HP 49G Pocket Guide

Contents

Version 2.0

hp HEWLETT® PACKARD

Quick Reference Chart

Tool	Access
Alarms	↱ TIME
Algebra	↱ ALG
Arithmetic	↰ ARITH
Calculus	↰ CALC
CAS modes	MODE CAS
Characters	↱ CHARS
Command Catalog	CAT
Complex Numbers	↱ CMPLX
Constants	APPS CONSTANTS LIB
Conversions	↰ CONVERT
Display	MODE DISP
Editing Tools	TOOL
Equation Writer	↰ EQW
File Manager	↰ FILES
Flags	MODE FLAGS
Libraries	↱ LIB
Math	↰ MTH
Matrix Writer	↰ MATRICES
Plotting	APPS PLOT FUNCTIONS
Printing	APPS I/O FUNCTIONS
Programming	↰ PRG
Solve, Financial	↰ FINANCE
Solve, Numeric	↱ NUM.SLV
Solve, Symbolic	↰ S.SLV
Statistics	↱ STAT
Tables	↰ TBLSET, ↰ TABLE
Transfer Data	APPS I/O FUNCTIONS
Trigonometry	↱ TRIG
Variables	VAR

2 Function Key Guide

This section explains the use of each item on the function key menu of the more commonly used HP 49G applications.

Equation Writer

EDIT	Opens the selected component in the command line editor. Make your changes, then press ENTER to return to Equation Writer.
CURS	Enables cursor mode. Use the arrow keys to enclose the part of the equation that you want to select in a box, then press ENTER to return to selection mode, with the boxed component selected.
BIG	Toggles Equation Writer between standard font and mini-font.
EVAL	Evaluates the selection. Equivalent to pressing ↱ EVAL.
FACTO	Applies the FACTOR command to the selection.
TEXPA	Applies the TEXPAND command to the selection.

File Manager

EDIT	Opens the selected object. If the object can be edited, it is opened in the command line editor.
COPY	Copies the selected object. After you press COPY, select the destination directory, and press OK to paste the object.
MOVE	Moves the selected object. After you press move, select the destination directory, and press OK to move the object to the directory.
RCL	Copies the selected object to the command line.
EVAL	Evaluates the selected object.
TREE	Returns to the File Manager opening screen, showing the ports and the HOME directory.
PURGE	Deletes the selected object or objects.
RENAM	Renames an object. The calculator prompts for a new name for the selected object.
NEW	Opens the New Variable input form, used to create a new variable or directory.
ORDER	When you select multiple objects (using ENTER) places the selected objects in the order in which you selected them.
SEND	Sends the selected object or objects to another calculator.
RECV	Receives objects sent from another calculator.
HALT	Suspends your File Manager session. You can return to the session by pressing ↰ ON.
VIEW	Displays the contents of the currently selected object. You cannot edit the contents.
EDITB	Opens the currently selected object in the most suitable editor.
HEADE	Toggles the File Manager header between memory and selection details, and path and content details.
LIST	Hides or shows the details of listed objects.

Continued

Stack

ECHO	Press ECHO, then ENTER to copy the contents of the current level to the command line. Edit the contents on the command line, and press ENTER to place them on level 1 of the stack.
VIEW	Displays the contents of the current level in textbook mode.
EDIT	Opens the contents of the current level in the most appropriate editor, ready for editing.
INFO	Displays information about the object at the current level, including its size in bytes.
PICK	Copies the contents of the current level to stack level 1. All existing objects are pushed up one level.
ROLL	Moves the contents of the current level to level 1. The portion of the stack below the current level is rolled up.
ROLLD	Moves the contents of level 1 to the current level. The portion of the stack beneath the current level is rolled down.
→LIST	Creates a list that contains the stack objects from 1 to the current level. The newly created list is placed on level 1 of the stack, and the original objects are removed.
DUPN	Duplicates the levels from the currently selected level to level 1, and pushes up the existing levels to accommodate the duplicated levels.
DROPN	Deletes all levels below the selected level.
KEEP	Deletes all levels above the selected level.
GOTO	Prompts for a stack level to select, then selects the level number that you enter.
LEVEL	Copies the current level number to level 1 of the stack.

Matrix Writer

EDIT	Places the contents of the currently selected cell on the command line, ready for editing.
VEC	For single-row matrices, sets that the row of values is a vector rather than a matrix. That is, when you place it on the command line, it is enclosed in a single pair of square brackets rather than two pairs.
←WID	Reduces the width of the columns.
WID→	Increases the width of the columns.
GO→	Sets that the cursor moves to the left by default when you enter data.
GO↓	Sets that the cursor moves down by default when you enter data.
+ROW	Adds a row filled with zeros at the cursor position
–ROW	Deletes the row at the cursor position.
+COL	Adds a column filled with zeros at the cursor position.
–COL	Deletes the column at the cursor position.
→STK	Copies the selected element only to the stack or the command line.
GOTO	Displays an input form that allows you to specify the column and row coordinates to select.
DEL	Fills a selected range with zeros.

Graphics Editor

DOT+	Turns on pixels beneath the cursor.
DOT–	Turns off pixels beneath the cursor.
LINE	Draws a line from a marked point to the cursor. (Press $\boxed{X}$ or MARK to mark a point).
TLINE	Same as LINE but toggles pixels on or off.
BOX	Draws a rectangle from a marked point to the cursor.
CIRCL	Draws a circle around a marked point with a radius indicated by the position of the cursor.
MARK	Marks a point. Same as pressing $\boxed{X}$.
+/–	Inverts the cursor when it crosses an object.
LABEL	Displays axes labels.
DEL	Deletes that part of the graphic bounded by a rectangle from a marked point to the cursor.
ERASE	Erases the entire graphic.
MENU	Hides the function-key menu. (Press $\boxed{F6}$, $\boxed{+}$, or $\boxed{-}$ to redisplay the menu.)
SUB	Copies to the stack that part of the graphic bounded by the rectangle from a marked point to the cursor
REPL	Pastes what was last copied with SUB.
PICT→	Copies the graphic to the stack.
X,Y→	Copies the cursor coordinates to the stack.
PICT	Replaces the edit menu with the picture menu.

❸ Reserved Names and Constants

You should avoid using certain names for variables, because their contents are interpreted by the calculator in set ways. Some examples are given in the following table.

Name	Use
0DETYPE	The differential equation type used in the DESOLVE command.
ALRMDAT	Data for current alarms.
CST	Current contents of a custom menu.
d#	Indicates a user-defined derivative, where # is the number of the defined derivative.
EPS	The smallest real value below which the calculator rounds to zero for some operations, for example EPSX0.
EQ	Current equation, plotting and numeric solving.
ERABLEMSG	Information relating to unevaluated integrations.
EXITED	If this variable contains a program, the program runs whenever the command line editor session is ended.
EXPR	Current expression, symbolic operations.
IERR	Uncertainty in current integration.
IOPAR	Current parameters for I/O operations.
MODULO	The value of the current modulo setting.
n1, n2,	Integer coefficients used by ISOL.
PPAR	Current parameters for plotting.
PRTPAR	Current parameters for printing.
s1, s2,	Sign coefficients used by ISOL and QUAD.
ΣDAT	Current matrix of data used for statistics.
ΣPAR	Parameters for statistics calculations.
PRIMIT	The last computed antiderivative.
REALASUME	A list of variables that the computer algebra system assumes are real values.
STARTED	If this variable contains a program, the program runs whenever the command line editor session is started with EDIT EDITB, VISIT, VISITB, or ▼ in RPN mode.
STARTERR	Used to customize error message displays.
STARTEQW	Used to apply a customized operation to a selected component in Equation Writer.
STARTOFF	If this variable contains a program, the program runs whenever the calculator turns off automatically.
STARTUP	If this variable contains a program, the program runs after a warm start.
TOFF	Sets the number of ticks before the calculator automatically turns off.
TPAR	Current parameters for viewing tables.
VPAR	Current parameters for viewing 3-D plots.
VX	The default variable used in symbolic operations.
ZPAR	Zoom parameters in plotting.

❹ Units

A unit object is comprised of a number and a unit separated by the underscore character. An example is 3_ ft/s. You can use the HP 49G to convert unit objects from one unit to another, comparable, unit. You can also use unit objects in calculations. The following table lists all the units you can use to create unit objects, grouped according to their category. You choose a category—and a unit—by first pressing ↱ UNITS. (Unit abbreviations are described in the *User's Guide*.)

Length						
	M	CM	MM	yd	ft	in
	Mpc	pc	lyr	au	km	Mi
	nmi	MiUS	chain	rd	fath	ftUS
	Mil	µ	Å	fermi		
Area						
	m^2	cm^2	b	yd^2	ft^2	in^2
	km^2	ha	a	mi^2	miUS^2	acre
Volume						
	m^3	st	cm^3	yd^3	ft^3	in^3
	l	galUK	galC	gal	qt	pt
	ml	cu	ozfl	ozUK	tbsp	tsp
	bbl	bu	pk	fbm		
Time						
	yr	d	h	min	s	Hz
Speed						
	m/s	cm/s	ft/s	kph	mph	knot
	c	ga				
Mass						
	kg	g	lb	oz	slug	lbt
	ton	tonUK	t	ozt	ct	grain
	u	mol				
Force						
	N	dyn	gf	kip	lbf	pdl
Energy						
	J	erg	Kcal	cal	Btu	ft×lbf
	therm	MeV	eV			
Power						
	W	hp				
Pressure						
	Pa	atm	bar	psi	torr	mmHg
	inHg	inH2O				
Temperature						
	ºC	ºF	K	ºR		
Electric Current						
	V	A	C	Ω	F	W
	Fdy	H	mho	S	T	Wb
Angle						
	º	r	grad	arcmin	arcs	sr
Light						
	fc	flam	lx	ph	sb	lm
	cd	lam				

Radium (i.e., radioactivity)						
	Gy	rad	rem	Sv	Bq	Ci
	R					
Viscosity						
	P	St				

5 Error and Status Messages

Errors during an operation or program execution generally cause the operation or program to abort and a message to appear.

The HP 49G enables you to detect and trap errors during program execution via the IFERR command. To identify the error after it has occurred, use the ERRN command to get its number, or ERRM to get its message.

You can also cause an error to occur in a program, via DOERR(*n*), where *n* is the error number of the desired error (see the table below). You can cause a customized error to occur, via DOERR("*message*"), where *message* is a character string of your choice.

