HP-48GII/49G+ Quick Reference

General

Version	A list of contents can be found at the very end of this booklet.
	This documentation applies to software version "HP49-C Revision #1.23" (see VERSION command) and CAS version "4.20031005" (see VER command).
RPN and Algebraic mode	This manual exclusively deals with the RPN (Reverse Polish Notation) mode. See **Keyboard Shortcuts** or **Flags** on how to switch between these modes.
Machine reset	Under some circumstances the machine freezes and doesn't react on keyboard entries any more. In this case briefly remove the main batteries or see chapter **Keyboard Shortcuts** for different reset commands.
Display	HP-49G+: 131x64 pixel black&white LCD.
Contrast	See **Keyboard Shortcuts** for LCD contrast adjustment.
Key clicks	Can be turned on/off using the MODE menu, see there.
Settings	General settings can be selected thru the MODE menu, see **Menus**.
Precision	12 BCD digits, exponent ±499. No hidden digits.
Abbreviations	As used in this Quick Reference:

↗	↖	↑	↓	←	→
Shift-right	Shift-left	Up arrow	Down arrow	Left arrow	Right arrow

Basic Operation & Editing

The stack	In RPN mode all calculation takes place on the stack. • Ie. the "+" command removes the objects from stack level 1 & 2, adds them up and pushes the result back to stack level 1. • The stack can hold an unlimited number of objects (as memory permits). • Notably, the stack can be empty. Note that different from other HP calculators an empty stack level does not contain a numerical 0 value! Thus, executing the "+" command on a stack that doesn't at least contain two objects will cause an error. • Furthermore, there are commands that take arguments from the stack but do not return anything, ie. the "STO" ccommand.
Command entry	When pressing a number key in idle mode the display of the stack contents is shifted up to make room for the "edit line". Numbers, strings, arrays, programs etc. can be entered on the command line. Pressing ENTER in edit mode will check the input data for syntax errors, potentially evaluate the commands (see **Evaluation Rules**) and push the result onto stack level 1.
	Important: Some keys (ie. "+") will cause and implicit ENTER and immediately evaluate the command line. If such a character is to be entered (ie. into a string) without causing an implicit ENTER then it

	must be entered in ALPHA mode, see further down.		
	To enter special characters use the CHARS menu, see there.		
Editing	Edit commands:		
	Arrow keys	Move the cursor.	
	↗ Arrow	Move to the top/bottom of the edited text or start/end of the edited line.	
	↖ Arrow	Same as above but when moving to the top or bottom of the text the current column position is preserved.	
	DEL	Delete character under the cursor.	
	⇦	Delete the character to the left.	
	COPY, CUT	Copy or cut selected text to the keyboard.	
	PASTE	Insert text from the clipboard.	
	BEGIN, END	Set start & end point of selected text.	
	↵	Enters a line feed	
	CLEAR, ON	Aborts edit mode	
	In edit mode the EDIT menu softkeys are displayed. See chapter **Menus** for details on softkeys. They can also be activated pressing the TOOL key:		
	←SKIP	Skip to the beginning of the previous word	
	SKIP→	Skip to the beginning of the next word	
	←DEL	Delete left word	
	DEL→	Delete right word	
	DEL L	Delete entire line	
	INS	Toggle insert/overwrite mode	
	SEARC	Displays the SEARCH menu which allows to find and optionally replace text. Note that the search does not wrap around. SEARCH menu commands are:	
		FIND	Display the find screen for case sensitive or insensitive forward search.
		REPL	Display the find & replace screen. Note that found text is not automatically replaced!
		NEXT	Find next occurence of specified text. Does not wrap around!
		R	Replace selection with replace-text.
		R/N	Replace selection with replace-text and find next.
		ALL	Find & replace all occurences.
		EDIT	Go back to EDIT menu.
	GOTO	Displays the GOTO menu which allows to place the cursor:	
		GOTOL	Go to a line by number.
		GOTOP	Go to a position by character offset.
		LABEL	Performs no action.
		EDIT	Go back to EDIT menu.
	EDIT	Activates the edit mode for the currently selected text and replaces the selection with the edit result.	
	→BEG	Jump to the beginning of the selected text or to the beginning of the edited text if nothing is selected.	
	→END	Jump to the end of the selected text or to the beginning of the edited text if nothing is selected.	

	STACK	Display STACK menu
	EXEC	Evaluates the current selection and replaces the selection with the result.
	HALT	Aborts the current editing process, returns to idle mode and activates the HLT symbol above the LCD. To resume the previousely halted edit session enter CONT. Multiple editing sessions can be halted. CONT will resume the most recently halted one.
	Style	Display the STYLE menu which allows to select bold, italic, underlined and inverted text as well as different fonts. However, some formatting options (ie. inverted) don't work at all and other modifications (ie. underlined, italic) are lost as soon as the editor is exited.
	INFO	Display command line info and various system infos
	TOOLS	Displays an empty softkey menu.

ALPHA mode

To enter individual characters rather then entire commands press the ALPHA key.

- The characters are printed in yellow on the key tops.
- Usually, pressing ALPHA twice locks the ALPHA mode so that multiple chars can be entered. ALPHA-lock mode can be controlled by flag 60, see **Flags**.
- Keys that don't have ALPHA labels (ie. the number and basic math keys) will create the normal symbols when pressed in ALPHA mode (ie. digits and math symbols).
- SPC and ENTER perform their normal opereration in ALPHA mode.
- ALPHA mode is indicated by small "α" symbol on top of the display.

In ALPHA mode the following shifted modes are available:

↗	COPY, CUT	Copy or cut selected text to clipboard. If no text is selected all of the currently edited text is cut.
	PASTE	Paste clipboard to current cursor position
	BEGIN, END	Start and end point of text selection
	CLEAR	Enter "CLEAR" string – which really doesn't make much sense.
	ALPHA	Toggles insert/overwrite mode
	↵	Enters a line feed
	7	Puts an accent on characters like a, e, to produce á, é.
	8	Puts an accent on characters like a, n to produce ã, ñ.
	9	Puts an accent on characters like a, e to produce å, æ.

Other:

A **α**	B **β**	C **Δ**	D **δ**	E **ε**	F **ρ**	I **|**	M **μ**	N **λ**	O **'**
P **Π**	Q **^**	R **√**	S **σ**	T **θ**	U **τ**	V **ω**	W **=**	X **<**	Y **>**
Z **/**	x **"**	4 **€**	5 ****	6 **∠**	- **_**	1 **~**	2 **!**	3 **?**	+ ◀◀ ▶▶
0 **→**	SPC **,**	ENT **@**							

↖	character	Enters the character in lower case

		DEL	Delete character under cursor										
		ALPHA	Toggles upper/lower case ALPHA mode. Note that this setting is preserved even when ALPHA mode is exited.										
		7	Puts an accent on characters like a, e, to produce à, è.										
		8	Puts an accent on characters like a, e, to produce â, ê.										
		9	Puts an accent on characters like a, e, to produce ä, ë.										
		Other:											
		x	4	5	6	-	1	2	3	+	0	.	SPC	ENTER
		[]	**$**	**£**	**§**	**()**	**%**	**;**	**#**	**{}**	**∞**	**:**	**π**	**&**
	↗ hold	Some heys have a special meaning:												
		O	6	2	3	SPC								
		Ω	**°**	**¡**	**¿**	**;**								

Memory & Display

Memory	

Variables

General	Data objects can exist on the stack or be stored in variables. Variables are created in the current directory, see **Directories**. There are multiple memory spaces, among them the optional SD flash card, see FILES menu in section **Menus** and further down.
Storing	To store the object on stack level 1 in a named variable enter an algebraic object containing a name and press the STO> key. Ie. `17 'A' STO` first pushes 17 and 'A' onto the stack, then takes both arguments from the stack and stores the value 17 into the variable named A. Variables can also be created using the FILE menu, see there.
	Special naming convention exist to store data in individual elements of a Vector, Matrix or List: `33 'M(1,1)' STO` stores the value 66 in the element (1,1) of Matrix M. For Vectors and Lists only one index may be given. This does not work for Strings. Note that the comma between the column and row index cannot be replaced by a space because this would create an ambiguity with complex numbers. It doesn't seem to be possible to store a complex number into a real matrix.
Recalling & Variable menu	Press the VAR key to display the variables and subdirectories of the current directory in a softmenu. Press a variable softkey to evaluate the contents of the variable, see **Evaluation Rules**. In general: • For numerical variable contents this will simply return the numerical value. • Programs will be executed. • Algebraic objects will be evaluated. • A softkey which refers to a subdirectory will change the current directory to this subdirectory.
	A special syntax exists to retrieve an element of a List, Matrix or Vector: `'M(1,1)' EVAL` retrieves element (1,1) from matrix M. Note that `'M(1,1)' RCL` causes an error. Also, `M(1,1)` returns the matrix and the complex number (1,1) on the stack.

	To recall the contents of a variable without evaluating it enter the variable name and press the RCL key.
	Variables can be moved or copied to other directories using the FILES menu, see there.
	Newly created variables are always displayed as the first entry in the VAR softmenu. To reorder the sequence of variables in the softmenu the VARS and ORDER commands can be used.
Memory spaces	By default there are memory spaces 0:IRAM, 1:ERAM, 2:FLASH and HOME. A SD-flash card would be 3:SD, see menu FILES in section **Menus**. Variables are normally stored in HOME but can also exist in other memory spaces: • `125 :1:V STO` stores 125 in variable V in ERAM. • `:1:V RCL` recalls this variable from ERAM. Notes: • Do not put the tagged name in single quotes since this would cause an error. • Memory spaces other than HOME might not support subdirectories.
Shortcuts	• Press Shift-right variable-softkey to recall a variable without evaluating it. • Press Shift-left variable-softkey to store the contents of stack level 1 in this variable. • Press Shift-right down-arrow to display a list of the variables of the current directory and their contents. • When the edit line is in algebraic mode after quotes haven been entered then pressing a variable's associated softkey inserts the variable name.
Editing variables	Programs stored in variables frequently need to be edited. The quickest way to do this: • Press Shift-right variable-softkey to put the program (=the contents of the variable) in stack level 1. • Press down-arrow to edit the contents of stack level 1. • Press ENTER to return the modified program to stack level 1. • Press Shift-left variable-softkey to move the modified program from stack level 1 back into the variable.
Variable naming	• Variable names may be really long although that doesn't make much sense because ambiguity arises when the names are displayed in the VAR softmenu. • Variable names are case sensitive. • Since variable names are entered as algebraic objects there must not be any ambiguity with algebraic expressions. Ie. 'A+' is not a valid variable name but '→POL' is. • In general variable names must start with a (possibly Greec) character but some symbols are allowed. • Variable names must never start with a number.
Local variables	These are most often needed in program to avoid conflicts with existing global variable names. See "→" command.
Deleting	Put the name of the variable or a list of variable names on the stack and execute the PURGE command. If the variable refers to a subdirectory it can only be removed if it is empty.
Variable	Best done thru the FILES menu, see **Menus**.

renaming	
Search path	If a variable cannot be found in the current directory it is searched in the parent directories. This only applies for evaluating or recalling a variable: • Nothing happens if a variable is purged that is located in a parent directory. • A value is never stored in a variable that exists in a parent directory. Rather a new variable in the current directory is created.
Varaibles with a special meaning:	
ΣDAT	Statistics data matrix. The number of rows correspond to the number of data samples and the number of columns to the number of independent variables within each data sample.
ΣPAR	Statistics parameter list. This variable is needed by a number of statistics functions, especially curve fitting. List contents are: • Column within ΣDAT which contains the X-values (default=1). • Column within ΣDAT which contains the Y-values (default=2). • Y-offset of the most recent curve fit. • Slope of the most recent curve fit. • Model of the most recent curve fit: BESTFIT, EXPFIT, LINFIT (default), LOGFIT, PWRFIT. If ΣPAR doesn't exist it is created with default values as indicated above.
PPAR	A variable holding a number of plot parameters used by various plot functions. Its contents are: • A Complex number specifying the bottom-left coordinate of the plot area. • A Complex number specifying the top-right coordinate of the plot area. • Name of the independent variable, mostly X. • Increment for the independent variable. If 0 then the increment is chosen so that it corresponds to a display pixel. • A Complex number specifying the center of the axes. Alternatively, instead of a single Complex number a List containing: • A Complex number specifying the center of the axes. • A list with the tick mark distances in absolute values (Real number) or in pixel (binary number). • The type of plot, ie. FUNCTION. • The name of y-axis, usually Y.
IOPAR	A List controlling the serial port containing. • The baudrate in bits/sec. • The other values are unknown.
VX	Located in the CASDIR subdirectory which is created whenever algebraic or infinitesimal calculations are performed. VX contains the name of the independent variable for algebraic transformations. Usually, it should be set to `'X'`.
EQ	????

Keyboard Shortcuts

General	The following key commands apply only in idle mode but not when in edit mode or when a special entry form is active:
↗-hold ENTER	Toggle between exact and approximate mode. See **Flags** or **Menus**.

↓	Edit object in stack level 1.
→	Swap contents of stack level 1 and 2. Same as the SWAP command.
↑	Access stack history. Same as HIST key.
←	Start picture view: Display last graph of picture.
↗ ↓	Display full names of softkey labels. In the VAR menu where the softkeys refer to variables/directories the contents of the variables is displayed as well. Any keypress returns to normal view.
↗ →	Start X-modem server.
↗-hold →	Start Kermit server.
↖-hold TOOL	Toggle between real and complex mode.
↗ Softkey	In VARS menu: Recall variable w/o evaluating it.
↖ Softkey	In VARS menu: Store data in variable (RPN only).
↖-hold PREV	Jumps to previous menu.
↖-hold UPDIR	Change directory to HOME. Note that when another memory space has been selected thru the LIB menu the directory change will onl be visible after pressing VAR.
↗-hold 7	SOLVE menu.
↖-hold MODE	Menu PRG/MODES.
↗-hold CHARS	Menu PRG/CHAR.
+/–	Inside the MODE menu this can be used to check/uncheck flags. Does not work in the FLAGS menu.
ON and +	Increase contrast
ON and –	Decrease contrast
ON – F1 – F6	Cold restart – all memory contents will be lost! Press & hold ON, then press F1 and F6 briefly.
ON – F2	Undo the most recent keypress
ON – F3	Warm boot, memory is not lost. If the unit freezes this won't help in most cases. Try briefly removing one of the main batteries.
ON – F4	Start interactive self-test.
ON – F5	Start continuous self-test.
ON –SPC	Quiet mode, timers off.
ON – F1	Create a screenshot.
ON – F4	Cancel the next alarm.

Data Types

General	The calculator knows a variety of built-in data types. Operations are usually defined for multiple types, ie. the "+" operator can add real or complex numbers, vectors, matrices, lists etc.
	Each data type has an associated type number which can be determined using the TYPE command.

