app has a group of functions which allow solving a complete triangle from the input of 3 consecutive parts of the triangle. The names of these commands use A to signify an angle, and S to signify a side length. To use these commands, enter 3 inputs in the specified order given by the command name.

AAS

`AAS`(*angle, angle, side*)

ASA

`ASA`(*angle, side, angle*)

SAS

`SAS`(*side, angle, side*)

SSA

`SSA`(*side, side, angle*)

SSS

`SSS`(*side, side, side*)

Returns a list of three results that correspond to the opposite angle or side from the given input. If the input given was an angle, the item in same position in the results list will be the opposite side.

Example:

In Degree mode, `SAS`(*2, 90, 2*) returns { 45, 2.82…,45}.

In the indeterminate case `AAS` where two solutions may be possible, `AAS` may return a list of two such lists containing both results.

Common app functions

In addition to the app functions specific to each app, there are two functions common to the following apps:

- Function
- Solve
- Statistics 1Var
- Statistics 2Var
- Parametric
- Polar
- Sequence

CHECK

Checks the Symbolic view variable `Symbn`. `Symbn` can be any of the following:

- `F0-F9`—for the function app
- `E0-E9`—for the Solve app
- `H1-H5`—for the Statistics 1Var app
- `S1-S5`—for the Statistics 2Var app
- `X0/Yo-X9/Y9`—for the parametric app
- `R0-R9`—for the Polar app
- `U0-U9`—for the Sequence app

`CHECK`**(***Symbn***)**

Example:

`CHECK(F1)` checks the Function app Symbolic view variable F1. The result is that F1(X) is drawn in the Plot view and has a column of function values in the Numeric view of the Function app.

UNCHECK

Unchecks the Symbolic view variable `Symbn`.

`UNCHECK`**(***Symbn***)**

Example:

`UNCHECK(R1)` unchecks the Polar app Symbolic view variable R1. The result is that R1(θ) is not drawn in the Plot view and does not appear in the Numeric view of the Polar app.

22

Reference information

Glossary

app	A small application, designed to study one or more related topics or to solve problems of a particular type. The built-in apps are Function, Solve, Statistics 1Var, Statistics 2Var, Inference, Parametric, Polar, Sequence, Finance, Linear Solver, Triangle Solver, Linear Explorer, Quadratic Explorer, and Trig Explorer. An app can be filled with the data and solutions for a specific problem. It is reusable (like a program, but easier to use) and it records all your settings and definitions.
command	An operation for use in programs. Commands can store results in variables, but do not display results.
expression	A number, variable, or algebraic expression (numbers plus functions) that produces a value.
function	An operation, possibly with arguments, that returns a result. It does not store results in variables. The arguments must be enclosed in parentheses and separated with commas.
Home	The basic starting point of the calculator. Go to Home to do calculations.
Library	For app management: to start, save, reset, send and receive apps.

list	A set of values separated by commas and enclosed in braces. Lists are commonly used to enter statistical data and to evaluate a function with multiple values. Created and manipulated by the List editor and catalog.
matrix	A two-dimensional array of values separated by commas and enclosed in nested brackets. Created and manipulated by the Matrix catalog and editor. Vectors are also handled by the Matrix catalog and editor.
menu	A choice of options given in the display. It can appear as a list or as a set of *menu-key labels* across the bottom of the display.
menu keys	The top row of keys. Their operations depend on the current context. The labels along the bottom of the display show the current meanings.
note	Text that you write in the Note Editor or the Info view of an app.
program	A reusable set of instructions that you record using the Program editor.
variable	The name of a number, list, matrix, or graphic that is stored in memory. Use STO ▸ to store and use Vars Chars A to retrieve.
vector	A one-dimensional array of values separated by commas and enclosed in single brackets. Created and manipulated by the Matrix catalog and editor.
views	The possible contexts for an app: Plot, Plot Setup, Numeric, Numeric Setup, Symbolic, Symbolic Setup, Info, and special views like split screens.