The following table lists both error messages and status messages. These are sorted into categories

Number	Message
MEMORY MESSSAGES	
1	Insufficient Memory
5	Memory Clear
11	No Room in Port
13	Recovering Memory
14	Try To Recover Memory?
15	Replace RAM, press ON
16	No Mem To Config All
17	Undefined FPTR Name
18	Invalid bank data
19	Full check Bad Crc
20	Cmprs: not a user bank
21	No or 2 system bank
22	Invalid bank
23	Invalid bank number
24	Inexisting pack
25	Pack twice
26	Ins. memory
27	Erase Fail, Rom faulty
28	Erase Fail, Low bats
29	Erase Fail, Locked Block
30	Write Adr outside ROM
31	Write Fail, Rom faulty
32	Write Fail, Low bats
33	Write Fail, Locked Block
257	No Room to Save Stack
305	No Room to Show Stack
309	Out of Memory
337	Low Memory Condition…Please Wait

Number	Message
NAME AND DIRECTORY MESSAGES	
2	Directory Recursion
3	Undefined Local Name
4	Undefined XLIB Name
10	Port Not Available
12	Object Not in Port
259	Invalid User Function
297	Circular Reference
298	Directory Not Allowed
299	Non-Empty Directory
300	Invalid Definition
301	Missing Library
316	Name Conflict
3095	Invalid Name
MISCELLANEOUS SYSTEM MESSAGES	
6	Power Lost
8	Invalid Card Data
9	Object In Use
258	Can’t Edit Null Char.
294	HALT Not Allowed
296	Wrong Argument Count
3092	Low Battery
PLOT AND STATISTICS MESSAGES	
260	No Current Equation
302	Invalid PPAR
343	Y= not available
1537	Invalid ΣData
1538	Nonexistent ΣDAT
1539	Insufficient ΣData
1540	Invalid ΣPAR
1541	Invalid ΣData LN (Neg)
1542	Invalid ΣData LN (0)
1543	Invalid EQ
1545	No current equation.
1546	Enter eqn, press NEW
1547	Name the equation, press ENTER
1548	Select plot type
1549	Empty catalog
1551	No stat data to plot
1552	Autoscaling
1554	No current data. Enter
1555	Data point, press Σ+
1556	Select a model
1567	Off Screen
1568	Invalid PTYPE
1569	Name the stat data, press ENTER
1570	Enter value (zoom out if >1) press ENTER
1571	Copied to stack
1572	x axis zoom w/AUTO.
1573	x axis zoom
1574	y axis zoom
1575	x and y axis zoom.
1582	Enter matrix, then NEW
1583	No Associated Numeric View

Number	Message
	STACK AND COMMAND LINE MESSAGES
262	Invalid Syntax
292	Last Stack Disabled
293	Last Cmd Disabled
311	Last Stack
312	Last Commands
315	Last Arguments
317	Command Line
339	Nonexistent Find Pattern
340	Not Found
341	Nonexistent Replace Pattern
342	Can’t Find Selection
344	Warning … Changes will not be saved
513	Too Few Arguments
514	Bad Argument Type
515	Bad Argument Value
516	Undefined Name
517	LASTARG Disabled
3093	Empty Stack
	MATRIX AND ARRAY MESSAGES
1281	Invalid Dimension
1282	Invalid Array Element
1283	Deleting Row
1284	Deleting Column
1285	Inserting Row
1286	Inserting Column
	SOLVE MESSAGES
303	Non-Real Result
2561	Bad Guess(es)
2562	Constant?
2563	Interrupted
2564	Zero
2565	Sign Reversal
2566	Extremum
	TIME AND ALARM MESSAGES
314	Alarms
1557	No alarms pending
1558	Press ALRM to create
1559	Next alarm:
1560	Past due alarm:
1561	Acknowledged
1562	Enter alarm, press SET
1563	Select repeat interval
3329	Invalid Date
3330	Invalid Time
3331	Invalid Repeat
3332	Nonexistent Alarm
	EQUATION WRITER AND SYMBOLIC MESSAGES
304	Unable to Isolate
345	Result not editable in EQW
518	Incomplete Subexpression
519	Implicit () off
520	Implicit () on

Number	Message
	ARITHMETIC MESSAGES
769	Positive Underflow
770	Negative Underflow
771	Overflow
772	Undefined Result
773	Infinite Result
	I/O AND PRINTING MESSAGES
3073	Bad Packet Block Check
3074	Timeout
3075	Receive Error
3076	Receive Buffer Overrun
3077	Parity Error
3078	Transfer Failed
3079	Protocol Error
3080	Invalid Server Cmd.
3081	Port Closed
3082	Connecting
3083	Retry #
3084	Awaiting Server Cmd.
3085	Sending
3086	Receiving
3087	Object Discarded
3088	Packet #
3089	Processing Command
3090	Invalid IOPAR
3091	Invalid PRTPAR
	UNITS MESSAGES
2817	Invalid Unit
2818	Inconsistent Units

6 System Operations

For system operations, you press *and hold* the ON key, then press and release certain other keys before releasing ON.

Keys	Operation
ON F1 F6	Cold restart. Erases home and port 0 memory and resets the calculator's default settings.
ON F2	Cancels keystroke (prior to key release).
ON F3	Warm restart. Preserves memory.
ON F4	Starts interactive self-test.
ON F5	Starts continuous self-test.
ON ▲	Sends screen dump to the serial port.
ON 9	Cancels next repeating alarm.
ON −	Decreases screen contrast.
ON +	Increases screen contrast.
ON F6	Factory test.

7 System Flags

Flags are mode settings and mode indicators. To see a list of system flags, press MODE FLAGS.

Many flags can be set and cleared from input forms (such as the Calculator Modes input form, Display Modes input form, and others). You can also set, clear, or test a flag, by specifying the flag number as the argument in a flag command (SF, CF, FS?, etc).

Flag		Description of modes (* = default)
–1	Set:	Symbolic commands return principal solution.
	Clear:*	Symbolic commands return general solutions.
–2	Set:	Symbolic constants evaluate to numbers.
	Clear:*	Symbolic constants stay symbolic (if flag –3 is clear).
–3	Set:	Symbolic arguments evaluate to numbers.
	Clear:*	Symbolic arguments stay symbolic.
–5	Set:*	1st bit (value 1) of binary integer size is 1.
	Clear:	1st bit (value 1) of binary integer size is 0.
–6	Set:*	2nd bit (value 2) of binary integer size is 1.
	Clear:	2nd bit (value 2) of binary integer size is 0.
–7	Set:*	3rd bit (value 4) of binary integer size is 1.
	Clear:	3rd bit (value 4) of binary wordsize is 0.
–8	Set:*	4th bit (value 8) of binary wordsize is 1.
	Clear:	4th bit (value 8) of binary wordsize is 0.
–9	Set:*	5th bit (value 16) of binary wordsize is 1.
	Clear:	5th bit (value 16) of binary wordsize is 0.
–10	Set:*	6th bit (value 32) of binary wordsize is 1.
	Clear:	6th bit (value 32) of binary wordsize is 0.
–11	Set:*	HEX with –12 set, OCT with –12 clear.
	Clear:	DEC with –12 clear, BIN with –12 set.
–12	Set:*	HEX with –11 set, BIN with –11 clear.
	Clear:	OCT with –11 set, DEC with –11 clear.

Flag		Description of modes (* = default)
–14	Set:	TVM calculations use BEGIN payment mode.
	Clear:*	TVM calculations use END payment mode.
–15	Set:	Spherical mode (with flag –16 set).
	Clear:*	Cylindrical mode (with flag –16 set).
–16	Set:	Polar coordinate mode.
	Clear:*	Rectangular coordinate mode.
–17	Set:*	Radians mode if –18 clear.
	Clear:	Degrees if –18 clear, gradians if –18 set.
–18	Set:	Gradians if –17 clear.
	Clear:*	Radians if –17 set, degrees if –17 clear.
–19	Set:	→V2 creates a complex number.
	Clear:*	→V2 creates a 2-D vector.
–20	Set:	Underflow treated as an error.
	Clear:*	Underflow returns 0; sets flag –23 or –24.
–21	Set:	Overflow treated as an error.
	Clear:*	Overflow sets flag –25 and returns ± MAXR.
–22	Set:	Infinite result sets flag –26, returns ± MAXR.
	Clear:*	Infinite result treated as an error.
–23	Set:	Negative underflow condition exists (if flag –20 is clear).
	Clear:*	No negative underflow condition exists.
–24	Set:	Positive underflow condition exists (if flag –20 is clear).
	Clear:*	No positive underflow condition exists.
–25	Set:	Overflow condition exists (if flag –21 is clear).
	Clear:*	No overflow condition exists.
–26	Set:	Infinite result condition exists (if flag –22 is set).
	Clear:*	No infinite result condition exists.
–27	Set:*	Symbolic complex expression displays as 'x + yi'.
	Clear:	Symbolic complex expression displays as '(x,y)'.
–28	Set:	Multiple equations plot simultaneously.
	Clear:*	Multiple equations plot sequentially.
–29	Set:	No axes drawn for 2-D and statistical plots.
	Clear:*	Axes drawn for 2-D and statistical plots.
–31	Set:	No curve filling (connecting of points) in plots.
	Clear:*	Curve filling (connecting of points) in plots.
–32	Set:	Graphics cursor is inverse of background.
	Clear:*	Graphics cursor is always dark.
–35	Set:	I/O objects sent in binary.
	Clear:*	I/O objects sent in ASCII.
–36	Set:	In receiving I/O, a matching name overwrites.
	Clear:*	In receiving I/O, a matching name is changed.
–39	Set:	I/O messages suppressed.
	Clear:*	I/O messages displayed.
–40	Set:	Clock is displayed, providing that you have not hidden the status area (i.e., the header).
	Clear:*	Clock is not displayed.
–41	Set:	24-hour clock format.
	Clear:*	12-hour clock format.