Real number	Example: `1.234E45` Bytes: 10.5, Type 0
Complex number	Example: `(1.3 4.)`. Note that the deliminator (period or comma depending on the fraction separator setting) must be used to separate the real and imaginary part when a complex number is entered as a part of an expression: `'1+(1.3,4.)'`. Alternatively, input is possibly using the symbol i: `'1+2*i'` returns `(1 2)` after an EVAL. Bytes: 18.5, Type 1
String	Example: `"String!"`. A string may be empty. Bytes: 5 + number of characters , Type 2
Real vector or matrix	Example: `[1. 2. 3.]` or `[[1. 2.][3. 4.]]` When entering a vector or matrix using the matrix writer (MTRW) then mixed real and complex elements are supported but yield a type 29 object. When entering the value using the command line and [] brackets and one of the elements is complex all elements will be converted to complex yielding a type 4 object. Type 3
Complex vector or matrix	Example: `[1. 2. 3.]` or `[(1. 2.) (3. 4.)]` or [[1. 2.][3. 4.]]. See type 29. Type 4
List	Example: `{"123" 3.45 (7 8) [1 2 3 4]}` The list may contain a mixture of other data types. Type 5
Global name	Example: `'X'` or `'δ0'` Names can refer to variables or subdirectories. Type 6
Local name	Example: ◀◀ → X ◀◀ 5 X + LN ▶▶▶▶ Here, X is a local name which does not interfere with any global variable named X. Local names can only exist while a program executes. Type 7
Program	Example: ◀◀ 3 * 17 + ▶▶ Type 8
Algebraic expression	Example: `1+2` or `A+B` Note that fractions using exact numbers and constants are treated as algebraic expressions. Type 9

Binary number	Example: `#17h` For more details see the BASE menu in section **Menus**. Type 10, Bytes: 13
Graphics object	A graphics object (GROB) created with the plot functions or LCD→. Type 11
Tagged object	Example: `:Result:125.` Tagged objects are often used to describe results that are returned to the stack. Normal calculations can be performed on tagged objects but this causes their tag to be lost. Type 12
Real number with unit	Example: 125_mm Note that only real numbers can have an associated unit. Type 13
Type 14	*unknown* Type 15
Directories	From the FILES menu it is possible to recall entire directories to the stack using the RCL softkey.
Libraries	Type 16
	unknown Type 17
Built-in functions	Ie. the "+" operator. It is pretty hard to create an object of this type on the stack. Try `'4+5' OBJ→`. If such an object is evaluated it performs its natural operation. Type 18
Built-in commands	NOVAL is an example for this type of object. Type 19
	unknown Types 20-27
Exact value	Example: `45` Exact values can be activated in the MODE/CAS menu. Bytes: 6.5, Type 28
Mixed real/complex or exact vector or matrix	Example: `[1. (2. 3.)]` It seems that any "odd" kind of matrix with mixed real and complex values gets type 29. Type 29

Menus

General	Menus are displayed in two different ways, depending on flag 117: Flag 117 set: Display as a soft menu. This is a row of keys at the bottom of the display which can be activated by pressing F1…F6. Flag 117 clear: Display as choose boxes. Use (optionally shifted) up/down arrow keys or number keys to select an entry and press ENTER or right arrow to activate the selected entry. Note that flag 117 does not affect all menus. Ie. the APPS menu is always displayed as a choose box and the TOOL menu is always displayed as a soft menu.

	Note that menus may be presented in different ways depending on whether edit mode is active or not. Ie. in idle mode "↗ TIME" pops up a choose box as described below. But in edit mode the TIME/Tools softmenu is displayed.
	Use NXT and PREV keys to display the next/previous set of soft labels. Note that there is no way to hide the soft labels. See also **Keyboard Shortcuts**.
	Soft menu keys show a small horizontal line on the top left if they refer to a submenu. This does not apply to submenus in the CAS menu (?). Submenus have as their last entry always a reference back to the previous menu level. This can be bit odd though: Ie. the BASE menu can be activated directly from the keyboard key "3" but still there is a MTH entry which activates the math menu…
	Below, menus are described in the order as they appear on the keyboard, from top left to bottom right. Menus that are not accessible thru the keyboard (PLOT, I/O, Constants) are covered at the end of this section. Note that different from ie. the HP-28S the keyboard printing does not indicate whether a key function corresponds to a menu.
	For more details on the commands mentioned below see section **Commands**.
Y=	Accessed via ↖-hold F1. Plot Function form: ??
WIN	Accessed via ↖-hold F2. Plot Window – Function form: ??
2D/3D	Accessed via ↖-hold F4. Plot Setup form: ??
TBLSET	Accessed via ↖-hold F5. Table Setup form: ??
APPS	Displays a choose box to access various other menus:

No.	*Menu*	*Action*	*Keyboard*
1.	Plot functions…	Displays PLOT commands choose box	
2.	I/O functions…	Displays file transfer and print commands choose box	
3.	Constants lib…	Displays list of built-in constants	
4.	Numeric Solver…	Displays NUM.SLV choose box	↗ 7
5.	Time & Date…	Displays TIME functions choose box	↗ 9
6.	Equation writer	Starts equation writer EQW	↗ '
7.	File manager	Displays FILES menu	↖ APPS
8.	Matrix writer	Starts the matrix writer	↖ '
9.	Text editor	Starts the text editor on the entry line	
10.	Math menu	Displays Math menu	↖ SYMB
11.	CAS menu	Displays CAS menu	

FILES	File management. Displays the root structure of the file system. On the HP-49g+ this contains 0:IRAM, 1:ERAM, 2:FLASH, 3:SD (optional) and HOME. • Use up and down arrow keys to navigate, optionally with shift-right. • Use the right arrow key or the CHDIR softmenu to change into one of the directories. This displays the file browser. • See section **Variables**: Memory spaces and menu LIB in section **Menus**.

File browser:

- Use up/down arrow keys to navigate.
- Use right arrow to change to a subdirectory.
- Use left arrow or ↖ UPDIR to change to the parent directory.
- On the right side the variable type and size is displayed.
- For subdirectories the size of its contents plus the directory overhead is given.
- Press ENTER to select/unselect an entry. The number of selected bytes is indicated in the header line. Some of the commands below apply to all the selected objects, ie. PURGE.

Softkeys:

EDIT	Edit a variable or entire directory (!)
COPY	Copy a variable or directory to another location/directory.
MOVE	Move a variable or directory to another location/directory.
RCL	Recall contents of a variable to stack level 1. It is possible to recall entire directories.
EVAL	Evaluate variable. Performs no action on directories.
TREE	Return to the root structure.
PURGE	Delete a variable or directory. Confirmation is controlled by flag 76, see **Flags**.
RENAM	Rename variable or directory.
NEW	Create a new object in the selected directory. Can be a variable or directory.
ORDER	When pressed the currently highlighted variable or directory will be moved to the top of the list. This affects the order in the VAR menu.
SEND	Send object over the USB or infrared port.
RECV	Receive object over the USB or infrared port.
HALT	HALT the current file menu session, activate the HLT symbol above the LCD display and return to normal mode. Use CONT to resume the halted session. Multiple sessions can be halted.
VIEW	View the object.
EDITB	Same as EDIT ??
HEADE	Toggle file browser header between Memory/Select and object count/directory display.
LIST	Toggle variable type and length display on/off.
SORT	Displays a choose box to select various sort methods. This affects the file browser display only!
	Original: Original order as displayed in the VAR menu.
	Type: Alphabetically be type name
	Name: Alphabetically be variable/directory name
	Size: By size
	Inv. Type: Reverse alphabetically be type name
	Inv. Name: Reverse alphabetically be variable/directory name
XSEND	Send object over the USB or infrared port using the Xmodem protocol.
CHDIR	Changes to the selected subdirectory of HOME and quits the FILES menu. It is not possible to change the current directory to

		one of the other memory spaces. Use the LIB menu to do this. Note that when another memory space has been selected thru the LIB menu the directory change will onl be visible after pressing VAR.
MODE	Selects various operation modes.	
	CALCULATOR MODES:	
	Softkeys:	
	FLAGS	Displays a list of system flags for modification, see **FLAGS**.
	CHK, CHOOS	Depending on the setting. CHK enables/disables a feature, CHOOS pops up a choose box to choose a setting from a list. The "+/-" key can be used insted of CHK.
	CAS	Displays the CAS settings screen, see further down.
	DISP	Displays the display settings screen, see further down.
	CANCEL	Discard changes, same as presing ON.
	OK	Accept changes, same as pressing ENTER.
	RESET	Pops up a choose box used to reset the selected or all settings to their defaults.
	Settings:	
	Operating Mode	RPN (Reverse Polish Notation) or algebraic
	Number Format	Number display format: Standard: Normal display with trailing-0 suppresion and automatic switching to exponential notation. Fixed: Fixed-point notation. An additional field allows to choose the number a digits (0-11). Scientific: Exponential notation. Engineering: Exponential notation where the exponent is always a multiple of 3.
	FM	Fraction mark selection, default is a period. When checked a comma is used as the fraction mark. Note that the respective other symbol is used as a delimiter.
	Angle Measure	Degrees (360°), Radians (2π), Grads (400)
	Coord System	Coordinate system for displaying complex numbers and 2- or 3-dimensional vectors. • Rectangular (x,y): For all types. • Polar (r,θ): For complex numbers and 2-dim vectors • Spherical (r,θ, φ): For 3-dim vectors It is important to note that this setting only affects the display format but not the internal represenatation of the data!
	Beep	When unchecked all sounds are suppressed, including alarms and key clicks.
	Key Click	Enables keyboard clicks
	Last Stack	Enables the "last stack" which is needed for the UNDO and ANS command. Can be quite memory consuming.
	CAS MODES:	
	Softkeys:	

	Edit	Different from the CHK/CHOOSE method of the CALCULATOR MODES menu the EDIT key is used to modify a value in the CAS MODES menu. Flags must be set to "0" or "1" but the "+/-" key still works.	
	CANCEL	Discard changes, same as presing ON.	
	OK	Accept changes, same as pressing ENTER.	
	RESET	Pops up a choose box used to reset the selected or all settings to their defaults.	
	CALC	????	
	TYPES	????	
	Settings:		
	Indep Var	????	
	Modulo	????	
	Numeric	????	
	Approx	????	
	Complex	????	
	Verbose	????	
	Step/Step	????	
	Incr Pow	????	
	Rigorous	????	
	Simp Non-Rational	????	
	DISPLAY MODES:		
	Font	Choose the font for the stack display	
	Edit: Small	Uses a small font on the edit line when entering data	
	Edit: Full Page	?? doesn't do anything	
	Edit: Indent	?? doesn't do anything	
	Stack: Small	Display main stack in a small font, except stack level 1.	
	Stack: Textbook	When active formulas and matrices on the stack are displayed in "textbook" mode.	
	EQW: Small	When active a small font is used in the equation writer by default.	
	EQW: Small Stack Disp	When active equations in textbook mode on the stack are displayed using the small font. However, all values on stack level 1 are displayed in a small font (??)	
	Header	Select the number of header lines on top of the display. Note that even with "no header" selected the space is not used for stack display! 0: No header 1: Directory + real/complex + approximate mode 2: Full information	
	Clock	When selected and two header lines are visible a clock is displayed.	
	Analog	When selected an analog clock is selected (pretty hard to read).	
CUSTOM	User-defined menu. Using the ???? command it is possible to create a specialized menu.		

	This is always displayed as a softmenu rather than a choose box.		
TOOL	Frequently used commands or edit commands in edit mode:		
	EDIT	Edit the object in stack level 1. See chapter **Basic Operation & Editing**. Special editors are available for equations (see menu EQW) and matrices (see manu MTRW). Other objects are edited using a simple full-screen text editor.	
	VIEW	View the object in text or graphics mode.	
	STACK	Display the STACK menu.	
	RCL	Recall the variable who's name is given in stack level 1.	
	PURGE	Delete the variable who's name is given in stack level 1.	
	CLEAR	Clear the stack.	
	CASCM	????	
	HELP	????	
VAR	Variables of the current directory.		
HIST	Stack editing		
CMD	Display a list of previousely entered commands.		
PRG	Programming commands, organized in these subdirectories:		
	STACK	Stack manipulation: DUP, SWAP, DROP, OVER, ROT, UNROT, ROLL, PICK, UNPICK, PICK3, DEPTH, DUP2, DUPN, DROP2, DROPN, DUPDUP, NIP, NDUPN	
	MEM	Memory and file related: PURGE, MEM, BYTES, NEWOB, *DIR, ARITH,* ARCHIVE, RESTORE. Submenus:	
		DIR	Directory manipulations: PURGE, RCL, STO, PATH, CRDIR, PGDIR, VARS, TVARS, ORDER
		ARITH	Variable arithmetic: STO+, STO–, STO*, STO/, INCR, DECR, SINV, SNEG, SCONJ
	BRCH	Branch instructions: *IF, CASE, START, FOR, DO, WHILE*, IFT, IFTE. Submenus:	
		IF	IF, THEN, ELSE, END
		CASE	CASE, THEN, END
		START	START, NEXT, STEP
		FOR	FOR, NEXT, STEP
		DO	DO, UNTIL, END
		WHILE	WHILE, REPEAT, END
	TEST	Comparisn and flag testing: ==, ≠, <, >, ≤, ≥, AND, OR, XOR, NOT, SAME, TYPE, SF, CF, FS?, FC?, FS?C, FC?C, LININ	
	TYPE	Type conversions: OBJ→, →ARRY, →LIST, →STR, →TAG, →UNIT, C→R, R→C, NUM, CHR, DTAG, EQ→, TYPE, VTYPE	
	LIST	List manipulation: *ELEM, PROC*, OBJ→, →LIST, SUB, REPL Submenus:	
		ELEM	Set & get elements of a list: GET, GETI, PUT, PUTI, SIZE, POS, HEAD, TAIL
		PROC	Process a list: DOLIST, DOSUBS, NSUB, ENDSUB, STREAM, REVLIST, SORT, SEQ
	GROB	Graphic-object manipulation: →GROB, BLANK, GOR, GXOR, SUB, REPL, →LCD, LCD→, SIZE, ANIMATE	
	PICT	Drawing commands: PICT, PDIM, LINE, TLINE, BOX, ARC, PIXON, PIXOFF, PIX?, PVIEW, PX→C, C→PX	

	CHARS	String functions: SUB, REPL, POS, SIZE, NUM, CHR, OBJ→, →STR, HEAD, TAIL, SREPL	
	MODES	Various system settings. Submenus:	
		FMT	Number formatting options: STD, FIX, SCI, END, FM,, ML
		ANGLE	Angular settings: DEG, RAD, GRAD, RECT, CYLIN, SPHERE
		FLAG	Flag operations: SF, CF, FS?, FC?, FS?C, FC?C, STOF, RCLF, RESET
		KEYS	Keyboard operations: ASN, STOKEYS, RCLKEYS, DELKEYS
		MENU	Menu manipulations: MENU, CST, TMENU, RCLMENU
		MISC	Various system settings: BEEP, CLK, SYM, STK, ARG, CMD, INFO
	IN	User input functions: INFORM, NOVAL, CHOOSE, INPUT, KEY, WAIT, PROMPT	
	OUT	Display functions: PVIEW, TEXT, CLLCD, DISP, FREEZE, MSGBOX, BEEP	
	TIME	Time and date functions: DATE, →DATE, TIME, →TIME, TICKS, ALRM, DATE+, DDAYS, →HMS, HMS→, HMS+, HMS-, TSTR, CLKADJ	
	ERROR	Error handling: DOERR, ERRN, ERRM, ERR0, LASTARG, IFERR	
	RUN	Program control and debugging: DBUG, SST, SST↓, NEXT, HALT, KILL, OFF	
CHARS	Displays a list of available characters. Use the arrow keys (optionally with the Shift-right key) to move the cursor to the desired character. On the bottom row left there's an indication how the character can be entered from the keyboard. Also, the ASCII code is displayed. Soft buttons:		
	MODIF	Enters the character editor which allows to modify the appearance of the character. Note: When in edit mode the MODIF softkey is not available. Use the following keys and softkeys for editing: • "." to turn off/on the pixel under the pixel-cursor. • Arrow keys or shift-arrow keys to move the pixel cursor. • SCAN to return to the list of characters. • CHR– to display the previous character. • CHR+ to display the next character. Modifications are automatically preserved.	
	ECHO1	Echos the selected character to the command line and quits the CHARS menu.	
	ECHO	Echos the selected character to the command line but does not exit the menu.	
MTRW	Activates the matrix writer. This is a full-screen editor which allows to enter Matrices (or Lists, see below) in an intuitive way.		