Resetting the HP 39gII

If the calculator "locks up" and seems to be stuck, you must reset it. This is much like resetting a PC. It cancels certain operations, restores certain conditions, and clears temporary memory locations. However, it does *not* clear stored data (variables, app databases, programs) *unless* you use the procedure below, "To erase all memory and reset defaults".

To reset

Press and hold ON/C and F3 simultaneously, then release them.

To erase all memory and reset defaults

If the calculator does not respond to the above resetting procedures, you might need to restart it by erasing all of memory. *You will lose everything you have stored.* All factory-default settings are restored.

1. Press and hold ON/C, F1,and F6 simultaneously.
2. Release all keys in the reverse order.

If the calculator does not turn on

If the HP 39gII does not turn on, follow the steps below until the calculator turns on. You may find that the calculator turns on before you have completed the procedure. If the calculator still does not turn on, please contact Customer Support for further information.

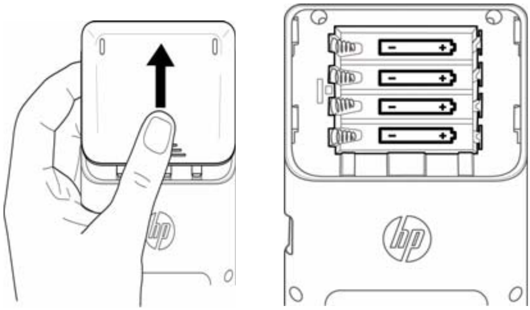
1. Press and hold ON/C for 10 seconds, then release.
2. Press and hold ON/C and F3 simultaneously, then release F3, then release ON/C.
3. Press and hold ON/C, F1,and F6 simultaneously. Release F6, then release F1, and then release ON/C.

4. Remove the batteries, press and hold ON/C (OFF) for 10 seconds, then put the batteries back in and press ON/C (OFF).

Batteries

The calculator takes 4 AAA (LR03) batteries as a main power source.

To install batteries



Warning: When the battery annunciator indicates that the batteries are low, you need to replace the batteries as soon as possible.

Please install the batteries according to the following procedure:

1. Turn off the calculator.
2. Slide up the battery compartment cover.
3. Insert 4 new AAA (LR03) batteries into the compartment.
4. Make sure each battery is inserted in the indicated direction.

5. After installing the batteries, press ON/C (OFF) to turn the calculator on.

Warning! There is danger of explosion if the battery is incorrectly replaced. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions. Do not mutilate, puncture, or dispose of batteries in fire. The batteries can burst or explode, releasing hazardous chemicals.

Operating details

Operating temperature: 0° to 45°C (32° to 113°F).

Storage temperature: –20° to 65°C (– 4° to 149°F).

Operating and storage humidity: 90% relative humidity at 40°C (104°F) maximum. *Avoid getting the calculator wet.*

Battery operates at 6.0V dc, 80mA maximum.

Variables

Home variables

The Home variables are:

Category	Available names
Complex	`Z1...Z9, Z0`
Graphic	`G1...G9, G0`

Library	`Function`
	`Solve`
	`Statistics 1Var`
	`Statistics 2Var`
	`Inference`
	`Parametric`
	`Polar`
	`Sequence`
	`Finance`
	`Linear Solver`
	`Triangle Solver`
	User-named programs

List	`L1...L9, L0`
Matrix	`M1...M9, M0`

Category	Available names (Continued)
Modes	Ans HAngle HDigits HFormat HComplex Language
Program	Function Solve Statistics 1Var Statistics 2Var Inference Parametric Polar Sequence Finance Linear Solver Triangle Solver *User-named programs*
Real	A...Z, θ

App variables

Function app variables

The Function app variables are:

Category	Available names	
Results	Area Extremum Isect	Root Slope
Symbolic	F1 F2 F3 F4 F5	F6 F7 F8 F9 F0
Plot	Axes Cursor GridDots GridLines Labels Method Recenter Tracing	Xmax Xmin Xtick Xzoom Ymax Ymin Ytick Yzoom

Category	Available names (Continued)	
Numeric	NumStart NumStep	NumType NumZoom
Modes	AAngle AComplex	ADigits AFormat

Solve app variables

The Solve app variables are:

Category	Available names	
Symbolic	E1 E2 E3 E4 E5	E6 E7 E8 E9 E0
Plot	Axes Cursor GridDots GridLines Labels Method Recenter Tracing	Xmax Xmin Xtick Xzoom Ymax Ymin Ytick Yzoom
Modes	AAngle AComplex	ADigits AFormat

Statistics 1Var app variables

The Statistics 1Var app variables are:

Category	Available names	
Results	NbItem Min Q1 Med Q3 Max	ΣX ΣX2 MeanX sX σX serrX

Category	Available names	(Continued)
Symbolic	H1 H2 H3 H4 H5	H1Type H2Type H3Type H4Type H5Type
Plot	Axes Cursor GridDots GridLines Labels Method Recenter Tracing	Xmax Xmin Xtick Xzoom Ymax Ymin Ytick Yzoom
Numeric	D1 D2 D3 D4 D5	D6 D7 D8 D9 D0
Modes	AAngle AComplex	ADigits AFormat

Statistics 2Var app variables

The Statistics 2Var app variables are:

Category	Available names	
Results	NbItem Corr CoefDet sCov σCov ΣXY MeanX ΣX ΣX2	sX σX serrX MeanY ΣY ΣY2 sY σY serrY
Symbolic	S1 S2 S3 S4 S5	S1Type S2Type S3Type S4Type S5Type

Category	Available names	(Continued)
Plot	Axes Cursor GridDots GridLines Labels Method Recenter Tracing	Xmax Xmin Xtick Xzoom Ymax Ymin Ytick Yzoom
Numeric	C1 C2 C3 C4 C5	C6 C7 C8 C9 C0
Modes	AAngle AComplex	ADigits AFormat

Inference app variables

The Inference app variables are:

Category	Available names	
Results	Result TestScore TestValue Prob DF	CritScore CritVal1 CritVal2
Symbolic	AltHyp Method	Type
Numeric	Alpha Conf Mean1 Mean2 n1 n2 μ0 π0	Pooled s1 s2 σ1 σ2 x1 x2
Modes	AAngle AComplex	ADigits AFormat

Parametric app variables

The Parametric app variables are:

Category	Available names	
Symbolic	X1 Y1 X2 Y2 X3 Y3 X4 Y4 X5 Y5	X6 Y6 X7 Y7 X8 Y8 X9 Y9 X0 Y0
Plot	Axes Cursor GridDots GridLines Labels Method Recenter Tracing	Xmax Xmin Xtick Xzoom Ymax Ymin Ytick Yzoom
Numeric	NumStart NumStep	NumType NumZoom
Modes	AAngle AComplex	ADigits AFormat

Polar app variables

The Polar app variables are:

Category	Available names	
Symbolic	R1 R2 R3 R4 R5	R6 R7 R8 R9 R0

Category	Available names (Continued)	
Plot	Axes Cursor GridDots GridLines Labels Method Recenter Tracing	Xmax Xmin Xtick Xzoom Ymax Ymin Ytick Yzoom
Numeric	NumStart NumStep	NumType NumZoom
Modes	AAngle AComplex	ADigits AFormat

Sequence app variables

The Sequence app variables are:

Category	Available names	
Symbolic	U1 U2 U3 U4 U5	U6 U7 U8 U9 U0
Plot	Axes Cursor GridDots GridLines Labels Method Recenter Tracing	Xmax Xmin Xtick Xzoom Ymax Ymin Ytick Yzoom
Functions	NumStart NumStep	NumType NumZoom
Modes	AAngle AComplex	ADigits AFormat

Finance app variables

The Finance app variables are:

Category	Available names	
Numeric	CPYR END FV GSize IPYR	NbPmt PMT PPYR PV

Linear Solver app variables

The Linear solver app variables are:

Category	Available names	
Results	LSolution	
Numeric	LSystem	Size
Modes	AAngle AComplex	ADigits AFormat

Triangle Solver app variables

The Triangle solver app variables are:

Category	Available names	
Numeric	AngleA AngleB AngleC Rect	SideA SideB SideC
Modes	AAngle AComplex	ADigits AFormat