Flag		Description of modes (* = default)
–42	Set:	DD.MM.YY date format.
	Clear:*	MM/DD/YY date format.
–43	Set:	Unacknowledged repeat alarms are not rescheduled.
	Clear:*	Unacknowledged repeat alarms are rescheduled.
–44	Set:	Acknowledged alarms are retained in the alarm list.
	Clear:*	Acknowledged alarms are deleted from alarm list.
–49	Set:	Fixed mode with –50 clear, engineering mode with –50 set.
	Clear:*	Standard mode with –50 clear, scientific mode with –50 set.
–50	Set:	Engineering mode with –49 set, scientific mode with –49 clear.
	Clear:*	Fixed mode with –49 set, standard mode with –49 clear.
–51	Set:	Fraction mark is a comma.
	Clear:*	Fraction mark is a period.
–52	Set:	Level 1 object is displayed on one line.
	Clear:*	Level 1 object is displayed on multiple lines.
–53	Set:	All parentheses are shown in algebraic expressions.
	Clear:*	Extra parentheses in algebraic expressions are removed.
–54	Set:	Small matrix values not set to 0; DET does not round.
	Clear:*	Small matrix values are set to 0; DET rounds.
–55	Set:	Most-recent arguments are not saved.
	Clear:*	Most-recent arguments are saved.
–56	Set:	Beep tone is enabled.
	Clear:*	Beep tone is disabled.
–57	Set:	Alarm tone is disabled.
	Clear:*	Alarm tone is enabled.
–58	Set:	Parameter and variable INFO not displayed.
	Clear:*	Parameter and variable INFO are displayed.
–60	Set:	Press alpha once for alpha mode lock.
	Clear:*	Press alpha twice for alpha mode lock.
–61	Set:	Press ↰ USER once for user mode lock.
	Clear:*	Press ↰ USER twice for user mode lock.
–62	Set:	User mode on.
	Clear:*	User mode off.
–63	Set:	User-defined ENTER is activated.
	Clear:*	ENTER evaluates the command line.
–64	Set:	The last GETI or PUTI wrapped index (to 1).
	Clear:*	The last GETI or PUTI does not wrap the index.
–65	Set:	Displays only the first level over multiple lines.
	Clear:*	Displays all levels over multiple lines.
–66	Set:	Displays long strings in single lines.
	Clear:*	Displays long strings in multiple lines.

Flag		Description of modes (* = default)
–67	Set:	When the clock shows (see flag –40), it is an analog display.
	Clear:*	When the clock shows (see flag –40), it is a digital display.
–68	Set:	Command line automatically indents.
	Clear:*	Command line does not automatically indent.
–69	Set:	Full-screen editing allowed.
	Clear:*	The cursor cannot move out of the text line.
–70	Set:	→GROB can accept multi-line strings.
	Clear:*	→GROB can accept only single-line strings.
–71	Set:	No addresses in ASM.
	Clear:*	Add addresses in ASM.
–72	Set:	The stack display uses mini-font.
	Clear:*	The stack display uses the current font.
–73	Set:	Command line editing uses mini-font.
	Clear:*	Command line editing uses the current font.
–74	Set:	The stack is left-justified.
	Clear:*	The stack is right-justified.
–76	Set:	File Manager purges need no confirmation.
	Clear:*	File Manager purges need confirmation.
–79	Set:	Algebraic objects display on the stack in standard form.
	Clear:*	Algebraic objects appear on the stack in textbook form.
–80	Set:	Textbook stack display uses minifont.
	Clear:*	Textbook stack display uses the current font.
–81	Set:	Editing a textbook grob uses minifont.
	Clear:*	Editing a textbook grob uses current font.
–82	Set:	Minifont used to edit algebraic in textbook mode.
	Clear:*	Current font used to edit algebraic in textbook mode.
–83	Set:	Grob description displayed on the stack.
	Clear:*	Grob contents displayed on the stack.
–85	Set:	SYSRPL stack display.
	Clear:*	Standard stack display.
–86	Set:	Program prefix off.
	Clear:*	Program prefix on.
–90	Set:*	Choose lists displayed in mini-font.
	Clear:	Choose lists displayed in the current font.
–91	Set:	Matrix Writer operates as a list of lists.
	Clear:*	Matrix Writer accepts arrays only.
–92	Set:	MASD SYSRPL.
	Clear:*	MASD assembler.
–94	Set:	In RPN mode, results are not stored in LASTCMD.
	Clear:*	In RPN mode, results are stored in LASTCMD.
–95	Set:	Algebraic mode.
	Clear:*	RPN mode.
–97	Set:	Lists are displayed vertically.
	Clear:*	Lists are displayed horizontally only.

Flag		Description of modes (* = default)
–98	Set:	Vectors are displayed vertically.
	Clear:*	Vectors are displayed horizontally only.
–99	Set:	CAS verbose mode.
	Clear:*	CAS concise mode.
–100	Set:	Final result mode.
	Clear:*	Step-by-step mode.
–103	Set:	Complex mode.
	Clear:*	Real mode.
–105	Set:	Approximate mode.
	Clear:*	Exact mode.
–106	Set:	TSIMP calls are not allowed in SERIES.
	Clear:*	TSIMP calls are allowed in SERIES.
–109	Set:	Numeric factorization is allowed.
	Clear:*	Numeric factorization is not allowed.
–110	Set:	Large matrices.
	Clear:*	Normal matrices.
–111	Set:	No recursive simplification in EXPAND and TSIMP.
	Clear:*	Recursive simplification in EXPAND and TSIMP.
–113	Set:	Do not apply linearity simplification when using integration CAS commands.
	Clear:*	Apply linearity simplification when using integration CAS commands.
–114	Set:	Polynomials expressed in increasing power order.
	Clear:*	Polynomials expressed in decreasing power order.
–116	Set:	Simplification to sine terms.
	Clear:*	Simplification to cosine terms
–117	Set:*	Menus displayed as choose lists.
	Clear:	Menus displayed as function keys.
–119	Set:	Non-rigorous mode.
	Clear:*	Rigorous mode.
–120	Set:	Calculator changes modes when necessary without prompting.
	Clear:*	Calculator prompts when it needs to change modes.

8 Object Types

The HP 49G makes use of 30 types of objects (listed in the table below). Commands relevant to object types are:

- TYPE(*obj*) Returns the object's type.
- VTYPE('*name*') Returns the named object's type.
- TVARS(*type*) Lists all objects of the specified type in the current directory.
- VARS Lists all objects in the current directory.

#	Type	Example
0	Real Number	–6.02E23
1	Complex Number	(.5,–1.57)
2	String	"Hi there!"
3	Real Array	[[1 2][3 4]]
4	Complex Array	[[(1,0) (5,–5)][(5,5) (0,1)]]
5	List	{ π 3.14 "PI" }
6	Global Name	X
7	Local Name	j
8	Program	« T 11 / »
9	Algebraic Object	4*π*r^2'
10	Binary Integer	# EFAC11h
11	Graphics Object	Graphic 131 × 64
12	Tagged Object	:Answer: 42
13	Unit Object	6_ft/min
14	XLIB Name	XLIB 543 8
15	Directory	DIR … END
16	Library	Library 440: …
17	Backup Object	Backup MYDIR
18	Built-in Function	SIN
19	Built-in Command	CLEAR
20	Internal Binary Integer	<123d>
21	Extended Real No.	Long Real
22	Extended Complex No.	Long Complex
23	Linked Array	Linked Array
24	Character Object	Character
25	Code Object	Code
26	Library Data	Library Data
27	Minifont	Font
28	Integer	5
29	Symbolic Vector/Matrix	[x x^2 x^3 x^4]
30	Font	Font

9 Character Keys

The following table lists all the characters available on the HP 49G. For each character, the table gives the character's internal number and the key or combination of keys that display the character. (An ampersand denotes that you hold down the first key while you press the second key). You can also display a character using the Characters tool (↱ chars).

Char.	No.	Key(s)	Char.	No.	Key(s)
…	31	↱ CHARS	U	85	ALPHA TAN
(sp)	32	SPC	V	86	ALPHA EEX
!	33	ALPHA ↱ 2	W	87	ALPHA +/-
"	34	ALPHA ↱ ×	X	88	x
#	35	↰ 3	Y	89	ALPHA 1/x
$	36	ALPHA ↰ 4	Z	90	ALPHA ÷
%	37	ALPHA ↰ 1	[	91	↰ × ▶ ◀
&	38	ALPHA ↰ ENTER	\	92	ALPHA ↱ 5
'	39	↱ EQW	]	93	↰ × ◀
(	40	↰ − ▶ ◀	^	94	y^x
)	41	↰ − ◀	_	95	↱ −
*	42	ALPHA ×	‘	96	↱ & EQW
+	43	ALPHA +	a	97	ALPHA ↰ F1
,	44	↱ SPC	b	98	ALPHA ↰ F2
-	45	ALPHA −	c	99	ALPHA ↰ F3
.	46	.	d	100	ALPHA ↰ F4
/	47	ALPHA ↱ ÷	e	101	ALPHA ↰ F5
0	48	0	f	102	ALPHA ↰ F6
1	49	1	g	103	ALPHA ↰ APPS
2	50	2	h	104	ALPHA ↰ MODE
3	51	3	l	105	ALPHA ↰ TOOL
4	52	4	j	106	ALPHA ↰ VAR
5	53	5	k	107	ALPHA ↰ STO▶
6	54	6	l	108	ALPHA ↰ NXT
7	55	7	m	109	ALPHA ↰ HIST
8	56	8	n	110	ALPHA ↰ CAT
9	57	9	o	111	ALPHA ↰ EQW
:	58	ALPHA ↰ .	p	112	ALPHA ↰ SYMB
;	59	ALPHA ↰ 2	q	113	ALPHA ↰ y^x
<	60	↱ x	r	114	ALPHA ↰ √x
=	61	↱ +/-	s	115	ALPHA ↰ SIN
>	62	↱ 1/x	t	116	ALPHA ↰ COS
?	63	ALPHA ↱ 3	u	117	ALPHA ↰ TAN
@	64	ALPHA ↱ ENTER	v	118	ALPHA ↰ EEX
A	65	ALPHA F1	w	119	ALPHA ↰ +/-
B	66	ALPHA F2	x	120	ALPHA ↰ x
C	67	ALPHA F3	y	121	ALPHA ↰ 1/x
D	68	ALPHA F4	z	122	ALPHA ↰ ÷
E	69	ALPHA F5	{	123	↰ + ▶ ◀
F	70	ALPHA F6	|	124	↱ TOOL
G	71	ALPHA APPS	}	125	↰ + ◀
H	72	ALPHA MODE	~	126	ALPHA ↱ 1
I	73	ALPHA TOOL	▒	127	↱ CHARS
J	74	ALPHA VAR	∡	128	ALPHA ↱ 6
K	75	ALPHA STO▶	x̄	129	↱ CHARS
L	76	ALPHA NXT	v̄	130	↱ CHARS
M	77	ALPHA HIST	√	131	√x
N	78	ALPHA CAT	∫	132	↱ TAN
O	79	ALPHA EQW	Σ	133	↱ SIN
P	80	ALPHA SYMB	▶	134	STO▶
Q	81	ALPHA y^x	π	135	↰ SPC
R	82	ALPHA √x	∂	136	↱ COS
S	83	ALPHA SIN	≤	137	↰ x
T	84	ALPHA COS	>	138	↰ 1/x