<table>
<tr><td></td><td colspan="2">

When activating the editor directly via ↖ MTRW it will always start with an empty matrix.
To put the matrix in stack level 1 into the matrix writer for editing press the down-arrow key or press EDIT in the TOOL menu.
The current size of the matrix is displayed in the top-left corner of the matrix writer form.
The current matrix position and its value is displayed in the bottom line of the matrix writer form.
When flag 91 is set the matrix writer does not create Matrices but Lists containing Lists. Note that with normal means it is very difficult to create a list containing other lists. Also, an exisiting list cannot be put back into the matrix writer for editing.

General operations:
<table>
<tr><td>Arrow keys</td><td>Move around the highlighted cursor which indicates the currently selected cell.</td></tr>
<tr><td>Data entry</td><td>Simply start typing to enter a value into the currently selected cell. Empty cells will be filled with 0 if necessary.</td></tr>
<tr><td>↖ Arrow →</td><td>View previous set of columns.</td></tr>
<tr><td>↖ Arrow ←</td><td>View next set of columns.</td></tr>
<tr><td>↖ Arrow ↑</td><td>View previous set of rows.</td></tr>
<tr><td>↖ Arrow ↓</td><td>View next set of rows.</td></tr>
<tr><td>↗ Arrow →</td><td>Jump to the rightmost column.</td></tr>
<tr><td>↗ Arrow ←</td><td>Jump to column 1.</td></tr>
<tr><td>↗ Arrow ↑</td><td>Jump to the bottommost row.</td></tr>
<tr><td>↗ Arrow ↓</td><td>Jump to row 1.</td></tr>
<tr><td>ENTER</td><td>Accept changes and quit editor.</td></tr>
<tr><td>ON</td><td>Discard changes and quit editor.</td></tr>
<tr><td colspan="2">Softkeys:</td></tr>
<tr><td>EDIT</td><td>Bring the contents of the selected cell to the edit line for modification.</td></tr>
<tr><td>VEC</td><td>Normally, a matrix with one row of data is still a matrix, ie. <code>[[3 4 5]]</code>. But when the "VEC" option is activated this will create a row Vector: <code>[3 4 5]</code>.
It is not possible to create a column vector.</td></tr>
<tr><td>←WID</td><td>Make columns display smaller.</td></tr>
<tr><td>→</td><td>Make columns display wider.</td></tr>
<tr><td>GO→</td><td>After number input jump to the next column.</td></tr>
<tr><td>GO↓</td><td>After number input jump to the next row.</td></tr>
<tr><td>+ROW</td><td>Insert a row above the current row.</td></tr>
<tr><td>–ROW</td><td>Delete current row.</td></tr>
<tr><td>+COL</td><td>Insert a column to the left of the current column.</td></tr>
<tr><td>–COL</td><td>Delete current column.</td></tr>
<tr><td>→STK</td><td>Copy the current matrix to the stack without leaving the matrix editor.</td></tr>
<tr><td>GOTO</td><td>Pops up a form to specify a row and column number and then jumps to this matrix element.</td></tr>
<tr><td>DEL</td><td>Set current cell contents to 0.</td></tr>
</table>
</td></tr>
<tr><td>EQW</td><td colspan="2">Activates the equation writer.</td></tr>
</table>

	This is a full-screen editor which allows to enter complicated formuals in an intuitive way. • Left/Right arrow moves the highlighted selection one expression left/right. • Down arrow narrows downs the selection to the next subexpression. • Up arrow enlarges the selection. • The UNDO operation is supported. Softkeys:		
	EDIT	Bring the currently highlighted expression to the line editor for manipulation.	
	CURS	????	
	BIG	Toggles font between normal and small.	
	EVAL	Evaluates the highlighted expression and replaces it with the result.	
	FACTO	????	
	SIMP	????	
	CMDS	????	
	HELP	????	
SYMB	Activates a menu which allows access to a number of submenus related to symbolic manipulations of functions (CAS). Submenus:		
	ALG	EXPAND, FACTOR, LIN, SUBST, TEXPAND, SYMB	
	ARITH	DIVIS, IEGCD, IQUOT, ISPRIME?, PROPRAC, IREMAINDER	
	CALC	DERIV, DERVX, IBP, INTVX, lim, SERIES,TAYLOR0	
	GRAPH	DEFINE, GROBADD, PLOT, PLOTADD, 2D/3D, SIGNTAB, TABVAL, TABVAR	
	SOLVE	LDEC, LINSOLVE, rref, SOLVEVX, SOLVE, ZEROS	
	TRIG	HALFTAN, TAN2SC, TAN2SC2, TEXPAND, TLIN, TRIG	
	EXPLN	EXPLN, LIN, LNCOLLECT, SINCOS, TEXPAND, SYMB	
MTH	Gives access to a number of submenus related to regular math functions:		
	VECTR	Vector operations: ABS, DOT, CROSS, V→, →V2, →V3, RECT, CYLIN, SPHERE	
	MATRX	Matrix operations: *MAKE, NORM, FACTR, COL, ROW,* LSQ, RSD, EGV, EGVL, →DIAG, DIAG→. Submenus:	
		MAKE	Utilities to create & convert matrices: CON, IDN, TRN, RDM, RANM, SIZE, GET, GETI, PUT, PUTI, SUB, REPL, →DIAG, DIAG→, VANDERMODE, HILBERT
		NORM	Various matrix conditions: ABS, SNRM, RNRM, CNRM, SRAD, COND, RANK, DET, TRACE, TRAN
		FACTR	Matrix decomposition: RREF, LU, LQ, QR, SCHUR, SVD, SVL
		COL	Column operations: →COL, COL→, COL+, COL–, CSWP
		ROW	Row operations: →ROW, ROW→, ROW+, ROW–, RCI, RCIJ
	LIST	List manipulation: ΔLIST, ΣLIST, ΠLIST, SORT, REVLIST, ADD	
	HYP	Hyperbolic trigonometric functions: SINH, ASINH, COSH, ACOSH, TANH, ATANH, EXPM, LNP1.	

	REAL	Various real number functions: %, %CH, %T, MIN, MAX, MOD, ABS, SIGN, MANT, XPON, IP, FP, RND, TRNC, FLOOR, CEIL, D→R, R→D	
	BASE	Binary number manipulation, same as ↗ 3: HEX, DEC, OCT, BIN, R→B, B→R, *LOGIC, BIT, BYTE*, STWS, RCWS. Submenus:	
		LOGIC	Logic functions: AND, OR, XOR, NOT
		BIT	Bit-wise rotation: RL, SL, ASR, SR, RR
		BYTE	Byte-wise rotation: RLB, SLB, SRB, RRB
	PROB	Probability functions: COMB, PERM, !, RAND, RDZ, UTPC, UTPF, UTPN, UTPT, NDIST	
	FFT	Fourier transforms: FFT, IFFT	
	CMPLX	Complex number functions: RE, IM, C→R, R→C, ABS, ARG, SIGN, NEG, CONJ	
	CONST	Built-in constants (not to be confused with units): e (symbolic), 2.718 (e as a number), i (symbolic), (0 1), π (symbolic), 3.141 (π as a number), MINR (symbolic), 1E-4 (MINR as a number), MAXR (symbolic) 9.999 (MAXR As number)	
		Label	*Results in*
		`e`	`'e'`
		`2.718`	`2.71828…`
		`i`	`'i'` (Complex mode is turned on)
		`(0 1)`	`(0. 1.)`
		π	'π'
		`3.141`	`3.1412…`
		`MINR`	`'MINR'`
		`1.E-4`	`1.E-499`
		`MAXR`	`'MAXR'`
		`9.999`	`9.9999…E+499`
	SPECI	Special functions related to the Gamma function: GAMMA, PSI, Psi	
CAT	Displays a scroll box with all implemented 762 commands. Press a character or symbol key to quickly jump to the first name that matches this character. Use shift-left up/down to page up/down. Use shift-right up/down to jump to the top/bottom of the list. Press ENTER to execute the selected command.		
S.SLV	Symbolic solver menu		
NUM.SLV	Numerical solver choose box		
EXP&LN	Exponential and logarithmic functions menu		
TRIG	Trigonometric functions		
FINANCE	Financial problems solver form		
TIME	Time functions and alarms management choose box. Note that in edit mode the time tools softmenu is displayed rather than the choose box.		
	Browse alarm… Allows to create, delete and edit alarms. The number of alarms is apparently not limited. When an alarm occurs the following happens:		

	• The number of the alarm is pushed onto the stack (1-10) as a Real. • The expression in the alarm's "Message" is evaluated. This can for example be a program that BEEPs and displays a message.	
	Set alarm…	Directly displays the "set alarm" entry form.
	Set time, date…	Displays the "Set time and date" entry form.
	Tools…	Displays the time tools softmenu.
CALC	Calculus menu.	
	Submenu	*Commands*
	DERIV	CURL, DERIV, DERVX, DIV, FOURIER, HESS, IBP, INTVX, LAPL, PREVAL, RISCH, SIGMA, SIGMAVX
	LIMIT	DIVPC, lim, SERIES, TAYLOR0, TAYLR
	DIFF	DESOLVE, ILAP, LDEC
	GRAPH	DEFINE, GROBADD, PLOT, PLOTADD, 2D/3D (submenu), SIGNTAB, TABVAL, TABVAR. The 2D/3D softkey pops up the "Plost Setup" form. ????
	DERVX	This is a command.
	INTVX	This is a command.
ALG	Algebraic manipulations menu. Commands: COLLECT, EXPAND, FACTOR, LNCOLLECT, LIN, PARTFRAC, SOLVE, SUBST, TEXPAND	
MATRICES	Matrix functions menu.	
	Submenu	*Commands*
	CREAT	COL, ROW, AUGMENT, IDN, CON, →DIAG, DIAG→, GET, GETI, HILBERT, PUT, PUTI, RANM, RDM, REPL, SUB, VANDERMONDE
	OPER	ABS, AXL, AXM, CNRM, COND, DET, HADAMARD, LSQ, MAD, RANK, RNRM, RSD, SIZE, SNRM, SRAD, TRACE, TRAN
	FACT	LQ, LU, QR, qr, SCHUR, SVD, SVL
	QUADF	AQX, CHOLESKY, GAUSS, QXA, SYLVESTER
	LIN S	LINSOLVE, REF, rref, RREF, SYST2MAT
	LINAP	IMAGE, ISOM, KER, MKISOM
	EIGEN	DIGMAP, EGV, EGVL, JORDAN, PCAR, PMINI
	VECT	BASIS, CROSS, DOT, GRAMSCHMIDT, IBASIS
STAT	Statistics functions choose box:	
	Single-var…	Single variable statistics form. ??
	Frequencies…	Frequencies form. ??
	Fit data…	Fit data form. ??
	Summary stats…	Summary statistics form. ??
	Hypoth. tests…	Displays the Hypothesis tests choose box:
	Z-INT: 1 μ	??
	Z-INT: μ1-μ2	??
	Z-INT: 1 P	??
	Z-INT: P1-P2	??
	T-INT: 1 μ	??
	T-INT: μ1-μ2	??

	Conf. interval…		Displays the Confidence intervals choose box:
		Z-INT: 1 μ	??
		Z-INT: μ1-μ2	??
		Z-INT: 1 P	??
		Z-INT: P1-P2	??
		T-INT: 1 μ	??
		T-INT: μ1-μ2	??
CONVERT	Submenus:		
	UNITS	Displays the UNITS menu.	
	BASE	???	
	TRIG	???	
	REWRITE	???	
	MATRX	???	
UNITS	Units menu		
ARITH	Arithmetic menu:		
	INTEG	???: EULER, IABCUV, IBERNOULLI, ICHINREM, IDIV2, IEGCD, IQUOT, IREMAINDER, ISPRIME?, NEXTPRIME, PA2B2, PREVPRIME	
	POLY	???: ABCUV, CHINREM, CYCLOTOMIC, DIV2, EGCD, FACTOR, FCOEF, FROOTS, GCD, HERMITE, HORNER, LAGRANGE, LCM, LEGENDRE, PARTFRAC, PCOEF, PROOT, PTAYL, QUOT, RESULTANT, REMAINDER, STURM, STURMAB	
	MODUL	???: ADDTMOD, DIVMOD, DIV2MOD, EXPANDMOD, FACTORMOD, GCDMOD, INVMOD, MOD, MODSTO, MULTMOD, POWMOD, SUBTMOD	
	PERM	???: CIRC, C2P, P2C	
	DIVIS	Various commands, see section **Commands**.	
	FACTORS	???	
	LGCD	???	
	PROPFRAC	???	
	SIMP2	???	
CMPLX	Complex functions: ARG, ABS, CONJ, i, IM, NEG, RE, SIGN		
LIB	Lib menu. Shows the available memory spaces in a softmenu: :0: (IRAM), :1: (ERAM), :2: (FLASH), :3: (SD, optional). • When pressing a softmenu key the current directory is changed to this memory space. • Note that the VAR key always displays the variables in the HOME memory space. • The HOME command and the CHDIR command in the FILES menu change the current directory of the "normal" memory space but do not activate it. Press the VAR key to do so. • Not all memory spaces support directories. • See section **Variables**, Memory spaces and FILES menu in section **Menus**.		
BASE	Binary number base selection and functions menu. See **Data Types**.		