Linear Explorer app variables

The Linear Explorer app variables are:

Category	Available names	
Modes	AAngle AComplex	ADigits AFormat

Quadratic Explorer app variables

The Quadratic Explorer app variables are:

Category	Available names	
Modes	AAngle AComplex	ADigits AFormat

Trig Explorer app variables

The Trig Explorer app variables are:

Category	Available names	
Modes	AAngle AComplex	ADigits AFormat

Functions and Commands

Math menu functions

The Math menu functions are:

Category	Available functions	
Calculus	∂ ∫ | (Where)	
Complex	ARG CONJ	IM RE
Constant	e i	MAXREAL MINREAL π
Distribution	normald normald_cdf normald_icdf binomial binomial_cdf binomial_icdf chisquare chisquare_cdf chisquare_icdf	fisher fisher_cdf fisher_icdf poisson poisson_cdf poisson_icdf student student_cdf student_icdf
Hyperbolic	ACOSH ASINH ATANH COSH SINH	TANH ALOG EXP EXPM1 LNP1
Integer	ichinrem idivis iegcd ifactor ifactors igcd iquo iquorem irem	isprime ithprime nextprime powmod prevprime euler numer denom
List	CONCAT ΔLIST MAKELIST πLIST POS	REVERSE SIZE ΣLIST SORT

Category	Available functions (Continued)	
Loop	ITERATE	
	Σ	
Matrix	COLNORM	QR
	COND	RANK
	CROSS	ROWNORM
	DET	RREF
	DOT	SCHUR
	EIGENVAL	SIZE
	EIGENVV	SPECNORM
	IDENMAT	SPECRAD
	INVERSE	SVD
	LQ	SVL
	LSQ	TRACE
	LU	TRN
	MAKEMAT	
Polynom.	POLYCOEF	POLYROOT
	POLYEVAL	
Prob.	COMB	UTPC
	!	UTPF
	PERM	UTPN
	RANDOM	UTPT
Real	CEILING	MIN
	DEG→RAD	MOD
	FLOOR	%
	FNROOT	%CHANGE
	FRAC	%TOTAL
	HMS→	RAD→DEG
	→HMS	ROUND
	INT	SIGN
	MANT	TRUNCATE
	MAX	XPON
Tests	<	AND
	≤	IFTE
	==	NOT
	≠	OR
	>	XOR
	≥	

Category	Available functions (Continued)	
Trig	ACOT ACSC ASEC	COT CSC SEC

App functions

The app functions are:

Category	Available functions
Function	AREA(*Fn*,[*Fm*,]*lower*,*upper*) EXTREMUM(*Fn*,*guess*) ISECT(Fn,*Fm*,*guess*) ROOT(*Fn*,*guess*) SLOPE(*Fn*,*value*)
Solve	SOLVE(*En*,*var*,*guess*)
Statistics 1Var	Do1VStats(*Hn*) SETFREQ(*Hn*,*Dn*) or SETFREQ(*Hn*,*value*) SETSAMPLE(*Hn*,*Dn*)
Statistics 2Var	Do2VStats(*Sn*) PredX(*value*) PredY(*value*) SetDepend(*Sn*,*Cn*) SetIndep(*Sn*,*Cn*)
Inference	DoInference()
Sequence	RECURSE(*Un*,*nthterm*[,*term1*, *term2*])
Finance	DoFinance(*TVMVar*)
Linear Solver	LinSolve(*matrix*)
Triangle Solver	AAS(*angle*,*angle*,*side*) ASA(*angle*,*side*,*angle*) SAS(*side*,*angle*,*side*) SSA(*side*,*side*,*angle*) SSS(*side*,*side*,*side*)

Program commands

The Program commands are:

Category	Available functions	
App	CHECK UNCHECK STARTAPP	STARTVIEW VIEWS
Block	BEGIN END	RETURN
Branch	IF THEN ELSE	END CASE IFERR
Drawing	PIXON PIXON_P PIXOFF PIXOFF_P GETPIX GETPIX_P RECT RECT_P INVERT INVERT_P ARC ARC_P LINE LINE_P	TEXTOUT TEXTOUT_P BLIT BLIT_P DIMGROB DIMGROB_P SUBGRB SUBGROB_P FREEZE GROBH GROBH_P GROBW GROBW_P
I/O	CHOOSE EDITMAT GETKEY ISKEYDOWN INPUT	MSGBOX PRINT WAIT debug
Loop	FOR FROM TO STEP END DO	UNTIL WHILE REPEAT BREAK CONTINUE
Matrix	ADDCOL ADDROW DELCOL DELROW EDITMAT RANDMAT	REDIM REPLACE SCALE SCALEADD SUB SWAPCOL SWAPROW
Strings	asc char expr string inString	left right mid rotate dim
Variable	EXPORT	LOCAL

Constants

Program constants

The Program constants are:

Category	Available names	
Angle	Degrees Radians	
H1Type...H5Type	Hist BoxW NormalProb LineP BarP ParetoP	
Format	Standard Fixed	Sci Eng
SeqPlot	Cobweb Stairstep	
S1Type...S5Type	Linear LogFit ExpFit Power Inverse Exponent	Logistic QuadFit Cubic Quartic Trig User
Stat1VPlot	Hist BoxW NormalProb LineP BarP ParetoP	

Physical Constants

The Physical constants are:

Category	Available names
Chemistry	Avogadro NA Boltmann, k molar volume, Vm universal gas, R standard temperature, StdT standard pressure, StdP
Phyics	Stefan-Boltzmann, σ speed of light, c permittivity, Σ_0) permeability, μ_0 acceleration of gravity, g gravitation, G
Quantum	Planck, h Dirac *h* electronic charge, q electron mass, me q/me ratio, qme proton mass, mp mp/me ratio, mpme fine structure, α magnetic flux, Φo) Faraday, F Rydberg, R∞ Bohr radius, a_0 Bohr magneton, μB nuclear magneton, μN photon wavelength, λ_0 photon frequency, f_0 Compton wavelength, λ_c

Status messages

Message	Meaning
Bad Argument Type	Incorrect input for this operation.
Bad Argument Value	The value is out of range for this operation.

Message	Meaning (Continued)
Infinity error	Math exception, such as 1/0.
Insufficient Memory	You must recover some memory to continue operation. Delete one or more matrices, lists, notes, or programs (using catalogs), or custom (not built-in) apps (using SHIFT *MEMORY*).
Insufficient Statistics Data	Not enough data points for the calculation. For two-variable statistics there must be two columns of data, and each column must have at least four numbers.
Invalid Dimension	Array argument had wrong dimensions.
Invalid Statistics Data	Need two columns with equal numbers of data values.
Invalid Syntax	The function or command you entered does not include the proper arguments or order of arguments. The delimiters (parentheses, commas, periods, and semi-colons) must also be correct. Look up the function name in the index to find its proper syntax.
Name Conflict	The | (where) function attempted to assign a value to the variable of integration or summation index.
No equations checked	You must enter and check an equation in the Symbolic view before entering the Plot view.
(OFF SCREEN)	Function value, root, extremum, or intersection is not visible in the current screen.

Message	Meaning (Continued)
Receive Error	Problem with data reception from another calculator. Re-send the data.
Too Few Arguments	The command requires more arguments than you supplied.
Undefined Name	The global variable named does not exist.
Undefined Result	The calculation has a mathematically undefined result (such as 0/0).
Out of Memory	You must recover a lot of memory to continue operation. Delete one or more matrices, lists, notes, or programs (using catalogs), or custom (not built-in) apps (using SHIFT *MEMORY*).

23

Appendix: Product Regulatory Information

Federal Communications Commission Notice

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio or television technician for help.

Modifications

The FCC requires the user to be notified that any changes or modifications made to this device that are not expressly approved by Hewlett-Packard Company may void the user's authority to operate the equipment.

Cables

Connections to this device must be made with shielded cables with metallic RFI/EMI connector hoods to maintain compliance with FCC rules and regulations. Applicable only for products with connectivity to PC/laptop.