Char.	No.	Key(s)	Char.	No.	Key(s)
≠	139	↰ +/-	Æ	198	ALPHA F5 ALPHA ↱ 9
α	140	ALPHA ↱ F1	Ç	199	ALPHA F3 ALPHA ↱ 9
→	141	↱ 0	È	200	ALPHA F5 ALPHA ↰ 7
←	142	↱ CHARS	É	201	ALPHA F5 ALPHA ↱ 7
↓	143	↱ CHARS	Ê	202	ALPHA F5 ALPHA ↰ 8
↑	144	↱ CHARS	Ë	203	ALPHA F5 ALPHA ↰ 9
γ	145	↱ CHARS	Ì	204	ALPHA TOOL ALPHA ↰ 7
δ	146	ALPHA ↱ F4	Í	205	ALPHA TOOL ALPHA ↱ 7
∊	147	ALPHA ↱ F5	Î	206	ALPHA TOOL ALPHA ↰ 8
η	148	↱ CHARS	Ï	207	ALPHA TOOL ALPHA ↰ 9
θ	149	ALPHA ↱ COS	Ð	208	ALPHA F4 ALPHA ↱ 9
λ	150	ALPHA ↱ CAT	Ñ	209	ALPHA CAT ALPHA ↱ 8
ρ	151	↱ CHARS	Ò	210	ALPHA EQW ALPHA ↰ 7
σ	152	ALPHA ↱ SIN	Ó	211	ALPHA EQW ALPHA ↱ 7
τ	153	ALPHA ↱ TAN	Ô	212	ALPHA EQW ALPHA ↰ 8
ω	154	ALPHA ↱ EEX	Õ	213	ALPHA EQW ALPHA ↱ 8
∆	155	ALPHA ↱ F3	Ö	214	ALPHA EQW ALPHA ↰ 9
Π	156	ALPHA ↱ SYMB	×	215	↱ CHARS
Ω	157	ALPHA ↱ EQW	Ø	216	ALPHA EQW ALPHA ↱ 9
▪	158	↱ CHARS	Ù	217	ALPHA TAN ALPHA ↰ 7
∞	159	↰ 0	Ú	218	ALPHA TAN ALPHA ↱ 7
€	160	ALPHA ↱ 4	Û	219	ALPHA TAN ALPHA ↰ 8
¡	161	ALPHA ↱ &2	Ü	220	ALPHA TAN ALPHA ↰ 9
¢	162	↱ CHARS	Ý	221	ALPHA 1/x ALPHA ↱ 7
£	163	ALPHA ↰ 5	Þ	222	ALPHA SYMB ALPHA ↱ 9
¤	164	↱ CHARS	ß	223	ALPHA ↱ F2
¥	165	↱ CHARS	à	224	ALPHA ↰ F1 ALPHA ↰ 7
¦	166	↱ CHARS	á	225	ALPHA ↰ F1 ALPHA ↱ 7
§	167	ALPHA ↰ 6	â	226	ALPHA ↰ F1 ALPHA ↰ 8
¨	168	↱ CHARS	ã	227	ALPHA ↰ F1 ALPHA ↱ 8
❸	169	↱ CHARS	ä	228	ALPHA ↰ F1 ALPHA ↰ 9
ª	170	↱ CHARS	å	229	ALPHA ↰ F1 ALPHA ↱ 9
«	171	↰ + ▶ ◀	æ	230	ALPHA ↰ F5 ALPHA ↱ 9
¬	172	↱ CHARS	ç	231	ALPHA ↰ F3 ALPHA ↱ 9
-	173	↱ CHARS	è	232	ALPHA ↰ F5 ALPHA ↰ 7
❷	174	↱ CHARS	é	233	ALPHA ↰ F5 ALPHA ↱ 7
¯	175	↱ CHARS	ê	234	ALPHA ↰ F5 ALPHA ↰ 8
°	176	ALPHA ↱ &6	ë	235	ALPHA ↰ F5 ALPHA ↰ 9
±	177	↱ CHARS	ì	236	ALPHA ↰ TOOL ALPHA ↰ 7
²	178	↱ CHARS	í	237	ALPHA ↰ TOOL ALPHA ↱ 7
³	179	↱ CHARS	î	238	ALPHA ↰ TOOL ALPHA ↰ 8
´	180	↱ CHARS	ï	239	ALPHA ↰ TOOL ALPHA ↰ 9
µ	181	ALPHA ↱ HIST	ð	240	ALPHA ↰ F4 ALPHA ↱ 9
¶	182	↱ CHARS	ñ	241	ALPHA ↰ CAT ALPHA ↱ 8
•	183	↱ CHARS	ò	242	ALPHA ↰ EQW ALPHA ↰ 7
¸	184	↱ CHARS	ó	243	ALPHA ↰ EQW ALPHA ↱ 7
¹	185	↱ CHARS	ô	244	ALPHA ↰ EQW ALPHA ↰ 8
º	186	↱ CHARS	õ	245	ALPHA ↰ EQW ALPHA ↱ 8
»	187	↰ + ◀	ö	246	ALPHA ↰ EQW ALPHA ↰ 9
¼	188	↱ CHARS	÷	247	↱ CHARS
½	189	↱ CHARS	ø	248	ALPHA ↰ EQW ALPHA ↱ 9
¾	190	↱ CHARS	ù	249	ALPHA ↰ TAN ALPHA ↰ 7
¿	191	ALPHA ↱ &3	ú	250	ALPHA ↰ TAN ALPHA ↱ 7
À	192	ALPHA F1 ALPHA ↰ 7	û	251	ALPHA ↰ TAN ALPHA ↰ 8
Á	193	ALPHA F1 ALPHA ↱ 7	ü	252	ALPHA ↰ TAN ALPHA ↰ 9
Â	194	ALPHA F1 ALPHA ↰ 8	ý	253	ALPHA ↰ 1/x ALPHA ↱ 7
Ã	195	ALPHA F1 ALPHA ↱ 8	þ	254	ALPHA ↰ SYMB ALPHA ↱ 9
Ä	196	ALPHA F1 ALPHA ↰ 9	ÿ	255	ALPHA ↰ 1/x ALPHA ↰ 9
Å	197	ALPHA F1 ALPHA ↱ 9			

Command Reference

All the HP 49G commands are listed in the table commencing on page 21. A brief description of each command is provided, together with the key or keys that provide access to the command. Where appropriate, at least one argument (input) and the corresponding result (output) is provided. In many cases, a command can take many more types of argument. To se a full listing of the arguments applicable to each command, see the *Advanced User's Guide*.

The commands are listed alphabetically. Commands referred to solely by a non-alphabetic character—for example,%—are listed after those referred to by alphabetic characters. Where a non-alphabetic character is the first character—for example, →DIAG—the command is sorted as if the character did not exist. In other cases where a command name includes a non-alphabetic character—for example, I→R and DIAG→—the non-alphabetic character is treated as 'Z' in sorting the commands.

The commands that are functions are indicated by an asterisk at the end of the command description. (You can include functions in an algebraic expression.)

The codes and abbreviations used to represent the inputs and outputs are set out in the following table.

Code	Meaning
x, *y*, *a*, *b*, etc	Real number
z	Real or complex number
x_units	Unit object
(*x*, *y*)	Complex number
n or *m*	Integer
#n or #m	Binary integer
[vector]	Real or complex vector
[[matrix]]	Real, symbolic, or complex matrix
[[array]]	Real or complex array
"*string*"	String of characters
'*symb*'	Expression
'*name*'	Variable name
T/F	True (non-zero value) or false (0)
grob	Graphics object
obj	Any object
{ *obj x z* }	List of objects

In algebraic mode, the order that the inputs are listed is the same as the order in which you must specify the arguments. Similarly, the outputs are listed in the order in which they are returned.

In RPN mode, the last input is what should be on level 1 prior to executing the command, the second last input is what should be on level 2, the third last on level 3, and so on. Similarly, the last output appears on level 1, the second last appears on level 2, and so on.

Name	Description	Access	Inputs	Outputs
ABCUV	Returns a solution in polynomials *u* and *v* of *au*+*bv* =*c* where *a* and *b* are polynomials, and *c* is a value.	↰ ARITH POLYNOMIAL	‘$symb_1$’ ‘$symb_2$’ *z* →	‘$symb_3$’ ‘$symb_4$’
ABS	Returns the absolute value of its argument.*	↰ ABS	*x* →	*|x|*
ACK	Acknowledges the oldest past-due alarm.	↱ TIME TOOLS ALRM		
ACKALL	Acknowledges all past-due alarms.	↱ TIME TOOLS ALRM		
ACOS	Returns the value of the angle with the given cosine.	↰ ACOS	*z* →	*acos z*
ACOS2S	Replaces cos() terms with equivalent asin() terms.*	↱ TRIG	‘$symb_1$’ →	‘$symb_2$’
ACOSH	Returns the inverse hyperbolic cosine of the argument.*	↱ TRIG HYPERBOLIC	*z* →	*acosh z*
ADD	Adds corresponding elements of two lists or adds a number to each of the elements of a list.	CAT	{ $list_1$ } { $list_2$ } →	{ $list_{result}$ }
ADDTMOD	Adds two expressions or values, modulo the current modulus.*	↰ ARITH MODULO	‘$symb_1$’ ‘$symb_2$’ →	‘$symb_3$’
ADDTOREAL	Adds the specified global name to the reserved variable REALASSUME.	CAT	‘*global*’ →	