<table>
<tr><td></td><td colspan="2">• Binary numbers must be entered using the digits of the currently selected number base. Ie. #1F causes an error unless in hex mode.
• Alternatively, it is possible to specify a suffix: #1Fh is accepted in all number bases.
• Suffixes are b, o, d and h for binary, octal, decimal and hex numbers.
• When the number base is changed only the display mode of Binary numbers in the stack changes!
• Commands STWS and RCWS are used to manipulate the word size of binary numbers. The word size can be 1–64 bit.
• Changing the word size does not modify any binary numbers nor does it restrict the range when entering binary numbers. Ie. entering #8888 at a word size of 3 bit creates a binary number that is displayed as #0. But when changing the word size to 64 bits #8888 reappears.
• However, the word size does have an effect as soon as calculations (including rotation) are involved: In this case the result is truncated to the word size.</td></tr>
<tr><td></td><td colspan="2">Submenus:</td></tr>
<tr><td></td><td>LOGIC</td><td>Logic operations: AND, OR, XOR, NOT</td></tr>
<tr><td></td><td>BIT</td><td>Bit rotation: RL, SL, ASR, SR, RR</td></tr>
<tr><td></td><td>BYTE</td><td>Byte rotation: RLB, SLB, SRB, RRB</td></tr>
<tr><td colspan="3">Menus not accessible only thru the APPS choose box:</td></tr>
<tr><td>PLOT</td><td colspan="2">???</td></tr>
<tr><td>MATHS</td><td colspan="2">Displays the CAS/MATHS menu, see Menus.</td></tr>
<tr><td></td><td>Submenu</td><td>Commands</td></tr>
<tr><td></td><td>CMPLX</td><td>i, ABS, ARG, CONJ, DROITE, FLOOR
Strangely enough, the FLOOR function is not defined for complex numbers!</td></tr>
<tr><td></td><td>CONSTANTS</td><td>e, i, ∞, π</td></tr>
<tr><td></td><td>HYPERBOLIC</td><td>ACOSH, ASINH, ATANH, COSH, SINH, TANH</td></tr>
<tr><td></td><td>INTEGER</td><td>DIVIS, EULER, FACTOR, GCD, IEGCD, IQUOT, IREMAINDER, ISPRIME?, LCM, NEXTPRIME, PREVPRIME</td></tr>
<tr><td></td><td>MODULAR</td><td>ADDTMOD, DIVMOD, EXPANDMOD, FACTORMOD, GCDMOD, INVMOD, MODSTO, MULTMOD, POWMOD, SUBTMOD</td></tr>
<tr><td></td><td>POLYNOMIAL</td><td>EGCD, FACTOR, GCD, HERMITE, LCM, LEGENDRE, PARTFRAC, PROPFRAC, PTAYL, QUOT, REMAINDER, TCHEBYCHEFF</td></tr>
<tr><td></td><td>TESTS</td><td>ASSUME, UNASSUME, >, ≥, <, ≤ , ==, ≠, AND OR NOT IFTE</td></tr>
<tr><td>CAS</td><td colspan="2">Activated by the MAIN command or (sometimes only!) thru the APPS/CAS menu.
Note that subdirectories are not indicated by smal bars ontop of the softkey labels. Submenus:</td></tr>
<tr><td></td><td>CASCF</td><td>???</td></tr>
<tr><td></td><td>ALGB</td><td>COLLECT, DEF, EXPAND, FACTOR, PARTFRAC, QUOTE, STORE, |, SUBST, TEXTPAND, UNASSIGN</td></tr>
<tr><td></td><td>DIFF</td><td>DERIV, DERVX, DIVPC, FOURIER, IBP, INTVX, lim, PREVAL, RISCH, SERIES, TABVAR, TAYLOR0, TRUNC</td></tr>
<tr><td></td><td>MATHS</td><td>Submenus: CMPLX, CONSTANTS, HYPERBOLIC, INTEGER,</td></tr>
</table>

		MODULAR, POLYNOMIAL
	TRIGO	ACOS2S, ASIN2C, ASIN2T, ATAN2S, HALFTAN, SINCOS, TAN2C2S, TAN2SC, TAN2SC2, TCOLLECT, TEXPAND, TLIN, TRIG, TRIGCOS, TRIGSIN, TRIGTAN
	SOLVER	DESOLVE, ISOL, LDEC, LINSOLVE, SOLVE, SOLVEVX
	CMPLX	i, ABS, ARG, CONJ, DROITE, FLOOR, IM, MOD, NEG, RE, SIGN
	ARIT	Submenus: INTEGER, MODULAR, POLYNOMIAL
	EXP&LN	TEXPAND, LIN, TSIMP, LNCOLLECT, EXPLN
	MATR	TRAN, HADAMARD, rref, REF, AXM, AXL, QXA, AXQ, GAUSS, SYLVESTER, PCAR, JORDAN, MAD, LINSOLVE, VANDERMONDE
	REWRITE	DISTRIB, EPSX0, EXPLN, EXP2POW, FDISTRIB, LIN, LNCOLLECT, POWEXPAND, SINCOS, SIMPLIFY, XNUM, XQ
I/O		
		???
		???
		???
		???
		???
Constants	Displays the Constants Library form. Contains 39 physical constants from NA (Avogadro's number) to I0 (ref intensity). Activated by the CONLIB command or thru APPS/Constants lib. Use the cursor keys or shift left/shift right cursor keys to navigate thru the list of constants. Softkeys:	
	SI	Display/return values in SI units.
	ENGL	Display/return values in English units.
	UNIT	Selectes whether to return a value with or without attached units.
	VALUE	Select list display with values plus units (SI or English) or with the constant's full name.
	→STK	Return a tagged constant to the stack. Obeys the SI/ENGL/UNIT settings above.
	QUIT	Return to normal stack display.

Display

General	The display of various kinds of objects can be controlled by a number of different flags. See **Flags** and menu MODE in chapter **Menus** on how to change flags. Some suggestions for flags affecting the stack display: • Do not set flag 52 (single line display) because it disables all other formatting features. • Better not set flag 65 (1/all level multiline) so that only stack level 1 is displayed in textbook mode ("pretty print"). Higher stack levels are displayed in a compact 1-line mode. • Do not set flag 72 because the small font is hard to read. • Do not set flag 83 (sysRPL display) because it disables all other formatting features. • Set flag 79 (textbook display) according to your preferences. Unfortunately, when textbook display is selected complex numbers are displayed in a

	single line only.

Command List

<table>
<tr><td>General</td><td colspan="3">• Commands usually operate on specific data types.
• Usually, if a function operates on a specific data type it also operates on a List of objects of this type. The resulting list contains the results of the function applied to the individual elements of the input list(s).
• Not all commands support List operations and some commands (notably "+") perform special operations when applied to lists.
• Operations can be performed on Real numbers that have units attached. However units may only be products of powers of base units. Thus <code>SQ(2_m)</code> returns <code>4_m^2</code> but <code>LN(2_m)</code> causes an error. Also, units must be compatible to the desired operation: <code>4_m 2_s ÷</code> returns <code>2_m/s</code> but <code>4_m 2_m +</code> causes an "inconsistent units" error.</td></tr>
<tr><td>!</td><td colspan="3">The faculty operator. Operates on Real numbers or a list of Real numbers. In the latter case the faculty operator is performed on each element of the list producing another list.
For fractional Reals it returns the Gamma function.</td></tr>
<tr><td>%</td><td colspan="3">Percent. Only real arguments.</td></tr>
<tr><td>%CH</td><td colspan="3">Percentual difference from value in level 2 to value in level 1.
Only real arguments.</td></tr>
<tr><td>%T</td><td colspan="3">Percentage of level 1 out of level 2:
<code>200 10 %T</code> returns 5 because 10 is 5% out of 200.
Only real arguments.</td></tr>
<tr><td>'</td><td colspan="3">Delimiter for names and algebraic expressions.</td></tr>
<tr><td>x</td><td colspan="3">Multiplication.</td></tr>
<tr><td></td><td>Stack 2</td><td>Stack 1</td><td>Result</td></tr>
<tr><td></td><td>Real</td><td>Real</td><td>Real</td></tr>
<tr><td></td><td>Real</td><td>Complex</td><td>Complex</td></tr>
<tr><td></td><td>Complex</td><td>Real</td><td>Complex</td></tr>
<tr><td></td><td>Binary</td><td>Real</td><td>Binary. Real number rounded to integer.</td></tr>
<tr><td></td><td>Real</td><td>Binary</td><td>Binary. Real number rounded to integer.</td></tr>
<tr><td></td><td>Binary</td><td>Binary</td><td>Binary</td></tr>
<tr><td></td><td>List</td><td>Real
Complex
Matrix
Vector</td><td>List with all list elements multiplied by the object.
Causes an error if multiplication is not defined for one of the list elements (ie. for programs).
<code>{1 2} [3 4] x</code> results in <code>{[3 4] [6 8]}</code></td></tr>
<tr><td></td><td>Real
Complex
Matrix
Vector</td><td>List</td><td>List with all list elements multiplied by the object. See above.</td></tr>
<tr><td></td><td>List</td><td>List</td><td>List with corresponding values multiplied.
See above.</td></tr>
<tr><td></td><td>Vector</td><td>Real</td><td>Vector/matrix with each element multiplied by</td></tr>
</table>

	Matrix	Complex	number.
	Real Complex	Vector	Vector with each element multiplied by number.
	Matrix	Vector	Vector resulting from matrix multiplication. Dimensions must match.
	Matrix	Matrix	Matrix resulting from matrix multiplication. Dimensions must match.
+	Addition.		
	Stack 2	*Stack 1*	*Result*
	Real	Real	Real
	Real	Complex	Complex
	Complex	Real	Complex
	Binary	Real	Binary. Real number rounded to integer.
	Real	Binary	Binary. Real number rounded to integer.
	Binary	Binary	Binary
	List	Object	List with object inserted at end. Also applies if object is a program. Does not apply if object is a list.
	Object	List	List with object inserted at beginning. Also see List/Object.
	List	List	Combined lists.
	Vector	Vector	Vector, dimensions must match.
	Matrix	Matrix	Matrix, dimensions must match.
	String	Object	String with object appended at end. Also applies if object is a string or a program. Does not apply if object is a list.
	Object	String	String with object inserted at beginning. Also see String/Object.
-	Subtraction.		
	Stack 2	*Stack 1*	*Result*
	Real	Real	Real
	Real	Complex	Complex
	Complex	Real	Complex
	Binary	Real	Binary. Real number rounded to integer.
	Real	Binary	Binary. Real number rounded to integer.
	Binary	Binary	Binary
	Vector	Vector	Vector, dimensions must match.
	Matrix	Matrix	Matrix, dimensions must match.
	List	Real Complex	List with the number substracted from all list elements. Causes an error if substraction is not defined for one of the list elements (ie. for programs).
	Real Complex	List	List with the list values substracted from the number. Also see List/Real.
	List	List	List with corresponding elements substracted from each other. Dimensions must match. Also see List/Real.
/	Division.		

<table>
<tr><td></td><td>
<table>
<tr><th>Stack 2</th><th>Stack 1</th><th>Result</th></tr>
<tr><td>Real</td><td>Real</td><td>Real</td></tr>
<tr><td>Real</td><td>Complex</td><td>Complex</td></tr>
<tr><td>Complex</td><td>Real</td><td>Complex</td></tr>
<tr><td>Binary</td><td>Real</td><td>Binary. Real is rounded to integer before division.
Note: All divisions involving binaries that divide by 0 do not cause an error but result in 0!</td></tr>
<tr><td>Real</td><td>Binary</td><td>Binary.</td></tr>
<tr><td>Binary</td><td>Binary</td><td>Binary.</td></tr>
<tr><td>List</td><td>Real
Complex
Matrix</td><td>List with all elements divided by number.
Causes an error if division is not defined for one of the list elements (ie. for programs).</td></tr>
<tr><td>Real
Complex
Vector
Matrix</td><td>List</td><td>List with number divided by each list element.
See above.</td></tr>
<tr><td>List</td><td>List</td><td>List with corresponding elements divided by each other.
See above.</td></tr>
<tr><td>Vector
Matrix</td><td>Real
Complex</td><td>Vector/matix with each element divided by number.</td></tr>
<tr><td>Vector</td><td>Matrix</td><td>??</td></tr>
<tr><td>Matrix</td><td>Matrix</td><td>??</td></tr>
</table>
</td></tr>
<tr><td>;</td><td>???</td></tr>
<tr><td><</td><td>Comparisn operator.
• Compares values stack2 < stack1.
• Returns Real values 0 or 1.
• Compares Real or Binary values but not in mixed mode.
• Compares strings lexically.
• Compares two lists component-by-component and returns a list containing the comparisn results. List lengths must match.</td></tr>
<tr><td>=</td><td>Equality operator in algebraic expressions, ie. <code>'Y=3*X+7'</code>.</td></tr>
<tr><td>==</td><td>Comparisn operator, returns Real number 0 or 1.
• Compares all data types but different data types are never equal, ie. <code>3 #3 ==</code> returns 0 because Reals and Binaries are of different type.
• Lists are not compared component-wise but as a whole. All their elements must be identical. Lists of different length are never equal.
• Programs are assumed to be equal if they contain the same code.
• Symbolic arguments are evaluated, see SAME.
• See "<" and also SAME.</td></tr>
<tr><td>></td><td>Comparisn operator, returns 0 or 1, see "<".</td></tr>
<tr><td>?</td><td>The undefined symbol. ??</td></tr>
<tr><td>ABCUV</td><td>Returns the solution U &V for the Bezout polynomial AU+BV=C.
<code>'X+1' 'X-1' 2 ABCUV</code> returns <code>1 -1</code>.
Variable X must not exist, radians and exact mode must be selected.</td></tr>
</table>

<table>
<tr><td></td><td>If this is not the case the calculator will ask to make the appropriate changes.</td></tr>
<tr><td>ABS</td><td>Absolute value of a Real or Complex number, Vector or Matrix.
For all types this is the square root of the sum of the squares of all elements. See NEG and CONJ.</td></tr>
<tr><td>ACK</td><td>Has something to do with alarms.
Not listed in the Reference Manual.</td></tr>
<tr><td>ACKALL</td><td>Has something to do with alarms.
Not listed in the Reference Manual.</td></tr>
<tr><td>ACOS</td><td>Arcus cosine.<table><tr><td>Stack 1</td><td>Result</td></tr><tr><td>Real</td><td>Real</td></tr><tr><td>Real</td><td>Complex, if argument >1</td></tr><tr><td>Complex</td><td>Complex</td></tr><tr><td>List</td><td>Apply function to all list elements.
Causes an error if not defined for one of the list elements.</td></tr></table></td></tr>
<tr><td>ACOS2S</td><td>Replaces ACOS(x) with π/2-ASIN(x) in expressions.
See ABCUV for general CAS issues.</td></tr>
<tr><td>ACOSH</td><td>Hyperbolic arcus cosine, see ACOS.</td></tr>
<tr><td>ADD</td><td>Identical to "+" with one exception: Instead of joining lists the list elements are added up element-wise. The dimensions must match:
<code>{1 2} {3 4} ADD</code> returns <code>{4 6}</code>.
ADD really only exists to overcome the traditional, non-standard behaviour of "+" with respect to List arguments.</td></tr>
<tr><td>ADDTMOD</td><td>Adds two expressions modulo the current modulus.
<code>'11X+5' '8X+6' ADDTMOD</code> returns <code>'6X-2'</code>
for modulus=13 because the sum '19X+11' modulo 13 is '6X-2' (why not '6X+11' ?)
See ABCUV for general CAS issues and MODSTO.</td></tr>
<tr><td>ADDTOREAL</td><td>Assumption on a variable to be real. ??
See ASSUME, UNASSUME.</td></tr>
<tr><td>ALGB</td><td>Displays the CAS/ALGB menu, see Menus.</td></tr>
<tr><td>ALOG</td><td>Exponential function base 10. Operates on Real or Complex number, or a List of Real/Complex numbers. Returns a symbolic expression for symbolic arguments, ie: <code>'A' EXP</code> returns <code>ALOG(A)</code>.
See EXP, EXPM, LN, LNP1, LOG.</td></tr>
<tr><td>AMORT</td><td>?? 6-39</td></tr>
<tr><td>AND</td><td>AND operator.
For Binary numbers it returns the bit-wise AND.
For Real numbers it returns the logical AND.
For Lists it returns a list containing the result of the component-wise AND. List lengths must match.
Not defined for mixed Real/Binary arguments.
Returns a symbolic expression for symbolic arguments.</td></tr>
<tr><td>ANIMATE</td><td>22-31</td></tr>
<tr><td>ANS</td><td>Replaces the value in stack value 1 by a copy of the value in the stack level indicated by the Real number in stack level 1.</td></tr>
</table>

	`10 20 30 40 3 PICK` returns `10 20 30 40 30`. This is very similar to but not quite the same as PICK, see there.
APPLY	???
ARC	?? drawing an arc 22-25
ARCHIVE	???
ARG	Returns as a Real number the angle of a Complex/Real number with the X-axis. The angle is expressed in the current angle mode (GRAD, RAD or DEG). For a plain Real number this is always 0. For (0 0) the result is 0 as well.
ARIT	Displays the arithmetic menu with subdirectories INTEG, MODUL, POLYN and MAIN which returns to the CAS menu. Note that this arithmetic menu is different from the one that can be accessed thru the APPS/CAS menu choose box.
ARRY→	Splits a vector or matrix into components and returns a list containing the vector/matrix dimensions. See →ARRY. Examples: • For `[[1 2 3][4 5 6]]` this returns `1 2 3 4 5 6 {2 3}` on the stack. • For `[1 2 3 4 5]` it returns `1 2 3 4 5 {5}`.
ASIN	Arcus sine, see ACOS.
ASIN2C	Replaces ASIN(x) with π/2-ACOS(x) in an expression. See ABCUV for general CAS issues.
ASIN2T	Replaces ASIN(x) with ATAN(x/√(1-x²)) in algebraic expressions. See ABCUV for general CAS issues.
ASINH	Hyperbolic arcus sine, see ACOS.
ASN	20-6
ASR	Shift Binary number one bit right. Duplicates the topmost bit and discards bit0.
ASSUME	Make an assumption on a variable. ??
ATAN	Arcus tangent, see ACOS.
ATAN2S	???
ATANH	Hyperbolic arcus tangent, see ACOS.
ATICK	Defines tick-mark parameters for plots axes. • A Real number sets the tick mark distance to this amount of units for the x- and y-axis. Units are not pixel but rather depend on the width and height of the plot. • A list `{x y}` of two Real numbers set the tick mark distance for the x and y axis independently. • A Binary number sets the tick mark distance in pixel. • A List of two Binary numbers sets the tick mark distance for the x- und y-axis independently in pixels.
ATTACH	???
AUGMENT	Add an object to a list. `{ 66 } 'X' AUGMENT` returns `{ 66 'X' }`. This is similar to the "+" operator but the latter can also append to a list that is located in stack level 1 which AUGMENT cannot: `'X' { 66 } AUGMENT` causes an error.
AUTO	Determines the plot range for 2-dimensional plots.