Declaration of Conformity for products Marked with FCC Logo, United States Only

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

If you have questions about the product that are not related to this declaration, write to:
Hewlett-Packard Company
P.O. Box 692000, Mail Stop 530113
Houston, TX 77269-2000

For questions regarding this FCC declaration, write to:
Hewlett-Packard Company
P.O. Box 692000, Mail Stop 510101 Houston, TX 77269-2000 or call HP at 281-514-3333

To identify your product, refer to the part, series, or model number located on the product.

Canadian Notice
This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Avis Canadien
Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

European Union Regulatory Notice

Products bearing the CE marking comply with the following EU Directives:

- Low Voltage Directive 2006/95/EC
- EMC Directive 2004/108/EC
- Ecodesign Directive 2009/125/EC, where applicable

CE compliance of this product is valid if powered with the correct CE-marked AC adapter provided by HP.

Compliance with these directives implies conformity to applicable harmonized European standards (European Norms) that are listed in the EU Declaration of Conformity issued by HP for this product or product family and available (in English only) either within the product documentation or at the following web site: **www.hp.eu/certificates** (type the product number in the search field).

The compliance is indicated by one of the following conformity markings placed on the product:

For non-telecommunications products and for EU harmonized telecommunications products, such as Bluetooth® within power class below 10mW.

For EU non-harmonized telecommunications products (If applicable, a 4-digit notified body number is inserted between CE and !).

Please refer to the regulatory label provided on the product.

The point of contact for regulatory matters is:
Hewlett-Packard GmbH, Dept./MS: HQ-TRE, Herrenberger Strasse 140, 71034 Boeblingen, GERMANY.

Japanese Notice

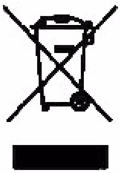
この装置は，クラスB情報技術装置です。この装置は，家庭環境で使用することを目的としていますが，この装置がラジオやテレビジョン受信機に近接して使用されると，受信障害を引き起こすことがあります。

取扱説明書に従って正しい取り扱いをして下さい。　VCCI－B

Korean Class Notice

B급 기기 (가정용 방송통신기기)	이 기기는 가정용(B급)으로 전자파적합등록을 한 기기로서 주로 가정에서 사용하는 것을 목적으로 하며, 모든 지역에서 사용할 수 있습니다.

Disposal of Waste Equipment by Users in Private Household in the European Union



This symbol on the product or on its packaging indicates that this product must not be disposed of with your other household waste. Instead, it is your responsibility to dispose of your waste equipment by handing it over to a designated collection point for the recycling of waste electrical and electronic equipment. The separate collection and recycling of your waste equipment at the time of disposal will help to conserve natural resources and ensure that it is recycled in a manner that protects human health and the environment. For more information about where you can drop off your waste equipment for recycling, please contact your local city office, your household waste disposal service or the shop where you purchased the product.

Chemical Substances

HP is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (*Regulation EC No 1907/2006 of the European Parliament and the Council).* A chemical information report for this product can be found at:

http://www.hp.com/go/reach

Perchlorate Material - special handling may apply
This calculator's Memory Backup battery may contain perchlorate and may require special handling when recycled or disposed in California.

产品中有毒有害物质或元素的名称及含量

根据中国《电子信息产品污染控制管理办法》

部件名称	有毒有害物质或元素					
	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr(VI))	多溴联苯 (PBB)	多溴二苯醚 (PBDE)
PCA	X	O	O	O	O	O
[illegible]	O	O	O	O	O	O

O：表示该有毒有害物质在该部件所有均质材料中的含量均在SJ/T 11363-2006
标准规定的限量要求以下。

X：表示该有毒有害物质至少在该部件的某一均质材料中的含量超出SJ/T 11363-2006
标准规定的限量要求。

表中标有"X" 的所有部件都符合欧盟RoHS法规

"欧洲议会和欧盟理事会2003年1月27日关于电子电器设备中限制使用某些有害物质的2002/95/EC
号指令"

注：环保使用期限的参考标识取决于产品正常工作的温度和湿度等条件

Index

I

K

L