* = function

Name	Description	Access	Inputs	Outputs
ALOG	Returns the common antilogarithm; that is, 10 raised to the given power.*	↰ 10^x	z →	10^z
AMORT	Amortizes a loan or investment based upon the current amortization settings.	↰ FINANCE	n →	*principal interest balance*
AND	Returns the logical AND of two arguments.*	↱ BASE LOGIC	$\#n_1$ $\#n_2$ →	$\#n_3$
ANIMATE	Displays graphic objects in sequence.	↰ PRG GROB	$grob_n...grob_1$ n_{grobs} →	*same stack*
ANS	Recalls the *n*th answer from history.	↰ ANS	n →	obj_n
APPLY	Creates an expression from the specified function name and arguments.*	CAT	{ $symb_1$ ... $symb_n$ } '*name*' →	'*name* ($symb_1$ … $symb_n$)'
ARC	Draws an arc in PICT counterclockwise.	↰ PRG PICT	(x, y) x_{radius} x_{q1} x_{q2} →	
ARCHIVE	Creates a backup copy of the *HOME* directory.	↰ PRG MEMORY	:n_{port}: *name* →	
ARG	Returns the (real) polar angle of a complex number.*	↱ ARG	(x, y) →	θ
ARIT	Displays a menu of arithmetic commands.	CAT		
→ARRY	Returns a vector of *n* real or complex elements or a matrix of *n* × *m* real or complex elements.	CAT	z_1 … z_n $n_{element}$ →	[*vector*]
ARRY→	Takes an array and returns its elements as separate real or complex numbers.	CAT	[*vector*] →	z_1 … z_n { $n_{element}$ }
ASIN	Returns the value of the angle with the given sine.*	↰ ASIN	z →	*asin z*

* = function

Name	Description	Access	Inputs	Outputs
ASIN2C	Replaces asin() terms with equivalent acos() terms.	↱ TRIG	'$symb_1$' →	'$symb_2$'
ASIN2T	Replaces asin() terms with equivalent atan() terms.	↱ TRIG	'$symb_1$' →	'$symb_2$'
ASINH	Returns the inverse hyperbolic sine of the argument.*	↱ TRIG HYPERBOLIC	z →	*asinh z*
ASN	Defines a key on the user keyboard by assigning the given object to the key x_{key}, specified as *row column.position.*	CAT	*obj* x_{key} →	
ASR	Shifts a binary integer one bit to the right, except for the most significant bit, which is maintained.	↱ BASE BIT	$\#n_1$ →	$\#n_2$
ATAN	Returns the value of the angle having the given tangent.*	↰ ATAN	z →	*atan z*
ATAN2S	Replaces atan(*x*) terms with asin(*x*) terms.	↱ TRIG	'$symb_1$' →	'$symb_2$'
ATANH	Returns the inverse hyperbolic tangent of the argument.*	↱ TRIG HYPERBOLIC	z →	*atanh z*
ATICK	Sets the axes tick-mark annotation in the reserved variable *PPAR.*	CAT	x →	
ATTACH	Attaches the library with the specified number to the current directory.	CAT	n_{library} →	
AUTO	Calculates a *y*-axis display range, or an *x*- and *y*-axis display range.	CAT		
AXES	Specifies the intersection coordinates of, and labels for, the *x*- and *y*-axes, and the tick-mark annotation.	CAT	(x, y) →	

* = function

Name	Description	Access	Inputs	Outputs
AXL	Converts a list to an array, or an array to a list.	↰ CONVERT	{*list*}/[[*array*]] →	[[*array*]]/{*list*}
AXM	Converts a numeric array into a symbolic matrix.	↰ MATRICES OPERATIONS	[[*array*]] →	[[*matrix*]]
AXQ	Converts a square matrix into the associated quadratic form.	↰ CONVERT	[[*matrix*]] →	'*symb*' [*vector*]
BAR	Sets the plot type to BAR.	CAT		
BARPLOT	Plots a bar chart of the specified column of the current statistics matrix (reserved variable Σ*DAT*).	CAT		
BASE	Displays a menu of basic algebra commands.	CAT		
BAUD	Specifies bit-transfer rate.	CAT	n_{baudrate} →	
BEEP	Sounds a tone at *n* hertz for *x* seconds.	↰ PRG OUT	$n_{\text{frequency}}$ x_{duration} →	
BESTFIT	Executes LR with the four curve fitting models, and selects the model yielding the largest correlation coefficient.	CAT		
BIN	Selects binary base for binary integer operations.	CAT		
BINS	Sorts the elements of the independent column of the current statistics matrix into (n_{bins} + 2) bins.	CAT	x_{min} x_{width} n_{bins} →	$[[\ n_{\text{bin 1}} \ldots n_{\text{bin n}}\ ]]$ $[\ n_{\text{bin L}}\ n_{\text{bin R}}\ ]$
BLANK	Creates a blank grob of the specified width and height.	↰ PRG GROB	$\#n_{\text{width}}$ $\#m_{\text{height}}$ →	$grob_{\text{blank}}$
BOX	Draws in *PICT* a box whose opposite corners are defined by the specified pixel or user-unit coordinates.	↰ PRG PICT	$\{ \#n_1\ \#m_1 \}$ $\{ \#n_2\ \#m_2 \}$ →	

* = function

Name	Description	Access	Inputs	Outputs
BUFLEN	Returns the number of characters in the serial input buffer and a single digit indicating whether an error occurred.	CAT		→ n_{chars} 0/1
BYTES	Returns the number of bytes and the checksum for the given object.	↰ PRG MEMORY	*obj*	→ $\#n_{\text{checksum}}$ x_{size}
B→R	Converts a binary integer to its floating-point equivalent.	↱ BASE	*#n*	→ *n*
CASCFG	Restores the default CAS mode settings.	CAT		
CASE	Starts CASE … END conditional structure.	↰ PRG BRCH		
CEIL	Returns the smallest integer greater than or equal to the argument.*	↰ MTH REAL	*x*	→ *n*
CENTR	Adjusts first two parameters in *PPAR,* (x_{min}, y_{min}) and (x_{max}, y_{max}), so that point (*x, y*) is plot center.	CAT	*(x, y)*	→
CF	Clears the specified user or system flag.	↰ PRG TEST	$n_{\text{flagnumber}}$	→
%CH	Returns the percent change from *x* to *y* as a percentage of *x*.*	↰ MTH REAL	*x y*	→ 100(*y* – *x*)/*x*
CHINREM	Solves a system of simultaneous polynomial congruences in the ring Z[*x*].	↰ ARITH POLYNOMIAL	[$vector_1$] [$vector_2$]	→ [$vector_3$]
CHOOSE	Creates a user-defined choose box.	↰ PRG IN	"*prompt*" { c_1 ... c_n } n_{pos}	→ *obj or result* "*1*"

* = function

Name	Description	Access	Inputs	Outputs
CHR	Returns a string representing the character corresponding to the character code *n.*	↰ PRG TYPE	n →	"*string*"
CKSM	Specifies the error-detection scheme.	CAT	n_{checksum} →	
CLEAR	Removes all objects from the stack or history.	↱ CLEAR	$obj_n \ldots obj_1$ →	
CLKADJ	Adjusts the system time by *x* clock ticks, where 8192 clock ticks equal 1 second.	↱ TIME TOOLS	x →	
CLLCD	Clears (blanks) the stack display.	↰ PRG OUT		
CLOSEIO	Closes the serial port, and clears the input buffer and any error messages for KERRM.	CAT		
CLΣ	Purges the current statistics matrix.	CAT		
CLVAR	Purges all variables and empty subdirectories in the current directory.	CAT		
CMPLX	Displays a menu of commands pertaining to complex numbers.	CAT		
CNRM	Returns the column norm (one-norm) of the array argument.	↰ MATRICES OPERATIONS	[*array*] →	$x_{\text{columnnorm}}$
COL–	Deletes column *n* of a matrix, and returns the modified matrix (or vector) and the deleted column (or element).	↰ MTH MATRIX COL	$[[\ matrix\ ]]_1\ \ n_{\text{column}}$ →	$[[\ matrix\ ]]_2\ \ [\ vector\ ]_{\text{column}}$
COL+	Inserts an array into a matrix at the position indicated by n_{index}, and returns the modified array.	↰ MTH MATRIX COL	$[\ vector\ ]_1\ \ n_{\text{element}}\ \ n_{\text{index}}$ →	$[\ vector\ ]_2$

* = function

Name	Description	Access	Inputs	Outputs
→COL	Transforms a matrix into a series of column vectors and returns the vectors and a column count.	↰ MTH MATRIX COL	[[*matrix*]]	→ [*vector*]$_{\text{col1}}$ [*vector*]$_{\text{coln}}$ n_{colcount}
COL→	Transforms a series of column vectors and a column count into a matrix containing those columns.	↰ MTH MATRIX COL	[*vector*]$_{\text{col1}}$ [*vector*]$_{\text{coln}}$ n_{colcount}	→ [[*matrix*]]
COLCT	Factorizes a polynomial or integer. Identical to FACTOR.	CAT	'$symb_1$'	→ '$symb_2$'
COLΣ	Specifies the independent-variable and dependent-variable columns of the current statistics matrix.	CAT	x_{col} y_{col}	→
COMB	Returns the number of possible combinations of *n* items taken *m* at a time.*	↰ MTH PROBABILITY	*n* *m*	→ $C_{\text{n,m}}$
CON	Returns a constant array, defined as an array whose elements all have the same value.	↰ MTH MATRIX MAKE	{ n_{columns} } z_{constant}	→ [$vector_{\text{constant}}$]
COND	Returns the 1-norm (column norm) condition number of a square matrix.	↰ MTH MATRIX NORMALIZE	[[*matrix*]]$_{\text{m×n}}$	→ $x_{\text{condition number}}$
CONIC	Sets the plot type to CONIC.	CAT		
CONJ	Conjugates a complex number or a complex array.*	↱ CMPLX	*x*	→ *x*
CONLIB	Opens the Constants Library catalog.	APPS CONSTANTS LIB		
CONST	Returns the value of a constant.*	CAT	'*name*'	→ *x*
CONT	Resumes execution of a halted program.	↰ CONT		