	Syntax ??
AXES	Draw axes in a plot. • `(0 0) AXES` draws axes at position (0,0). • `{(1 2) 5. "t" "V"} AXES` draws axes at position (1,2), tick marks every 5 units, labels the x-axis with "t" and the y-axis with "V". See ATICK for more details on tick marks.
AXL	9-30
AXM	11-16
AXQ	11-59
BAR	???
BARPLOT	???
BASIS	???
BAUD	Select the baud rate for the serial port – even though the HP-49g+ doesn't have one. Automatically creates the IOPAR variable in the current directory if it doesn't already exist.
BEEP	Creates a tone with given frequency and duration. `1000 2 BEEP` creates a tone at 1000 Hz for 2 seconds. Frequency and duration must be Real numbers.
BESTFIT	Sets the curve fitting model to "best fit" and fits a curve into the current statistics data in ΣDAT. No value is returned but the y-offset and slope in ΣPAR are updated to reflect the fit. Also, the fit model is automatically to the best model, that is: EXPFIT, LINFIT, LOGFIT or PWRFIT. If ΣPAR does not exist it is created in the current directory. See ΣPAR in section **Variables**.
BIN	Sets binary display format for Binary numbers. See DEC, HEX, OCT.
BINS	see reference manual pg. 18-19.
BLANK	???
BOX	???
BUFLEN	???
BYTES	Takes an object from stack level 1 and returns two values: Its address as a Binary number and its size in bytes. See **Data Types**.
B→R	Convert a Binary number to a Real number. For very large Binary numbers the conversion looses significant digits. See R→B.
C2P	???
CASCFG	???
CASCMD	Displays a choose box containing all the CAS commands. Pressing OK displays a help screen on the selecting command.

CASE	Program control instruction. Syntax:
	`CASE`
	`test1 THEN code1 END`
	`test2 THEN code2 END`
	`…`
	`codeDefault`
	`END`
	The testx instructions are tested until one evaluates to true. In this case the corresponding codex instructions are executed and the

	execution continues after END. If none of the testx instructions evaluates to true the optional codeDefault is executed. See IF.
CEIL	Returns the next integer number ≥ stack level 1. Operates only on Real numbers and lists of Real numbers. Ie. `-2.5 CEIL` returns `-2`.
CENTR	22-7
CF	Clear Flag(s). The flag number must be given as a Real number or a list of Real numbers. When a list is specified all the corresponding flags are cleared. See SF and FC?.
CHINREM	5-12
CHOLESKY	???
CHOOSE	21-35
CHR	Creates a string with a single character from the given ASCII value. Defined for real arguments and lists. Note that strange strings can be created from arguments ≤0.
CIRC	12-54
CKSM	???
CLEAR	Clear the stack.
CLKADJ	25-3
CLLCD	Clear LCD screen. Mainly used when plotting.
CLOSEIO	???
CLVAR	???
CLΣ	Clear the statistics variable ΣDAT. Does nothing if ΣDAT doesn't exist.
CMPLX	Displays the CAS/CMPLX menu, see **Menus**.
CNRM	???
COL+	10-22
COL–	10-22
COLCT	???
COLLECT	5-4
COLΣ	???
COL→	???
COMB	Combinations. Ie. A B COMB returns A! / [B! • (A – B)!] This is the number of possibilities to select B elements from a group of A different elements where different sequences do not count separately. See PERM.
CON	Takes a List of two Real numbers and a Real number from the stack and returns a Matrix of the dimensions specified in the List. All elements are set to the value of the 2nd argument: `{2 3} 4 CON` returns `[[4 4 4][4 4 4]]`.
COND	???
CONIC	???
CONJ	Conjugate of Real or Complex number, Vector or Matrix. This negates the imaginary part of the number (if any). See ABS and NEG.
CONLIB	Displays the constants library form. See APPS menu in **Menus**.
CONST	???

CONSTANTS	Displays the MATHS/CONSTANTS menu, see **Menus**.
CONT	Contine a halted program. Does nothing if there is no halted program. See HALT, DBUG, SST, KILL.
CONVERT	Converts a Real number with units to a number expressed in specified other units: `125_km 1_m CONVERT` returns `125000_m`. Incompatible units cause an error. See UBASE.
CORR	Returns the correlation coefficient of the most recent curve fit. A value of 1 indicates a perfect fit. See BESTFIT.
COS	Cosine, see ACOS.
COSH	Hyperbolic cosine, see ACOS.
COV	Returns the covariance of the most recent curve fit. A value of 1 indicates a perfect fit. See BESTFIT.
CR	Sends a CR-LF sequence to the printer. ??
CRDIR	Takes a name from the stack and creates a subdirectory with this name under the current directory.
CROSS	Cross product of two Vectors. Either vector may be 2- or 3-dimensional, mixed dimensions are allowed.
CSWP	???
CURL	???
CYCLOTOMIC	???
CYLIN	Selects cylindical display mode for 3-dim Vectors and polar format for Complex numbers and 2-dim Vectors. See MODE menu in **Menus**. See SPHERE and CYLIN.
C→PX	Converts a Complex number representing a point in the current plot to Binary pixel coordinates. Ie. if the lower-left corner of the plot as specified in PPAR is (0,0) and the upper-right corner is (1,1) and decimal base is selected: `(0 0) C→PX` returns `{#0d #63d}`. `(1 1) C→PX` returns `{#130d 0d}`. Note that plot coordintes increase from bottom to top whereas pixel coordinates increase from top to botton, (#0 #0) referring to the top-left corner. If the specified coordinates lie outside the plot area the returned pixel coordinates lie beyond the actual LCD display borders. See PX→C.
C→R	Split a Complex object into two Real objects. • A Complex number is split into two Real numbers. • A complex Vector is split into two real Vectors: `[(1 2)(3 4)(5 6)] C→R` returns `[1 3 5] [2 4 6]` • A complex Matrix is split into two real Matrices. See R→C.
DARCY	???
DATE	Return current date as a Real in the form `d.mmyyyy`.
DATE+	Adds a number of days in stack level 1 to a date in stack level 2. `18.042004 100 DATE+` returns `27.072004`. Add negative numbers to substract days. Both arguments must be Real numbers. See DATE, DDAYS.
DBUG	Takes a program name from the stack and starts the code in debug mode which allows single-stepping.

	See SST, SST↓, NEXT, HALT, KILL.
DDAYS	Returns the number of days between two dates. `1.052004 1.042004 DDAYS` returns `-30`. Both arguments must be Real numbers. See DATE, DATE+.
DEC	Sets decimal display format for Binary numbers. See BIN, HEX, OCT.
DECR	Take a variable name, increments the variable's numeric value and returns the value to the stack. See INCR, STO*.
DEDICACE	???
DEF	???
DEFINE	A shortcut used to create small programs in functional notation: `'FF(X)=X*X+3' DEFINE` creates a variable FF with the following contents: ◀◀ → `X 'X*X+3'` ▶▶.
DEG	Selects degress (360) for trigonometric calculations, see RAD, GRAD.
DEGREE	Returns the degree of a polynomial. `'X^2-X' DEGREE` returns 2. `'X+8' DEGREE` returns 1. `17 DEGREE` returns 0. `0 DEGREE` returns –1. See ABCUV for general CAS issues.
DELALARM	Delete the specified alarm 1-10. Causes an error if the alarm doesn't exist.
DELAY	Choose printer delay. Arg: Real 0..7
DELKEYS	???
DEPND	???
DEPTH	Return number of objects on the stack. Result: Real.
DERIV	???
DERVX	???
DESOLVE	???
DET	Calculates the determinant of a matrix. ??
DETACH	???
DIAGMAP	???
DIAG→	Takes a Vector of values and a size-2 List and creates a Matrix with the given dimensions and the values of the vector in the diagonal. Values are discarded or zeros added as necessary. `[1 2 3 4] {2 3} DIAG` returns `[[1 0 0][0 2 0]]`. See →DIAG.
DIFF	Displays the CAS/DIFF menu, see **Menus**.
DIFFEQ	???
DIR	???
DISP	???
DISPXY	???
DISTRIB	???
DIV	???
DIV2	???
DIV2MOD	???
DIVIS	???

DIVMOD	???
DIVPC	???
DO	Syntax: `DO code UNTIL test END` • Performs code until test returns a non-0 value. • The test value must be a Real. • code is executed at least once. • The test clause may be omitted if a suitable value is already present on the stack. • The code clause may be omitted.
DOERR	Throws an error and aborts the current program. The arguments can be a Real or Binary number, or a string: • `0 DOERR` displays "Interrupted". All arguments ≤0 cause this output. • `1 DOERR` displays "Error: Insufficient memory" and similar messages for other error codes. • `99 DOERR` displays "Error:" because error 99 is not defined. • `"FATAL" DOERR` displays FATAL.
DOLIST	???
DOMAIN	???
DOSUBS	???
DOT	Calculates to dot-product of two vectors of the same length or two square matrices. ??
DRAW	???
DRAW3DMATRIX	???
DRAX	???
DROITE	Calculates a line thru two points given by two complex numbers. The result is a symbolic expression. `(1 1) (2 2) DROITE` returns `'Y=X+1-1'`.
DROP	Discard object in stack level 1. In idle mode the delete-left arrow key performs the same operation. Note that all the DROPx commands will issue an error if there aren't enough stack objects present.
DROP2	Discard objects in stack level 1 and 2.
DROPN	Takes a Real number from the stack and the drops the specified amount of stack object.
DTAG	Remove the tag from the object in stack level 1: `:Result:125 DTAG` returns `125`.
DUP	Duplicates the object on top of the stack but doesn't evaluate it. Note that all the DUPx commands will issue an error if there aren't enough stack objects present.
DUP2	Duplicate two topmost stack elements: `A B DUP2` returns `A B A B`.
DUPDUP	Double DUP: `A DUPDUP` returns `A A A`.
DUPN	Takes Real value n from the stack and then doublicates the topmost n elements of the stack: `A B C 3 DUPN` returns `A B C A B C`.
D→R	Converts a Real number or a list of Real numbers from degrees (360) to radians (2π). Note that this independent of the current angle mode. See R→D.
EDIT	???

EDITB	???
EGCD	???
EGV	???
EGVL	???
ELSE	Used with IF, see there.
END	Used for various programming constructs. See IF, DO, WHILE.
ENDSUB	???
ENG	Selects the engineering display format, see MODE menu in section **Menus**. The argument in stack level 1 plus 1 is the number of displayed digits. When a list is specified the last list element determines the number of digits. See FIX, SCT and STD.
EPSX0	???
EQW	???
EQ→	???
ERASE	???
ERR0	Clears the error code.
ERRM	Returns the most recent error as an English language textual String. Returns an empty string if there was no error.
ERRN	Returns the most recent error code as a Binary number or 0 if there was no error.
EULER	???
EVAL	Evaluate expression in stack level 1.
EXLR	
EXP	Exponential function base e (2.71828…). See ALOG for details.
EXP&LN	Displays the CAS/EXP&LN menu, see **Menus**.
EXP2HYP	
EXP2POW	
EXPAN	
EXPAND	
EXPANDMOD	
EXPFIT	Sets the curve fitting model to "exponential fit" and fits a curve into the current statistics data in ΣDAT. The resulting curve is of the form y=a*exp(b*x). See BESTFIT.
EXPLN	
EXPM	Returns e^X–1 for more accuracy if X is close to 0. See ALOG for details. Inverse function is LNP1.
EYEPT	?? reference pg. 22-11
F0λ	Calculates the fraction of the energy F0λ(T,λ) emitted by a black body radiator of temperature T which falls into the wave length interval 0 and λ. T and λ can have associated dimensions. If omitted Kelvin and meters are assumed. `6000 1E-6 F0λ` returns 0.7377922… which indicates that the sun (a black body surface temperature 6000 Kelvin) radiates 74% of its total energy emission at wavelengths of 1 µm or more.
FACT	
FACTOR	
FACTORMOD	

FACTORS	
FANNING	
FAST3D	
FC?	Test specified flag whether it is clear. • The flag number must be a Real or a list of Reals. • When specifying a list then all the flags in the list are tested and a list with the results is returned. • The returned value is a Real 0 or 1. • See CF, FC?C, FS?, FS?C
FC?C	Test specified flag whether it is clear and then clear it, see FC?
FCOEF	
FDISTRIB	
FFT	
FILER	
FINDALARM	Seems to try to find an alarm by date/time ??
FINISH	
FIX	Selects the fixed-point display format, see ENG.
FLASHEVAL	
FLOOR	Returns the next integer number ≤ stack level 1. Operates only on Real numbers and lists of Real numbers. Ie. `-2.5 FLOOR` returns –3
FONT6	
FONT7	
FONT8	
FONT→	
FOR	Syntax 1: `start end FOR name code NEXT` • The "code" is executed end-start+1 times. • The current loop counter value is accessible thru variable "name". • start and end may be omitted if suitable values already exist on the stack. • If start<=end the "code" is executed once. • start and end must be Real or Binary but not in mixed mode. • Also see START command. Syntax 2: `start end FOR name step-size STEP` • Here the internal loop counter is encremented by step-size. • The loop stops if the internal loop counter is ≥end.
FOURIER	
FP	Return fractional part of number. Operates on a Real number or a List of Real numbers. Preserves the sign: `-2.5 FP` returns `-0.5` See IP.
FREE	
FREEZE	
FROOTS	
FS?	Test specified flag whether it is set, see FC?
FS?C	Test specified flag whether it is set and then clear it, see FC?
FUNCTION	
FXND	CAS.
GAMMA	Gamma function of a Real or Complex number.