* = function

Name	Description	Access	Inputs	Outputs
CONVERT	Converts a source unit object to the dimensions of a target unit.	↰ CONVERT UNITS TOOLS	$x1\_units_{\text{source}}$ $x2\_units_{\text{target}}$	→ $x_3\_units_{\text{target}}$
CORR	Returns the correlation coefficient of the independent and dependent data columns in the current statistics matrix.	TOOL		→ $x_{\text{correlation}}$
COS	Returns the cosine of the argument.*	COS	z	→ cos z
COSH	Returns the hyperbolic cosine of the argument.*	↱ TRIG HYPERBOLIC	z	→ cosh z
COV	Returns the sample covariance of the independent and dependent data columns in the current statistics matrix.	CAT		→ $x_{\text{covariance}}$
CR	Prints the contents, if any, of the printer buffer.	CAT		
CRDIR	Creates an empty subdirectory with the specified name in the current directory.	↰ PRG MEMORY DIRECTORY	‘*global*’	→
CROSS	Returns the cross product C = A × B of vectors A and B.	↰ MTH VECTOR	$[\ vector\ ]_A$ $[\ vector\ ]_B$	→ $[\ vector\ ]_{A \times B}$
CSWP	Swaps columns *i* and *j* of the argument matrix and returns the modified matrix.	↰ MATRICES CREATE COLUMN	$[[\ matrix\ ]]_1$ $n_{\text{column i}}$ $n_{\text{column j}}$	→ $[[\ matrix\ ]]_2$
CURL	Returns the curl of a three-dimensional vector function.	↰ CALC DERIV AND INTEG	$[\ vector_1\ ]$ $[[\ array_1\ ]]$	→ ‘$symb_1$’
CYLIN	Sets Cylindrical coordinate mode.	CAT		
C→PX	Converts the specified user-unit coordinates to pixel coordinates.	↰ PRG PICT	(*x*, *y*)	→ { #*n*, #*m* }

* = function

Name	Description	Access	Inputs	Outputs
C→R	Separates the real and imaginary parts of a complex number or complex array.	↰ PRG TYPE	(x, y)	→ x y
DARCY	Calculates the Darcy friction factor of certain fluid flows.*	CAT	xe / D yRe	→ x_{Darcy}
→DATE	Sets the system date to *date.*	↱ TIME TOOLS	*date*	→
DATE	Returns the system date.	↱ TIME TOOLS		→ *date*
DATE+	Returns a past or future date, given a date in argument 1/level 2 and a number of days in argument 2/level 1.	↱ TIME TOOLS	$date_1$ x_{days}	→ $date_{\text{new}}$
DBUG	Starts program execution, then suspends it as if HALT were the first program command.	CAT	« *program* » or '*program name*'	→
DDAYS	Returns the number of days between two dates.	↱ TIME TOOLS	$date_1$ $date_2$	→ x_{days}
DEC	Selects decimal base for binary integer operations. (The default base is decimal).	CAT		
DECR	Takes a variable, subtracts 1, stores the new value back into the original variable, and returns the new value.	↰ PRG MEMORY ARITHMETIC	'*name*'	→ x_{new}
DEFINE	Stores the expression on the right side of the = in the variable specified on the left side, or creates a user-defined function.	↰ DEF	'*name=exp*'	→
DEG	Sets the angle mode to degrees.	CAT		
DELALARM	Deletes the specified alarm.	↱ TIME TOOLS ALRM	n_{index}	→

* = function

Name	Description	Access	Inputs	Outputs
DELAY	Specifies how many seconds the HP 49 waits between sending lines of information to the printer.	CAT	x_{delay}	→
DELKEYS	Clears user-defined key assignments.	CAT	x_{key}	→
DEPND	Specifies the dependent variable (and its plotting range for TRUTH plots).	CAT	‘*global*’	→
DEPTH	Returns a real number representing the number of objects present on the stack (before DEPTH was executed).	⬑ PRG STACK		→ n
DERIV	Returns the partial derivatives of a function, with respect to the specified variables.*	⬑ CALC DERIV AND INTEG	‘$symb_1$’ z	→ ‘$symb_2$’
DERVX	Returns the derivative of a function with respect to the current variable.*	⬑ CALC DERIV AND INTEG	‘$symb_1$’	→ ‘$symb_2$’
DESOLVE	Solves certain first-order ordinary differential equations with respect to the current variable.	⬑ S.SLV	‘$symb_1$’ ‘$symb_2$’	→ ‘$symb_3$’
DET	Returns the determinant of a square matrix.	⬑ MATRICES OPERATIONS	[[*matrix*]]	→ $x_{\text{determinant}}$
DETACH	Detaches the library with the specified number from the current directory.	CAT	n_{library}	→
→DIAG	Returns a vector that contains the major diagonal elements of a matrix.	⬑ MATRICES CREATE	[[*matrix*]]	→ [*vector*]$_{\text{diagonals}}$

* = function

Name	Description	Access	Inputs	Outputs
DIAG→	Takes an array and a dimension and returns a matrix whose major diagonal is the elements of the array.	↰ MATRICES CREATE	$[\ \mathit{array}\ ]_{\text{diagonals}}$ { *dim* } →	[[*matrix*]]
DIFF	Displays a menu of calculus commands.	CAT		
DIFFEQ	Sets the plot type to DIFFEQ.	CAT		
DISP	Displays *obj* in the *n*th display line.	↰ PRG OUT	*obj* *n* →	
DIV	Returns the divergence of a vector function.	↰ CALC DERIV AND INTEG	[[array_1]] [[array_2]] →	‘symb_1’
DIV2	Performs euclidean division on two expressions. Step-by-step mode is available with this command.	↰ ARITH POLYNOMIAL	‘symb_1’ ‘symb_2’ →	‘symb_3’
DIV2MOD	Performs euclidean division on two expressions modulo the current modulus.	↰ ARITH MODULO	‘symb_1’ ‘symb_2’ →	‘symb_3’
DIVIS	Returns a list of the divisors of a polynomial or an integer.	↰ ARITH	‘symb_1’ →	{ list_1 }
DIVMOD	Divides two expressions modulo the current modulus.*	↰ ARITH MODULO	‘symb_1’ *z* →	‘symb_2’
DIVPC	Returns a Taylor polynomial for the quotient of two expressions.	↰ CALC LIMITS & SERIES	‘symb_1’ ‘symb_2’ *z* →	‘symb_3’
DO	Starts DO … UNTIL … END indefinite loop structure.	↰ PRG BRANCH		
DOERR	Executes a “user-specified” error, causing a program to behave exactly as if a normal error occurred.	↰ PRG ERROR	n_{error} →	

* = function

Name	Description	Access	Inputs	Outputs
DOLIST	Applies commands, programs, or user-defined functions to lists.	↰ PRG LIST PROCEDURES	$\{$ *list* $\}_1$ ... $\{$ *list* $\}_n$ *n* « *program* »	→ { *results* }
DOSUBS	Applies a program or command to groups of elements in a list.	↰ PRG LIST PROCEDURES	$\{$ *list* $\}_1$ *n* « *program* »	→ $\{$ *list* $\}_2$
DOT	Returns the dot product A·B of two arrays A and B.	↰ MATRICES VECTOR	[$array_A$] [$array_B$]	→ *x*
DRAW	Plots the mathematical data in the reserved variable *EQ*.	CAT		
DRAW3DMATRIX	Draws a 3D plot from the values in a specified matrix.	CAT	[[*matrix*]] v_{min} v_{max}	→
DRAX	Draws axes in *PICT.*	CAT		
DROP	Removes the level 1 object from the stack.	↰ PRG STACK	*obj*	→
DROP2	Removes the first two objects from the stack.	↰ PRG STACK	obj_1 obj_2	→
DROPN	Removes the first *n* + 1 objects from the stack (the first *n* objects excluding the integer *n* itself).	↰ PRG STACK	obj_1 ... obj_n *n*	→
DTAG	Removes all tags (labels) from an object.	↰ PRG TYPE	*tag:obj*	→ *obj*
DUP	Returns a copy of the argument (or the object on level 1).	↰ PRG STACK	*obj*	→ *obj obj*
DUP2	Returns copies of the two arguments (or the objects on levels 1 and 2 of the stack).	↰ PRG STACK	obj_2 obj_1	→ obj_2 obj_1 obj_2 obj_1
DUPDUP	Duplicates an object twice.	CAT	*obj*	→ *obj obj obj*

* = function

Name	Description	Access	Inputs	Outputs
DUPN	Takes integer *n* from level 1 of the stack, and returns copies of objects on stack levels 2 through to *n* + 1.	↰ PRG STACK	obj_1 … obj_n n	→ obj_1 … obj_n obj_1 … obj_n
D→R	Converts a real number representing an angle in degrees to its equivalent in radians.*	↰ MTH REAL	x	→ $(\pi/180)x$
e	Returns the symbolic constant *e* or its numerical representation, 2.71828182846.*	ALPHA ↰ E		→ '*e*'
EDIT	Moves specified object to the command line where it can be edited.	CAT	*obj*	→
EDITB	Opens an object in the most suitable editing environment.	CAT	*obj*	→
EGCD	Given two polynomials, *u* and *v*, returns *a*, *b*, and *c* where: *au* + *bv* = *c.*	↰ ARITH POLYNOMIAL	'$symb_1$' '$symb_2$'	→ '$symb_3$' '$symb_4$' '$symb_5$'
EGV	Computes the eigenvalues and right eigenvectors for a square matrix.	↰ MATRICES EIGENVECTOR	$[[matrix]]_A$	→ $[[matrix]]_{EVec}$ $[vector]_{EVal}$
EGVL	Computes the eigenvalues of a square matrix.	↰ MATRICES EIGENVECTOR	$[[matrix]]_A$	→ $[vector]_{EVal}$
ELSE	Starts false clause in conditional or error-trapping structure.	↰ PRG BRANCH		
END	Ends conditional, error-trapping, and indefinite loop structures.	↰ PRG BRANCH		
ENDSUB	Provides a way to access the total number of sub-lists contained in the list used by DOSUBS.	↰ PRG LIST PROCEDURES		

* = function

Name	Description	Access	Inputs	Outputs
ENG	Sets the number display format to engineering mode.	CAT	*n*	→
EPSX0	Replaces with zero those coefficients in a polynomial that have an absolute value less than the variable EPS.	CAT	'$symb_1$'	→ '$symb_2$'
EQW	Opens Equation Writer, where you can edit an expression.	CAT	exp_1	→ exp_2
EQ→	Separates an equation into its left and right sides.	↰ PRG TYPE	'$symb_1$=$symb_2$'	→ '$symb_1$' '$symb_2$'
ERASE	Erases *PICT*, leaving a blank *PICT* of the same dimensions.	CAT		
ERR0	Clears the last error number (and message) so that a subsequent execution of ERRN returns # 0h.	↰ PRG ERROR		
ERRM	Returns a string containing the error message of the most recent calculator error.	↰ PRG ERROR		→ "*error message*"
ERRN	Returns the error number of the most recent calculator error.	↰ PRG ERROR		→ $\#n_{\text{error}}$
EULER	Returns the number of integers less than a specified integer that are co-prime with the integer.*	↰ ARITH INTEGER	z_1	→ z_2
EVAL	Evaluates the object.	↱ EVAL	*obj*	→
EXLR	Returns the left- and right-hand sides of an equation as discrete expressions.	CAT	'$symb_1$'	→ '$symb_2$' '$symb_3$'
EXP	Returns the exponential, or natural antilogarithm, of the argument; that is, *e* raised to the given power.*	↰ e^x	*z*	→ e^z

* = function

Name	Description	Access	Inputs	Outputs
EXPAN	Expands and simplifies an algebraic expression.	CAT	‘$symb_1$’ →	‘$symb_2$’
EXPAND	Expands and simplifies an algebraic expression.	↱ ALG	‘$symb_1$’ →	‘$symb_2$’
EXPANDMOD	Expands and simplifies an expression, modulo the current modulus.*	↰ ARITH MODULO	‘$symb_1$’ →	‘$symb_2$’
EXPFIT	Stores EXPFIT in Σ*PAR*, thus subsequent executions of LR will use the exponential curve fitting model.	CAT		
EXPLN	Transforms the trigonometric terms in an expression to exponential and logarithmic terms.	↰ CONVERT	‘$symb_1$’ →	‘$symb_2$’
EXPM	Returns $e^x - 1$.*	↰ MTH HYPERBOLIC	x →	$e^x - 1$
EYEPT	Specifies the coordinates of the eye point in a perspective plot.	CAT	x_{point} y_{point} z_{point} →	
F0λ	Returns the fraction of total black-body emissive power at temperature x_T between wavelengths 0 and y_{lambda}.*	CAT F0λ	y_{lambda} xT →	x_{power}
FACT	FACT is the same as !. See !.	↰ MTH PROBABILITY	n →	n!
FACTOR	Factorizes a polynomial or an integer.	↱ ALG	‘$symb_1$’ →	‘$symb_2$’
FACTORMOD	Factorizes a polynomial modulo the current modulus. The modulus must be less than 100, and a prime number.*	↰ ARITH MODULO	‘$symb_1$’ →	‘$symb_2$’

* = function

Name	Description	Access	Inputs	Outputs
FACTORS	For a value or expression, returns a list of prime factors and their multiplicities.	↰ ARITH	z →	{ *list* }
FANNING	Calculates the Fanning friction factor of certain fluid flows.*	CAT	$x_{x/D}$ y_{Re} →	x_{fanning}
FAST3D	Sets the plot type to FAST 3D.	CAT		
FCOEF	From an array of roots and multiplicities/poles, returns a rational polynomial with a leading coefficient of 1.	↰ ARITH POLYNOMIAL	[[$array_1$]] →	'$symb_1$'
FC?	Tests whether the specified system or user flag is clear, and returns a corresponding test result.	↰ PRG TEST	$n_{\mathrm{flagnumber}}$ →	0/1
FC?C	Tests whether the specified system or user flag is clear, returns a corresponding test result, and then clears the flag.	↰ PRG TEST	$n_{\mathrm{flagnumber}}$ →	0/1
FFT	Computes the one- or two-dimensional discrete Fourier transform of an array.	↰ MTH FFT	[*array* $]_1$ →	[*array* $]_2$
FILER	Opens File Manager.	↰ FILES		
FINDALARM	Returns the alarm index n_{index} of the first alarm due after the specified time.	↱ TIME TOOLS ALRM	*date* →	n_{index}
FINISH	Terminates Kermit Server mode in a connected device.	CAT		
FIX	Sets the number display format to fix mode, which rounds the display to *n* decimal places.	CAT	n →	

* = function

Name	Description	Access	Inputs	Outputs
FLASHEVAL	Evaluates unnamed flash functions.	CAT	$\#n_{\text{function}}$	→
FLOOR	Returns the greatest integer that is less than or equal to the argument.*	↰ MTH REAL	x	→ n
→FONT	Returns the current system font.	CAT		→ *obj*
FONT6	Returns the system FONT 6 object.	CAT		→ *obj*
FONT7	Returns the system FONT 7 object.	CAT		→ *obj*
FONT8	Returns the system FONT 8 object.	CAT		→ *obj*
FONT→	Sets the system font.*	CAT	*obj*	→
FOR	Starts FOR … NEXT and FOR … STEP definite loop structures.	↰ PRG BRANCH	*FOR* x_{start} x_{finish}	→
FOURIER	Returns the n^{th} coefficient of a complex Fourier series expansion.*	↰ CALC DERIV. & INTEG	'$symb_1$' z_1	→ z_2
FP	Returns the fractional part of the argument.*	↰ MTH REAL	x	→ y
FREEZE	Freezes the specified part of the display so that it is not updated until a key is pressed.	↰ PRG OUT	$n_{\text{displayarea}}$	→
FROOTS	For a rational polynomial, returns an array of its roots and poles, with their corresponding multiplicities.	↰ ARITH POLYNOMIAL	'$symb_1$'	→ [[$array_1$]]

* = function

Name	Description	Access	Inputs	Outputs
FS?	Tests whether the specified system or user flag is set, and returns a corresponding test result.	↰ PRG TEST	$n_{\text{flagnumber}}$	→ 0/1
FS?C	Tests whether the specified system or user flag is set, returns a corresponding test result, then clears the flag.	↰ PRG TEST	$n_{\text{flagnumber}}$	→ 0/1
FUNCTION	Sets the plot type to FUNCTION.	CAT		
FXND	Splits an object into a numerator and a denominator.	CAT	‘$symb_1$’	→ ‘$symb_2$’ ‘$symb_3$’
GAUSS	Returns the diagonal representation of a quadratic form.	↰ MATRICES QUADRATIC FORM	‘$symb_1$’ [$vector_1$]	→ [[$array_1$]] [[$array_2$]] ‘$symb_2$’ { $list$ }
GCD	Returns the greatest common divisor of two objects.	↰ ARITH POLYNOMIAL	‘$symb_1$’ ‘$symb_2$’	→ z
GCDMOD	Finds the greatest common divisor of two polynomials modulo the current modulus.*	↰ ARITH MODULO	‘$symb_1$’ ‘$symb_2$’	→ ‘$symb_3$’
GET	Retrieves the specified object from a matrix, a list, or an array.	↰ PRG LIST ELEMENTS	[[$matrix$]] n_{position}	→ z_{get}
GETI	Retrieves the specified object from a matrix, a list, or an array, and the index of the next object.	↰ PRG LIST ELEMENTS	[[$matrix$]] $n_{\text{position1}}$	→ [[$matrix$]] $n_{\text{position2}}$ z_{get}
GOR	Superimposes $grob_1$ onto $grob_{\text{target.}}$	↰ PRG GROB	$grob_{\text{target}}$ { #n #m } $grob_1$	→ $grob_{\text{result}}$
GRAD	Sets Grads angle mode.	MODE		
GRIDMAP	Sets the plot type to GRIDMAP.	CAT		

* = function

Name	Description	Access	Inputs	Outputs
→GROB	Creates a graphics object from a specified object, where the argument $n_{\text{char size}}$ specifies the size of the object.	CAT	obj n_{charsize}	→ $grob$
GROBADD	Combines two graphic objects.	CAT	$grob_1$ $grob_2$	→ $grob_3$
GXOR	Superimposes $grob_1$ onto $grob_{\text{target}}$.	↰ PRG GROB	$grob_{\text{target}}$ { #n #m } $grob_1$	→ $grob_{\text{result}}$
HADAMARD	Performs an element by element multiplication of two matrices (Hadamard product).	↰ MATRICES OPERATIONS	[[$matrix_1$]] [[$matrix_2$]]	→ [[$matrix_3$]]
HALFTAN	Replaces sin(x), cos(x) and tan(x) terms with tan(x/2) terms.	↱ TRIG	'$symb_1$'	→ '$symb_2$'
HALT	Halts program execution.	↰ PRG RUN & DEBUG		
HEAD	Returns the first element of a list or string.	↰ PRG CHARS	{ obj_1 ... obj_n }	→ obj_1
HEADER→	Returns the size, in lines, of the display header.	CAT		→ z
→HEADER	Sets the size, in lines, of the display header.	CAT	z	→
HERMITE	Returns the nth Hermite polynomial.*	↰ ARITH POLYNOMIAL	z	→ '$symb_1$'
HESS	Returns the Hessian matrix and the gradient of an expression with respect to the specified variables.	↰ CALC DERIV & INTEG	'$symb_1$' [$vector_1$]	→ [[$matrix$]] z [$vector_2$]
HEX	Sets hexadecimal base for binary integer operations.	CAT		

* = function

Name	Description	Access	Inputs	Outputs
HILBERT	Returns a square Hilbert matrix of the specified order.	↰ MATRICES CREATE	z	→ [[*matrix*]]
HISTOGRAM	Sets the plot type to HISTOGRAM.	CAT		
HISTPLOT	Plots a frequency histogram.	CAT		
→HMS	Converts a real number representing hours or degrees with a decimal fraction to hours-minutes-seconds format.	↱ TIME TOOLS	x	→ *HMS*
HMS–	Returns the difference of two real numbers, where the arguments and the result are interpreted in hours-minutes-seconds format.	↱ TIME TOOLS	HMS_1 HMS_2	→ $HMS_1 - HMS_2$
HMS+	Returns the sum of two real numbers, where the arguments and the result are interpreted in hours-minutes-seconds format.	↱ TIME TOOLS	HMS_1 HMS_2	→ $HMS_1 + HMS_2$
HMS→	Converts a real number in hours-minutes-seconds format to its decimal form.	↱ TIME TOOLS	*HMS*	→ x
HOME	Makes the *HOME* directory the current directory.	CAT		
HORNER	Executes a Horner scheme on a polynomial.	↰ ARITH POLYNOMIAL	‘$symb_1$’ z_1	→ ‘$symb_2$’ z_2 z_3
i	Returns the symbolic constant *i* or its numerical representation, (0, 1).*	↰ *i*		→ ‘i’