	See faculty operator "!", PSI and Psi. Note that Gamma(x) = Γ(x) = (x–1)!.
GAUSS	
GBASIS	
GCD	
GCDMOD	
GET	Returns an element from a List, Vector or Matrix: `'M' {1 1} GET` returns element (1,1) of matrix M. `'VL' {2} GET` or `'VL' 2 GET` returns the 2nd element of vector or list VL. Note that all arguments are dropped, including the name of the matrix. See PUT, GETI.
GETI	Similar to GET but does not discard the source List/Matrix/Vector nor the index. Even more, the index is incremented. `'M' {1 1} GETI` returns `'M' {1 2} 5` (assuming M(1,1)=5). The column index is incremented first and only when it wraps the row is incremented. Finally, the index wraps back to `{1 1}`. `'VL' {2} GETI` returns `'VL' {3}` 5 (assuming VL(2)=5). `'VL' 2 GETI` returns `'VL' 3` 5 (assuming VL(2)=5). Once the index reaches the maximum value it wraps back to 1. See GET, PUTI.
GOR	
GRAD	Selects degress (360) for trigonometric calculations, see RAD, DEG.
GRAMSCHMIDT	
GREDUCE	
GRIDMAP	
GROB	
GROBADD	
GXOR	
HADAMARD	
HALFTAN	
HALT	This instruction halts execution of the program. A small "HLT" symbol on top of the LCD display indicates that there is one or more halted programs. While a program is in halted state another program can be executed which may be halted as well. A halted program can be single-stepped using SST commands or terminated using KILL or continued using CONT. See DBUG, SST, CONT, KILL.
HEAD	Returns the first element of a List as a plain object. Returns the first character of a string as a string. An error occurs if the List or String is empty. See TAIL.
HEADER→	
HELP	
HERMITE	
HESS	
HEX	Sets hexadecimal display format for Binary numbers. See BIN, DEC, OCT.
HILBERT	

HISTOGRAM	
HISTPLOT	
HMS+	Add up Real numbers in h.mmssf format where h is hours, mm is minutes, ss is seconds and f is fractional seconds. Can also operator on two equal-length lists of Real numbers. Cannot operate on an exact number like '1/3600'. Note that the h.mmssf format is different from the d.mmff format used in GPS devices where d is degrees, mm is minutes and f is fractional minutes! See →HMS and other HMSx functions.
HMS-	Substract Real numbers in h.mmssf format. See →HMS and other HMSx functions.
HMS→	Convert a Real number from h.mmssf format to fractional hours. Operates on a Real number or a list of real numbers only. See HMS+, →HMS and other HMSx functions.
HOME	Change the current directory to toplevel HOME.
HORNER	
HYPERBOLIC	Displays the MATH/HYPERBOLIC menu, see **Menus**.
IABCUV	
IBASIS	
IBERNOULLI	
IBP	
ICHINREM	
IDIV2	
IDN	• Take a Real number and returns a square identity Matrix of the given size. • Takes a square Matrix and converts it to an identity Matrix. • Takes the name of a Matrix and converts it to an identity Matrix. The name is removed from the stack.
IEGCD	
IF	Syntax: `IF <test> THEN <code1> ELSE <code2> END` • The test instruction must produce a Real number value on the stack which is tested to be non-zero. In this case code1 is executed, otherwise code2. No other data types except Reals are permitted! • The ELSE branch may be omitted. • The test instruction may be omitted if a suitable stack value is already present. Thus "IF X THEN A ELSE B" is in RPN mode equivalent to "X IF THEN A ELSE B". • code1 and and code2 may be omitted.
IFERR	Syntax: `IFERR <test> THEN <error> ELSE <normal> END` • The error code is executed if an error occurs while the test code is processed. Otherwise the normal code is executed. • Here the test instructions cannot be omitted. • code1 and and code2 may be omitted. • The ELSE branch may be omitted.
IFFT	
IFT	Syntax: <`test> <true-code> IFT`

	• If the test instruction evaluates to a non-zero Real number the true-code is evaluated. • The test instruction may be omitted if a suitable value on the stack already exists.
IFTE	Syntax: `<test> <true-code> <false-code> IFT` • If the test instruction evaluates to a non-zero Real number the true-code is evaluated, otherwise the false-code. • The test instruction may be omitted if a suitable value on the stack already exists. • Can be used in an algebraic expression: `'y=IFTE(x>0,x*x,x/2)'`.
ILAP	Inverse Laplace transformation. 16-12
IM	Return the imaginary part of an object. Object can be a Real or Complex number, a real or complex Vector or Matrx, a List of Real or Complex numbers. See RE.
IMAGE	
INCR	Take a variable name, decrements the variable's numeric value and returns the value to the stack. See INCR, STO*.
INDEP	
INFORM	
INPUT	
INT	
INTEGER	Displays the MATH/INTEGER menu, see **Menus**.
INTVX	
INV	1/x function. Operates on Real number, List of Real numbers or square Matrix. For a matrix the inverse is calculated. In exact mode 1/0 returns the infinite result '∞'.
INVMOD	
IP	Return integer part of number. Operates on a Real number or a List of Real numbers. Preserves the sign: `-2.5 IP` returns `-2`. See FP.
IQUOT	
IREMAINDER	
ISOL	
ISOM	
ISPRIME?	
I→R	Coverts an exact integer number into a Real number. See R→I.
JORDAN	
KER	
KERRM	
KEY	
KEYEVAL	
KEYTIME→	
KGET	
KILL	Kills (terminates) the most recently halted program. Does nothing if there is no halted program. See HALT, CONT, SST.

LABEL	
LAGRANGE	
LANGUAGE→	
LAP	Laplace transformation. 16-12
LAPL	
LASTARG	Returns the arguments of the most recent command to the stack. Note that this is different from UNDO because LASTARG does not remove the potential results of the command: `2 3 + LASTARG` returns `5 2 3`. `2 3 + UNDO` returns `2 3`.
LCD→	Returns a 131x64 pixel graphics object (GROB, type 11) containing the current contents of the LCD display. See LCD→.
LCM	
LCXM	
LDEC	Solves linear differential equations with constant coefficients.
LEGENDRE	
LGCD	
LIBEVAL	
LIBS	
LIN	
LINE	
LINFIT	Sets the curve fitting model to "linear fit" and fits a curve into the current statistics data in ΣDAT. The resulting curve is of the form y=a*x + b. See BESTFIT.
LININ	Not listed in the Reference Manual.
LINSOLVE	
LIST→	Splits a list into separate intems in the stack. Stack level 1 receives the number of list objects. See →LIST. Example: `{'A' 17 (3 4) [5 6] 'LN(C)'} LIST→` returns `'A' 17 (3 4) [5 6] 'LN(C)' 5`. For the empty list 0 is returned. See →LIST.
LN	Logarith base e (2.71828…). See ALOG for details.
LNAME	Not listed in the Reference Manual.
LNCOLLECT	
LNP1	Returns LN(1+X) for more accuracy if X is close to 0. See ALOG for details. Inverse function is EXPM.
LOCAL	Not listed in the Reference Manual.
LOG	Logarithm base 10. See ALOG for details.
LOGFIT	Sets the curve fitting model to "logarithmic fit" and fits a curve into the current statistics data in ΣDAT. The resulting curve is of the form y=a+b*ln(x). See BESTFIT.
LQ	
LR	Returns two tagges values indicating the y-axis interception offset (`Intercept:`) and the slope (`Slope:`) of a statistical data fit of ΣDAT based on ΣPAR. See BESTFIT.
LSQ	
LU	Performs a LU decomposition of a square matrix. ??

LVAR	
MAD	
MAIN	Displays the CAS menu, see **Menus**.
MANT	Returns the mantissa of a number. The result x is always in the range $10 < x \leq 0$. Operates on a Real number or a List of Real numbers.
MAP	Takes a List of objects and an expression or program and returns a List that contains the expression/program applied to all of the objects in the input List: `{"10" "20" "30"}` ◀◀`NUM`▶▶ `MAP` returns `{49 50 51}` which are the ASCII codes of the first characters of the Strings. • For some reason it is not possible to specify the name of a program as the 2nd argument. Assume that variable P contains the ◀◀`NUM`▶▶ program: `{"10" "20" "30"} 'P' MAP` returns `{NOVAL NOVAL NOVAL}`. It is unclear why. • Note that in general all commands that operate on a certain object type can also operate on a List of this type: `{"10" "20" "30"} NUM` also returns `{49 50 51}`. • But there are some slight differences: `{ 1 2 3 } 1 →LIST` converts stack level 2 into a size-1 list, returning `{{1 2 3}}`. But `{ 1 2 3 }` ◀◀`1 →LIST`▶▶ `MAP` applies the program to each element, returning `{ {1} {2} {3} }`. Also, some operators (ie. "+") have a non-standard behaviour when applied to Lists. See SEQ.
MATHS	Display the MATHS menu, see **Menus**.
MATR	Displays the CAS/MATR menu, see **Menus**.
MAX	Returns the larger of two numbers. Operates on a Real number or a List of Real numbers. See MIN.
MAXR	Reaturns 'MAXR'. When evaluated in approximate mode or converted to a number this results in 9.99999999999E499. See '∞' but different from MAXR '∞' immediately evaluates to a Real number in approximate mode.
MAXΣ	Finds the maximum value within each column of the ΣDAT statistics matrix and returns a vector. See Σ+.
MCALC	
MEAN	Calculates the mean values of all columns in the ΣDAT statistics matrix and returns a vector. See Σ+.
MEM	Returns the available amount of memory in bytes as a Real number.
MENU	Takes a Real number and displays the associated softmenu (each menu has a certain number assigned). Strange things happen when a Binary number is used as an argument.
MENUXY	
MERGE	
MIN	Returns the larger of two numbers.

	Operates on a Real number or a List of Real numbers. See MAX.
MINIFONT→	
MINIT	
MINR	Return 'MINR' which evaluates to the smallest real number, 1E-499
MINΣ	Finds the minimum value within each column of the ΣDAT statistics matrix and returns a vector. See Σ+.
MITM	
MKISOM	
MOD	Modulo function which returns the reminder after division: `8 3 MOD` returns 2, `-8 3 MOD` returns 1 because –3*3+1=–8. Operates on a Real number or a List of Real numbers.
MODSTO	Changes the CAS modulo setting to the specified Real number. Only works in exact mode. The modulo is stored in variable MODULO in the CASDIR subdirectory. The directory is created if it doesn't already exist.
MODULAR	Displays the MATH/MODULAR menu, see **Menus**.
MROOT	
MSGBOX	
MSLV	
MSOLVR	
MULTMOD	
MUSER	
NDIST	Returns the value of the Normal distribution: NDIST(m,sq,x) = exp[– (x–m)²/(2*sq)] / sqrt(2π*sq) Example: `10 2 12 NDIST` returns 0.10377687…
NDUPN	Duplicates stack level 2 n-1 times and leaves argument intact: `10 20 30 2 NDUPN` returns `10 20 30 30 2`. `10 20 30 0 NDUPN` returns `10 20 0`, effectively deleting level 2. The same happens for all arguments ≤0.
NEG	Negates a number. Operates on a Real number, a Complex number, a List of Real/Complex numbers, a real/complex Vector or Matrix. For a Binary number this calculates 2^64–X. See ABS and CONJ.
NEWOB	
NEXT	Used with FOR and START, see there.
NEXT	Not in catalog, located in the PRG/RUN softmenu. When debugging a program pressing the NEXT softkey displays the next two instructions on top of the display but does not execute these instructions. Does nothing if there is no halted program. See DBUG, SST, HALT.
NEXTPRIME	
NIP	Removes the element in stack level 2 and drops down higher levels. The contents of level 1 are unaffected.
NOT	NOT operator, see AND. Inverts the bits of a binary number.
NOVAL	Puts the NOVAL object onto the stack. It is unclear what this is needed for.
NSUB	
NUM	Takes a string and returns the ASCII code of its first character. If the

	string is empty 0 is returned. See CHR.
NUMX	
NUMY	
NΣ	Returns the number of columns of the statistical data in ΣDAT. This is the number of variables within on data sample.
OBJ→	The command splits all sorts of types into components: • For a Real or Binary number (or other "atomic" objects) or a program an error is produced. • A Complex number is split into real and imaginary part. • A real Vector is split into Real numbers and a list containing a single Real indicating the length of the Vector. • A real Matrix is split into Real numbers and a list containing two Reals indicating the matrix dimensions. • A List is split into components and a Real number indicating the length of the list. List elements are not split up. • A tagged Real is split into its value and the tag name which is returned as a string. • An expression is split as well: `'4+5' OBJ→` returns `4 5 2 +`. `'X*(Y+4)' OBJ→` returns `'X' 'Y+4' 2 *`. • A string is converted into a name: `"ABC" OBJ→` returns `'ABC'` but `"3Z/" OBJ→` causes an error.
OCT	Sets octal display format for Binary numbers. See BIN, DEC, HEX.
OFF	Turns the calculator off.
OLDPRT	
OPENIO	
OR	OR operator, see AND.
ORDER	Takes a List of variable names and arranges the softkey labels of the current directory in the order specified by the list. If the list contains only a subset of the existing variables these will be moved to the beginning of the menu. An error occurs if the list contains a name that doesn't refer to a variable in the current directory but the reordering will be executed until the errorneous name is encountered.
OVER	Pushes a copy of the object in stack level 2 onto the stack: `2 3 OVER` returns `2 3 2`.
P2C	
PA2B2	
PARAMETRIC	
PARITY	
PARSURFACE	
PARTFRAC	
PATH	Returns the current directory as a List of directory names, starting with HOME.
PCAR	
PCOEFF	
PCONTOUR	
PCOV	Returns the covariance of the most recent curve fit. A value of 1 indicates a perfect fit. See BESTFIT. This is the "Grundgesamtheitscovariance" ??

PDIM	
PERM	Permutations. Ie. A B PERM returns A! / (A – B)! This is the number of possibilities to select B elements from a group of A different elements where different sequences do count separately. See COMB.
PEVAL	
PGDIR	Takes a subdirectory name and removes it without any question even if it is not empty.
PICK	Takes a Real number from the stack and then pushes a copy of the object in stack level n onto the top of the stack. `10 20 30 40 3 PICK` returns `10 20 30 40 20`. See ROLL, ROLLD and ANS.
PICK3	Pushes a copy of the object in stack level 3 onto the top of the stack. Same as "3 PICK". See ROLL and ROLLD.
PICT	
PICTURE	
PINIT	
PIX?	
PIXOFF	
PIXON	
PKT	
PLOT	
PLOTADD	
PMAX	
PMIN	
PMINI	
POLAR	
PLOYNOMIAL	Displays the MATH/POLYNOMIAL menu, see **Menus**.
POP	
POS	• Search String in stack level 2 for string in stack level 1 and return the position of first occurence or 0 if not found. `"123 123" "23" POS` returns `2`. • Search the List in stack level 2 for the object in stack level 1. `{1 2 (4 5)} (4 5) POS` returns 3.
POTENTIAL	
POWEXPAND	
POWMOD	
PR1	Print object in stack level 1. ??
PREDV	
PREDX	
PREDY	
PREVAL	
PREVPRIME	
PRLCD	Print contents of the LCD screen. ??
PROMPT	
PROMPTSTO	
PROOT	
PROPFRAC	

PRST	Print entire stack. ??
PRSTC	
PRVAR	Print variables in current directory. ??
PSDEV	Calculates the standard deviation of all columns in the ΣDAT statistics matrix and returns a vector. See Σ+, SDEV. This is with regard to the ?? Grundgesamtheit.
PSI	
PTAYL	
PURGE	Takes a symbolic name and deletes the associated variable or directory. PURGE will cause an error if a non-empty directory is deleted.
PUSH	
PUT	Overwrite a matrix, vector or list element with a new value: `'M' {1 1} 5 PUT` set element (1,1) of matrix M to 5. `'VL' {2} 5 PUT` or `'VL' 2 5 PUT` sets the 2nd element of vector or list VL to 5. All arguments are dropped. But: `[[1 2][3 4]] {1 1} 8 PUT` returns `[[8 2][3 4]]` – in this case the target of the PUT operation is not dropped. See GET, PUTI.
PUTI	Similar to PUT but does not discard the source List/Matrix/Vector nor the index. Even more, the index is incremented. `'M' {1 1} 6 PUTI` returns `'M' {1 2}` and sets element (1,1) of matrix M to 6. The column index is incremented first and only when it wraps the row is incremented. Finally, the index wraps back to `{1 1}`. `'VL' {2} 6 PUTI` returns `'VL' {3}` and sets VL(2) to 6. `'VL' 2 PUTI` returns `'VL' 3` and sets VL(2) to 6. Once the index reaches the maximum value it wraps back to 1. See PUT, GETI.
PVAR	Calculates the variance of all columns in the ΣDAT statistics matrix and returns a vector. See Σ+, VAR. This is with regard to the ?? Grundgesamtheit.
PVARS	
PVIEW	
PWRFIT	Sets the curve fitting model to "power fit" and fits a curve into the current statistics data in ΣDAT. The resulting curve is of the form y=a+x^b. See BESTFIT.
PX→C	The reverse operation of C→PX, see there.
PSI	Defined a the n-th derivative of the Digamma function: PSI(x,n) = dn/dx Psi(x). I've verified that this is correct for n=1. `10.5 2 PSI` returns `-9.975E-3`. The 2nd argument must be integer but Binary numbers are not allowed. See GAMMA and Psi.
Psi	Not in the catalog but accessible thru the MTH/SPECI menu and also programmable. Digamma function: psi(x) = d/dx ln[Gamma(x)]. It is defined for Real and Complex numbers. See GAMMA and PSI.
QR	

QUAD	
QUOT	Returns the Euclidean quotient of two polynomials. `'X^2+2*X+1' 'X' QUOT` returns `'X+2'`. See ABCUV for general CAS issues.
QUOTE	
QXA	Expresses a quadratic form in matrix form. `'X^2+2*X*Y' ['X' 'Y'] XQA` returns `[[1 1][1 0]] ['X' 'Y']`. See ABCUV for general CAS issues.
RAD	Selects degress (360) for trigonometric calculations, see DEG, GRAD.
RAND	Return a Real random number in the range 0≤X<1. See RDZ.
RANK	
RANM	Takes a matrix or a size-2 list containing matrix dimensions from the stack and creates a matrix of matching dimensions with randomly chosen Real numbers for its elements. The Real numbers are integer values in the range –9 to 9.
RATIO	
RCEQ	Retrieve the value of the reserved variable EQ. Causes an error if it doesn't exist. See STEQ, RCL.
RCI	
RCIJ	
RCL	Takes a name from the stack and returns the value of the variable with this name. Causes an error if the variable doesn't exist. See STO.
RCLALARM	
RCLF	Returns a list with four 64-bit values which represent the current settings of all flags: • Flags –64 to –1, system flags. • Flags 65 to 1, user flags. • Flags –128 to –65, system flags. • Flags 128 to 65, user flags. Ie. user flag 1 is located in bit0 of the 2nd word of the list. Together with STOF it is possible to save the system state and restore it after some calculations. See STOF.
RCLKEYS	
RCLMENU	
RCLVX	
RCLΣ	
RCWS	Return the current Binary number word size as an integer (1-64). See STWS.
RDM	Re-dimensions a Matrix or Vector. Vectors can be converted to Matrices and vice versa. Existing Vector/Matrix values will be discarded or zeros appended as necessary. Takes a Matrix/Vector or a corresponding variable name and a List containing one or two Real numbers specifying the new dimension(s). `[1 2 3 4 5] {2 3} RDM` returns `[[1 2 3][4 5 0]]`. `[[1 2][3 4]] {3} RDM` returns `[1 2 3]`.
RDZ	Initialize the random number generator with a Real number seed. For identical seeds the RAND functions returns identical sequences of

	random numbers. Except, if the argument is 0 then the current time is used to initialize the random number generator. See RAND.
RE	Return the real part of an object. Object can be a Real or Complex number, a real or complex Vector or Matrx, a List of Real or Complex numbers. See RE. Ie. `(1 2) RE` returns `1`.
RECN	
RECT	Selects rectangular display mode for Complex numbers and 2-dim or 3-dim Vectors. See MODE menu in **Menus**. See SPHERE and CYLIN.
RECV	
REF	
REMAINDER	
RENAME	
REORDER	
REPEAT	Used with WHILE, see there.
REPL	Relaces elements of a string list matrix vector?
RES	
RESTORE	
RESULTANT	
REVLIST	Returns a list with elements in reversed order. Mixed-type list elements are allowed.
REWRITE	Displays the CAS/REWRITE menu, see **Menus**.
RISCH	
RKF	
RKFERR	
RKFSTEP	
RL	Rotate Binary number one bit left. The topmost bit (depending on the word size) is rotated into bit0.
RLB	Rotate Binary number one byte left. The topmost byte (depending on the word size) is rotated into the lower 4 bits. Example at a word size of 10 bit and bin mode: `#1010100101 RLB` returns `#0110101001`. The low-byte (10100101) is shifted up one byte and then truncated to 01 because it doesn't fit into the word. The higher 8 bits (10101001) become the low-byte.
RND	Round number in stack level 1 to the number of fractional digits in stack level 1: `1.235 2 RND` returns `1.24`. The number to be rounded can be a Real or Complex number, Vector or Matrix or a List with any of these types. The number of digits must be a Real number. See TRNC.
RNRM	
ROLL	Takes a Real number n from the stack and then moves the n-th stack element onto the top of the stack: `10 20 30 40 3 ROLL` returns `10 30 40 20`. See PICK.
ROLLD	Takes a Real number n from the stack and then moves stack element 1 to the given stack position: `10 30 40 20 3 ROLL` returns `10 20 30 40`. The is the reverse operation of ROLL. See PICK.

ROMUPLOAD	In earlier versions used to upload the contents of the flash ROM thru the serial port. When used on the HP-49G+ a "Not Available" error occurs.
ROOT	
ROT	Rotates up the topmost 3 stack objects: `1 2 3 ROT` returns `2 3 1`. See UNROT and PRG/STACK menu in section **Menus**.
ROW+	
ROW–	
ROW→	
RPL>	
RR	Rotate Binary number one bit right. Bit0 is rotated into the topmost bit.
RRB	Rotate Binary number one byte right. Bit0-7 are rotated into the topmost 8 bits. See RLB.
RREF	
RREFMOD	
RRK	
RRKSTEP	
RSBERR	
RSD	
RSWP	
RULES	
R→B	Convert a Real number to a Binary number. The Real number is rounded to integer before conversion. Negative reals are always converted to 0. Very large reals are converted to the maximum Binary number 2^64-1. See B→R.
R→C	Convert two numbers into a Complex object: • Two Real numbers yield a Complex number: `1 2 R→C` returns `(1 2)` • Two real Vectors of same size are combined to a complex Vector: `[1 2] [3 4] R→C` returns `[(1 3) (2 4)]` • Two real Matrices of equal dimensions are combined to a complex Matrix. • Two complex Vectors are combined to a complex Vector: `[(1 2) (3 4)] [(5 6) (7 8)] R→C` returns `[(1 5) (2 6)]`. `[(1 2) (3 4) (5 6)] [(7 8) (9 10) (11 12)] R→C` returns `[(1 7) (2 8) (3 9)]`. • Two complex Matrices are combined to a complex Matrix. These complex-to-complex conversions are a bit odd… See C→R.
R→D	Converts a Real number or a list of Real numbers from radians (2π) to degrees (360). Note that this independent of the current angle mode. See D→R.
R→I	Converts an integer Real number to an exact Real number. Causes an error if the argument has a fractional part. See I→R.
SAME	Similar to the '==' operator. However, SAME never evaluates its

	arguments and does never return a symbolic expression. Also, it can compare arbitrary types: `'A' 'B' SAME` returns 0. `'A' 'A' SAME` returns 1. `'A' 'B' ==` returns `'A==B'`. `'A' 'A' ==` returns 1. `'SQ(2)' '4'` SAME returns 0. `'SQ(2)' '4' ==` returns 1. ◀◀ `6 *` ▶▶ `[1 2] SAME` returns 0. ◀◀ `6 *` ▶▶ `[1 2] ==` returns 0.
SBRK	
SCALE	
SCALEH	
SCALEW	
SCATRPLOT	
SCATTER	
SCHUR	
SCI	Selects the scientific (eponential) display format, see ENG.
SCLΣ	
SCONJ	Takes a name and conjugates (negates the imaginary part of) the value of the variable with this name. `(1 2) 'X' STO 'X' SCONJ` results in variable X being set to (1 –2). See STO*.
SCROLL	
SDEV	Calculates the standard deviation of all columns in the ΣDAT statistics matrix and returns a vector. See Σ+.
SEND	
SEQ	Creates a List by evaluating an expression or a program multiple times: `'SQ(X)' 'X' 3 6 1 SEQ` returns `{9 16 25 36}`: The first value of X is 3 and incremented by 1 until it reaches 6. For each value of X the formula 'SQ(X)' is evaluated to generate the List values. Alternatively, the example could be written using a program: ◀◀ `X SQ` ▶▶ `'X' 3 6 1 SEQ`. The parameters for SEQ are: Expression/Program, Variable, Start, End, Increment. See MAP.
SERIES	
SERVER	
SEVAL	
SF	Set Flag(s). The flag number must be given as a Real number or a list of Real numbers. See CF.
SHOW	
SIDENS	Calculates the density of Silicium in 1/cm³ at a given temperature (in Kelvin).
SIGMA	
SIGMAVX	
SIGN	Returns the sign of a Real number as +1 or –1.

	For a complex number a unity-length vector in the direction of the complex number vector is returned.
SIGNTAB	
SIMP2	
SIMPLIFY	
SIN	Sine, see ACOS.
SINCOS	
SINH	Hyperbolic sine, see ACOS.
SINV	Takes a name and inverts the value of the variable with this name. `8 'X' STO 'X' SINV` results in variable X being set to 0.125. See STO*.
SIZE	Returns the size of the object in stack level 1 as a Real number. • Real, Complex and Binary number: Returns 1. • List: Returns number of list elements or 0 for an empty list. • String: Returns string length or 0 for an empty string. • Program: Returns 1. • Vector and matrix: Returns a list containing the dimensions of the vector/matrix. Ie. for `[[1 2 3][4 5 6]]` this returns `{2 3}`. • Algebraic expression: Number of elements. Note that a multi-character variable name counts as a single element. Ie. for `'XX+17'` this returns `3`. • Real number with a unit: Returns 2 plus the number of characters in the unit. Ie. `123_1/kg^2` returns 9 because there are 7 characters in "_1/kg^2".
SL	Shift Binary number one bit left. Set bit0 to 0 and discard the topmost bit.
SLB	Shift Binary number one byte left. Set bit0-7 to 0 and discard the topmost 8 bits.
SLOPERFIELD	
SNEG	Takes a name and negates the value of the variable with this name. `8 'X' STO 'X' SNEG` results in variable X being set to –8. See STO*.
SNRM	
SOLVE	
SOLVER	Displays the CAS/SOLVER menu, see **Menus**.
SOLVEVX	
SORT	Takes a list and sorts in in ascending order. Elements can be Real numbers, Binary numbers or Strings. Mixed types are not allowed. See "<" operator, REVLIST and xLIST commands.
SPHERE	Selects spherical display mode for 3-dim Vectors and polar format for 2-dim Vectors and complex numbers. See MODE menu in **Menus**. See RECT and CYLIN.
SQ	Square of a Real or Complex number or real/complex quare matrix. When applied to a Real number with unit then the unit is squared as well, ie. `2_m` squares to `4_m^2`.
SR	Shift Binary number one bit right. Set topmost bit to 0 and discard bit0.
SRAD	

SRB	Shift Binary number one byte right. Set topmost 8 bits to 0 and discard bit0-7.
SRECV	
SREPL	
SST	Not in the catalog. Single-steps thru a program but does not branch into subprograms. The previousely executed command is displayed in the top left corner of the LCD display. See DBUG.
SST↓	Not in the catalog. Single-steps thru a program and does step into subprograms. See DBUG and SST.
START	Syntax 1: `start end START code NEXT` • Performs code end-start+1 times. • start and end may be omitted if suitable values already exist on the stack. • start and end must be Real or Binary but not in mixed mode. • If start>=end the code is executed once. • The current loop counter is not accessible to the program, see FOR. Syntax 2: `start end START code step-size STEP` • Here the internal loop counter is encremented by step-size. • The loop stops if the internal loop counter is ≥end.
STD	Selects the standard display format, see MODE menu in section **Menus**.
STEP	Used with FOR and START, see there.
STEQ	Store an algebraic expression in the reserved variable EQ which is used for many purposes (plotting, root finding, solver, etc.). See STO, RCEQ.
STIME	
STO	Takes an object and a name from the stack and stores the object in a named variable. `123 'X' STO` stores 123 in the variable named X. Both arguments are removed from the stack. See RCL and ▸.
STO*	Takes an object and a name from the stack and multiplies the contents of the named variable with the object. Note that this type of variable multiplication is more restricted with regard to object types than the regular multiplication. Ie. it is not possible to multiply a Binary number with a Real number or perform a symbolic multiplication. This restriction applies to many other register operations. `123 'X' STO*` multiplies the contents of variable X with 123. Both arguments are removed from the stack. See STO+, STO–, STO/, SNEG, SINV, SCONJ, INCR, DECR.
STO+	Variable addition. See STO*.
STO–	Variable substraction. See STO*.
STO/	Variable division. See STO*.
STOALARM	Creates a new alarm from parameters specified in a list. `{6.022004 18.15 "Movie" 1} STOALARM` creates a new alarm: • Its index is 1. The index number is apparently not used internally. • It will occur on on 6.2.2004 18:15.

	• When it alarm occurs it outputs the string "Movie". Alternatively, expressions or program names can be stored there. • In general, when an alarm occurs it output its number (starting from 1) and performs the desired action. If the action is a string the calculator will display information about the alarm an beep a few times. • The time must be given in 24-hour format. • The STOALARM returns the number of the newly created alarm. • Apparently, the number of alarms is unlimited. See STOALARM, RCLALARM, DELALARM, FINDALARM, ACK, ACKALL, menu TIME in section **Menus**.
STOF	Set 128 user and 128 system flags from four 64 bit values stored in a list. See RCLF.
STOKEYS	
STORE	
STOVX	
STOΣ	
STREAM	
STR→	Splits a string into objects and evaluates them. `"3 4 + 10 *" STR→` returns `70`. This is essentially what ENTER does to the command line when pressed in edit mode. See →STR.
STURM	
STURMAB	
STWS	Sets the Binary number word size to the number of bits specified in stack level 1: • Argument can be a Real or Binary number. • Argument can be a list of real numbers. In this case the rightmost value specifies the word size. • Real numbers are rounded to integers first, ie. 3.5 selects a word size of 4. • Values <1 select a word size of 1 bit, values >64 a word size of 64 bits. • Use RCWS to recall the current word size.
SUB	• Get a substring from a String: `"12345678" 2 4 SUB` returns `"234"`. If start-index > end-index an empty string is returned. Indices start from 1. Indices <1 are treated as 1. Indices larger than the number of elements are treated as the number of elements. • Get a sublist from a List in a similar way. • Get a subvector from a Vector. If start-index > end-index the indices are exchanged. Out-of-bound indices cause an error. • Depending on the type of index SUB can return a submatrix from a given Matrix: `[[1 2 3][4 5 6][7 8 9]] {1 2} {2 3} SUB` returns `[[2 3][5 6]]`. This two List specify the top-left and bottom-right values of the returned Matrix.

	If plain Reals are given instead of Lists the values are interpreted as index values into row 1.
SUBST	
SUBTMOD	
SVD	
SVL	
SWAP	Exchange stack level 1 and 2. In idle mode the right-arrow key performs the same operation.
SYLVESTER	
SYSEVAL	
SYST2MAT	
TABVAL	
TABVAR	
TAIL	Removes the first element of a List and returns the remaining List. If the last element has been removed an empty List is returned. Does not do anything on an empty List. Removes the first character of a string and returns the remaining String. See HEAD.
TAN	Tangent, see ACOS.
TAN2CS2	
TAN2SC	
TAN2SC2	
TANH	Hyperbolic tangent, see ACOS.
TAYLOR0	
TAYLR	
TCHEBYCHEFF	Takes an integer Real number n and returns the Tchebycheff polynomial of the 1st kind of order n for n>0 and the Tchebycheff polynomial of the 2nd kind of order n for n<0. `3 TCHEBYCHEFF` returns `'4*X^3-3*X'`. `-2 TCHEBYCHEFF` returns `'2*X'`.
TCOLLECT	
TDELTA	Calculates temperature difference between two values. Mainly used if the given temperature values include units. The returned value will have the same units as the first argument: `25_°F 52_°C TDELTA` returns `-100.6_°F`. The input values must either have both temperature units or no units at all. See TINC.
TESTS	Displays the MATH/TESTS menu, see **Menus**.
TEVAL	
TEXPAND	
TEXT	
THEN	Used with IF, see there.
TICKS	Return the number of system ticks since 00:00 of Jan 1st, year 0 as a binary number. A system tick is 1/8192th second.
TIME	Return the current time as a Real in the form `h.mmssffffff`. f are fractional seconds.
TINC	Adds temperatures. `100_K 10_°C TINC` returns `110_K`. See TDELTA.

TLIN	
TLINE	
TMENU	
TOT	Calculates the sum of all values in each column of the ΣDAT statistics matrix and returns a vector. See Σ+.
TRACE	
TRAN	Transposes a Matrix: `[[1 2][3 4]]` TRANS results in `[[1 3][2 4]]`. `[[1 2 3][4 5 6]] TRAN` results in `[[1 4][2 5][3 6]]`. Not allowed for vectors. See TRN.
TRANSIO	
TRIG	
TRIGCOS	
TRIGO	Displays the CAS/TRIGO menu, see **Menus**.
TRIGSIN	
TRIGTAN	
TRN	This is a combination of TRAN (transposition) and CONJ (conjugation) of a Matrix. See there.
TRNC	Truncate number in stack level 1 to the number of fractional digits in stack level 1: `1.235 2 TRNC` returns `1.23`. The number to be truncated can be a Real or Complex number, Vector or Matrix or a List with any of these types. The number of digits must be a Real number. See RND.
TRUNC	
TRUTH	
TSIMP	
TSTR	Takes a date in the form d.mmyyyy and a time in the form h.mmssf and returns a textual string representing the date and time. `DATE TIME TSTR` may result in `"SAT 10.01.04 17:56:07"`. Note that MODE settings for the data and time display determine the output format. See DATE, TIME.
TVARS	Apparently takes a Real number and returns a List of variable names of the current directory which match the given data type. Ie. if the current directory contains variables S containing a string then `2 TVARS` returns { S }. If there is no variable of this type then an empty list is returned.
TVM	
TVMBEG	
TVMEND	
TVMROOT	
TYPE	Return type of object in stack level 1 as a Real. See **Data Types**.
UBASE	Converts a Real number with units to a corresponding number expressed in SI base units: `125_lb/ft UBASE` returns `186.020…_kg/m`. If the input value has no attached units it is left unchanged. See CONVERT.
UFACT	See UNITS/TOOLS menu. Completely mysterious what this does.
UFL1→MINIF	

UNASSIGN	
UNASSUME	Remove all assumptions about a variable. ??
UNBIND	
UNDO	Not in the catalog. Reverses the most recent action. Available for stack operations and some other edit operations (ie. in the equation writer EQW). See LASTARG.
UNPICK	Take index N and number X from the stack, then replaces stack level N with the number X: `10 20 30 40 2` returns `10 40 30`. `10 20 30 40 1` returns `10 20 40`. `10 20 30 40 0` returns `10 20 30`, effectively discarding stack level 1. The same occurs for all index values ≤0.
UNROT	Reverse ROT operation: Rotate lower 3 stack elements down in a circular manner: `1 2 3 4 UNROT` returns `1 4 2 3`.
UNTIL	Used with DO, see there.
UPDIR	Change the current directory to one level up the directory hierarchy.
UTPC	Upper tail probability of Chi-square distribution c(n,x): c(n,x) = x^(n/2 – 1) * exp(-x/2) / [2^(n/2) * Γ(n/2)] UTPC(n,x) = ∫ [x, ∞] c(n,p) dp
UTPF	Upper tail probability of f-distribution f(n,d,x): f(n,d,x) = [Γ((n+d)/2) * (n/d)^(n/2) * x^(n/2 – 1)] / [Γ(n/2) * Γ(d/2) * (1 – n*x/d)^((n+d)/2)] UTPF(n,d,x) = ∫ [x, ∞] f(n,d,p) dp
UTPN	Upper tail probability of normal distribution N(s,m,x): N(sq,m,x) = exp[– (x–m)²/(2*s)] / sqrt(2π*sq) UTPN(sq,m,x) = ∫ [x, ∞] N(sq,m,p) dp
UTPT	Upper tail probability of Student's t-distribution t(n,x): t(n,x) = Γ((n+1)/2) / [Γ(n/2)*sqrt(π*n)] * (1+x²/n)^(-(n+1)/2) UTPT(n,x) = ∫ [x, ∞] t(n,p) dp
UVAL	Extracts the plain Real number from a number that has an attached unit. `125_m UVAL` returns 125. If the input values doesn't have an attached unit it is left unchanged.
VANDERMONDE	Takes a List or a Vector of values and creates a Vandermonde Matrix containing n-th powers of the list numbers in column n: `{2 4 6} VANDERMONDE` returns `[[1 2 4][1 4 16][1 6 36]]`.
VAR	Calculates the variance of all columns in the ΣDAT statistics matrix and returns a vector. See Σ+.
VARS	Returns a list of all the variable and directory names of the current directory.
VER	Displays the version number of the CAS system by returning a Real number: `4.20031005` which is Version 4 dated 5.10.2003.

VERSION	Displays the version of the operating system by returning two strings:
	`"Version HP49-C Revision #1.23"`
	`"Copyright HP 2003"`

VISIT	
VISITB	
VPOTENTIAL	

VTYPE	
V→	Split a Vector into into individual components: `[1 2 3] V→` returns `1 2 3` on the stack. Matrices are not allowed. See →V2 and →V3.
WAIT	Takes a Real number and suspends program execution for the given amount of seconds. Strange things happen when the argument is <0.
WHILE	Syntax: `WHILE test REPEAT code END` • Performs code while test returns a non-0 value. • The tested value must be a Real. • If test initially evaluates to false the code is never executed. • The test clause may be omitted if a suitable value is already present on the stack. • The code clause may be omitted.
WIREFRAME	
WSLOG	
XCOL	
XGET	
XMIT	
XNUM	
XOR	XOR operator, see AND.
XPON	Returns the exponent of a Real number: `12345 EXPON` returns 4.
XPUT	
XQ	
XRECV	
XRNG	
XROOT	Calculate n-th root of a value: `8 3 XROOT` returns 2. Defined for Real and Complex numbers.
XSEND	
XSERV	
XVOL	
XXRNG	
YCOL	
YRNG	
YSLICE	
YVOL	
YYRNG	
ZEROS	
ZFACTOR	Calculates a correction factor for the compressibility number of non-ideal carbo-hydratic gases. Takes as arguments the reduced temperature (quotient of current temperature and pseudo-critical temperature of the gas) and the reduced pressure (quotient of current pressure and pseudo-critical pressure). Return value is a Real number.
ZVAL	
^	Exponential operator: `2 3 ^` returns 8.
_	The underscore is used to append units to numbers, ie. `125_m`.
dB	

e	Returns `'e'` and when evaluated 2.71828…
gmol	
i	Equivalent to the complex number `(0,1)`.
lbmol	
lim	
qr	
rpm	
rref	
|	
√	Sqaure root of a Real or Complex number.
∫	
Σ	
Σ+	Take a Real or Complex number, Vector or Matrix and adds the values to the statistics matrix ΣDAT. If it doesn't exist ΣDAT is created in the current directory. New data is appended as a new row of values at the bottom of the ΣDAT matrix. When ΣDAT exists newly added data must contain the same number of colums. In essence the rows are the different data samples and the colunms are different entities within one sample. If subsequent data samples are given as a List the following program converts this list into a column vector that can be appended to the ΣDAT statistics matrix: ◀◀ OBJ→ 1 2 →LIST →ARRY ▶▶. This program executed on `{1 2 3 4}` returns `[[1][2][3][4]]`.
Σ–	Removes the last data sample (=the bottom row) from the ΣDAT matrix and returns it to the stack.
ΣLINE	Returns an algebraic expression that represents the most recently best-fit curve calculated for the statistics data in ΣDAT. See BESTFIT, EXPFIT, LINFIT, LOGFIT, PWRFIT and ΣPAR. If ΣPAR doesn't exist it is created automatically.
ΣLIST	Takes a list of type S from the stack and add up all elements of the list: `{2 4 6}` `ΣLIST` returns `12`. Mixed-type list elements are allowed as long as the addition is defined for subsequent elements: `{2 (3 4) 5}` `ΣLIST` returns `(10 4)`. See ΠLIST and ΔLIST.
ΣX	Returns the sum of the X-values of ΣDAT. The ΣPAR variable (see **Variables**) which column of ΣDAT corresponds to the X- and Y-values.
ΣX2	Returns the sum of the squared X-values of ΣDAT. See ΣX.
ΣXY	Returns the sum of the X*Y-products of ΣDAT. See ΣX.
ΣY	Returns the sum of the Y-values of ΣDAT. See ΣX.
ΣY2	Returns the sum of the squared Y-values of ΣDAT. See ΣX.
▸	Takes a name from the stack and stores the object in stack level 2 in the specified name. Different from the STO command this does not removes the stored variable from the stack! See STO.
π	Puts the value of Pi (3.141592…) onto the stack either in exact or approoximate mode.
∂	

≤	Comparisn operator, return 0 or 1, see "<".
≥	Comparisn operator, return 0 or 1, see "<".
≠	Comparisn operator, return 0 or 1, see "==".
→	Local variable operator. ◀◀ → X Y ◀◀ X 3 * Y + ▶▶ ▶▶ ◀◀ → X Y 'X*3+Y' ▶▶ Both programs takes two arguments from the stack and store them in local variables called X and Y. Note that X and Y are only visible inside the program and do not conflict with global variables of the same name. Then the program calculates a return value in RPN or algebraic mode.
→ARRY	Reverse ARRY→ command, see there. Array arguments may be Real or Complex. If one of the numbers is complex the entire matrix or vector will be complex.
→COL	
→DATE	Set internal clock to the specified date in `d.mmyyyy` format.
→DIAG	Takes a matrix in stack level 1 and returns its diagonal elements in form of a vector. Does not accept a variable name that refers to a matrix. If the matrix is not quadratic then exceeding rows or columns are not considered. See DIAG→.
→FONT	
→GROB	
→HEADER	
→HMS	Convert a Real number in fractional hour format into h.mmssf format. See HMS+, HMS→ and other HMSx functions.
→KEYTIME	
→LANGUAGE	
→LCD	Takes a graphics object and displays it in the top left corner of the LCD display. ?? See LCD→.
→LIST	Takes a Real number n from the stack and then combines the next n stack objects to a list: `10 20 30 3 →LIST` returns `{10 20 30}`. `0 →LIST` returns the empty list `{ }`.
→MINIFONT	
→NDISP	
→NUM	Tries to convert the object in stack level 1 into a plain number. Variable names are replaced by their contents, constants are replaced by their values. If a symbolic name does not correspond to a variable or a program the name is left unchanged.
→Q	
→Qπ	
→ROW	
→STR	Convert object in stack level 1 into string. • The display format is preserved. • No additional quotes are put around strings. • Symbolic names or programs are not evaluated. • If the multi-line display format is enabled then CR-LFs will be inserted in the string as well.

	See STR→.
→TAG	Combine a quoted name or a string in stack level 1 and an object in stack level 2 to a tagged object: `125 'X' →TAG` returns `X:125`. `'X' 125 →TAG` returns `125:X`. `10 "RES" →TAG` returns `RES:10`. Multiple tags are possible: `125 'X' →TAG 'Y' →TAG` returns `Y:X:125`. `125 'X' 'Y' →TAG →TAG` returns `X:125`. Tags can be copied over to another value: `'Y' X:125 →TAG` returns `125:Y`. See OBJ→.
→TIME	Set internal clock to the specified time in h.mmss format.
→UNIT	Takes a Real number as input value 1 and combines it with the units of input value 2: `125 48_s →UNIT` returns `125_s`. Input value 1 must not have any attached units and input value 2 must have attached units, otherwise an error occurs.
→V2	Combines two Real number arguments into a 2-dim vector. Can also operate on equally-length lists containing Real numbers returning a list containing 2-dim vectors. See →V3 and V→.
→V3	Same as →V2 but combines three Real numbers into a 3-dim vector. See →V2 and V→.
↓MATCH	
↑MATCH	
ΔLIST	Takes a list of type S from the stack and returns a size S-1 list containing the differences of subsequent elements: `{1 2 4} ΔLIST` returns `{1 2}`. Mixed-type list elements are allowed as long as the minus-operator is defined, see there and ΣLIST and ΠLIST.
Π LIST	Takes a list of type S from the stack and multiplies out all elements of the list: `{2 4 6} ΠLIST` returns `48`. Mixed-type list elements are allowed as long as the multiplication-operator is defined: `{2 (3 4) [5 6]} ΠLIST` returns `[(30 40) (36 48)]`. Note that the "Π" in the command name is an upper case Pi that can be reached by ALPHA shift-right P. See ΠLIST and ΣLIST.
∞	The infinite number. In exact mode this pushes the symbol '∞' onto the stack. In approximate the result is 9.9999999999E499. See MAXR.
◀◀	Program delimiter.
▶▶	Program delimiter.

EVAL: Converts a list into individual elements!!!

What are the menus?
What are the keyboard shortcuts for editing, cursor movement, variable manipulation, unit conversions etc.?
Flag 65: Level 1 multipline

In "pretty print" mode (flag 52) use selected font for equation display in stack level 1.

↖ ↗ ← ↑ → ↓ ↔ ⇦ ↵ ◀◀ ▶▶
α δ π ∂ θ λ φ μ
Δ Ω Π Σ
≤ ≥ ± • ÷ ≠ √ ∫ ¡ ¿
∞