* = function

Name	Description	Access	Inputs	Outputs
IABCUV	Returns a solution in integers *u* and *v* of *au* + *bv* = *c*, where *a*, *b*, and *c* are integers.	↰ ARITH INTEGER	n_1 n_2 n_3 →	z_1 z_2
IBERNOULLI	Returns the *n*th Bernoulli number for a given integer.*	CAT	n_1 →	z_1
IBP	Performs integration by parts on a function.	↰ CALC DERIV & INTEG	'$symb_1$' '$symb_2$' →	'$symb_3$' '$symb_4$'
ICHINREM	Solves a system of two congruences in integers using the Chinese Remainder theorem.	↰ ARITH INTEGER	[$vector_1$] [$vector_2$] →	[$vector_3$]
IDIV2	For two integers, *a* and *b*, returns the integer part of *a*/*b*, and the remainder, *r*.	↰ ARITH INTEGER	n_1 n_2 →	n_3 n_4
IDN	Returns an identity matrix.	↰ MATRICES CREATE	*n* →	[[$R\text{-}matrix_{\text{identity}}$]]
IEGCD	Given two integers *x* and *y*, returns three integers, *a*, *b*, and *c*, such that: *ax* + *by = c.*	↰ ARITH INTEGER	n_1 n_2 →	n_3 n_4 n_5
IF	Starts IF … THEN … END and IF … THEN … ELSE … END conditional structures.	↰ PRG BRANCH		
IFERR	Starts IFERR … THEN … END and IFERR … THEN … ELSE … END error trapping structures.	↰ PRG ERROR IFERR		
IFFT	Computes the one- or two-dimensional inverse discrete Fourier transform of an array.	↰ MTH FFT	[*array* $]_1$ →	[*array* $]_2$
IFT	Executes *obj* if *T/F* is nonzero; discards *obj* if *T/F* is zero.	↰ PRG BRANCH	*T/F* *obj* →	

* = function

Name	Description	Access	Inputs	Outputs
IFTE	Executes the *obj* in argument 2 or level 2 if *T/F* is nonzero. Executes the *obj* in argument 3 or level 1 if *T/F* is zero.*	↰ PRG BRANCH	T/F obj_{true} obj_{false}	→
ILAP	Returns the inverse Laplace transform of an expression. The expression must evaluate to a rational fraction.*	↰ CALC DIFFERENTIAL EQNS	'$symb_1$'	→ '$symb_2$'
IM	Returns the imaginary part of its complex argument.*	↱ CMPLX	x	→ 0
INCR	Takes a variable, adds 1, stores the new value back into the original variable, and returns the new value.	↰ PRG MEMORY ARITHMETIC	'*name*'	→ $x_{\text{increment}}$
INDEP	Specifies the independent variable and its plotting range.	CAT	'*global*'	→
INFORM	Creates a user-defined input form (dialog box).	↰ PRG IN	"*title*" { s_1, s_2, ... s_n } *format* { *resets* } { *init* }	→ { *vals* } 1
INPUT	Prompts for data input to the command line and halts stack or command line operations.	↰ PRG IN	"*stack prompt*" "*command-line prompt*"	→ "*result*"
INT	Calculates the antiderivative of a function for a given variable, at a given point.*	CAT	'$symb_1$' '$symb_2$' '$symb_3$'	→ '$symb_4$'
INTVX	Finds the antiderivative of a function symbolically, with respect to the current default variable.*	↰ CALC DERIV. & INTEG	'$symb_1$'	→ '$symb_2$'
INV	Returns the reciprocal or the matrix inverse.*	1/x	z	→ $1/z$
INVMOD	Performs modular inversion on an object modulo the current modulus.*	↰ ARITH MODULO	obj_1	→ obj_1

* = function

Name	Description	Access	Inputs	Outputs
IP	Returns the integer part of the argument.*	↰ MTH REAL	x	→ n
IQUOT	Returns the integer quotient of two integers.*	↰ ARITH INTEGER	n_1 n_2	→ n_3
IREMAINDER	Returns the remainder of an integer division.*	CAT	n_1 n_2	→ n_3
ISOL	Returns an algebraic $symb_2$ that rearranges $symb_1$ to isolate the first occurrence of variable *global*.	↰ S.SLV	‘$symb_1$’ ‘*global*’	→ ‘$symb_2$’
ISPRIME?	Tests if a number is prime.*	↰ ARITH INTEGER	obj_1	→ *T/F*
I→R	Converts an integer into a real number.*	CAT	n	→ z
JORDAN	Computes the eigenvalues, eigenvectors, minimum polynomial, and characteristic polynomial of a matrix.	↰ MATRICES EIGENVECTORS	[[$matrix_1$]]	→ ‘$symb_1$’ ‘$symb_2$’ { $list_1$ } [[$array_1$]]
KERRM	Returns the text of the most recent Kermit error packet.	CAT		→ “*error message*”
KEY	Suspends program execution until a key is pressed, then returns the row-column location x_{nm} of that key.	↰ PRG IN		→ $x_{n\,m}$ 1
KEYEVAL	Actions the specified key press.	CAT	$rc.p_1$	→
→KEYTIME	Sets a new keytime value, or the time in ticks after a keypress until another key is actioned.	CAT	*time*	→
KEYTIME→	Displays the current keytime value.	CAT		→ *time*
KGET	Used by a local Kermit to get a Kermit server to transmit the named object(s).	CAT	‘*name*’	→

* = function

Name	Description	Access	Inputs	Outputs
KILL	Cancels all currently halted programs. If KILL is executed within a program, that program is also canceled.	↰ PRG RUN & DEBUG		
LABEL	Labels axes in *PICT* with variable names and with the minimum and maximum values of the display ranges.	CAT		
LAGRANGE	Returns the interpolating polynomial of minimum degree for a pair of values.	↰ ARITH POLYNOMIAL	[[$matrix_1$]]	→ '$symb_1$'
LANGUAGE→	Returns a value indicating the message language.	CAT		→ *z*
→LANGUAGE	Sets the language used in messages.	CAT	*z*	→
LAP	Performs a Laplace transform on an expression with respect to the current default variable.*	↰ CALC DIFFERENTIAL EQNS	'$symb_1$'	→ '$symb_2$'
LAPL	Returns the Laplacian of a function with respect to a vector of variables.	↰ CALC DERIV & INTEG	'$symb_1$' [$vector_1$]	→ '$symb_2$'
LASTARG	Returns copies of the arguments of the most recently executed command.*	↰ PRG ERROR		→ obj_n … obj_1
→LCD	Displays the specified graphics object with its upper left pixel in the upper left corner of the display.	CAT	*grob*	→
LCD→	Returns the current stack and menu display as a 131 × 64 graphics object.	↰ PRG GROB		→ *grob*
LCM	Returns the least common multiple of two objects.*	↰ ARITH POLYNOMIAL	'$symb_1$' '$symb_2$'	→ '$symb_3$'

* = function

Name	Description	Access	Inputs	Outputs
LCXM	From a program with two arguments, builds a matrix with the specified number of rows and columns, with $a_{ij} = f(i,j)$.	(CAT)	n_1 n_2 «*program*»	→ [[$matrix_1$]]
LDEC	Solves a linear differential equation with constant coefficients.	↰ (S.SLV)	'$symb_1$' '$symb_2$'	→ '$symb_3$'
LEGENDRE	Returns the *n*th degree Legendre polynomial.*	↰ (ARITH) POLYNOMIAL	n_1	→ '$symb_1$'
LGCD	Returns the greatest common divisor of a list of expressions or values.*	↰ (ARITH)	{$list_1$}	→ {$list_1$} z_1
LIBEVAL	Evaluates unnamed library functions.	(CAT)	#n_{function}	→
LIBS	Lists the title, number, and port of each library attached to the current directory.	(CAT)		→ {"*title*" n_{lib} n_{port} …"*title*" n_{lib} n_{port} }
LIMIT	Returns the limit of a function as it approaches a specified value.*	↰ (CALC) LIMITS & SERIES	'$symb_1$' '$symb_2$'	→ '$symb_3$'
LIN	Linearizes expressions involving exponential terms.	↰ (EXP&LN)	'$symb_1$'	→ '$symb_2$'
LINE	Draws a line in *PICT* between the input coordinates.	↰ (PRG) PICT	(x_1, y_1) (x_2, y_2)	→
ΣLINE	Returns an expression representing the best fit line according to the current statistical model.	(CAT)		→ '$symb_{\text{formula}}$'
LINFIT	Stores LINFIT in the reserved variable Σ*PAR*. Subsequent executions of LR will use the linear curve fitting model.	(CAT)		

* = function

Name	Description	Access	Inputs	Outputs
LININ	Tests whether an algebraic is structurally linear for a given variable.*	↰ PRG TEST	‘*symb*’ ‘*name*’	→ 0/1
LINSOLVE	Solves a system of linear equations.	↰ S.SLV	[[$array_1$]] [$vector_1$]	→ ‘$symb_1$’ { $list_1$ } ‘$symb_2$’
ΣLIST	Returns the sum of the elements in a list.	↰ MTH LIST	{ *list* }	→ *z*
ΔLIST	Returns the first differences of the elements in a list.	↰ MTH LIST	{ *list* }	→ { *differences* }
ΠLIST	Returns the product of the elements in a list.	↰ MTH LIST	{ *list* }	→ *z*
→LIST	Takes *n* specified objects and returns a list of those objects.	CAT	obj_1 … obj_n *n*	→ { obj_1 … obj_n }
LIST→	Takes a list of *n* objects and returns each object separately, and returns the total number of objects to item.	CAT	{ obj_1 ...obj_n }	→ obj_1 … obj_n *n*
LN	Returns the natural (base *e*) logarithm of the argument.*	↱ LN	*z*	→ ln *z*
LNAME	Returns the variable names in a symbolic expression.	CAT	‘$symb_1$’	→ [$vector_1$]
LNCOLLECT	Simplifies an expression by collecting logarithmic terms.	↱ ALG	‘$symb_1$’	→ ‘$symb_2$’
LNP1	Returns ln (*x* + 1).*	↰ MTH HYPERBOLIC	*x*	→ ln (*x* + 1)
LOG	Returns the common logarithm (base 10) of the argument.*	↱ log	*z*	→ log *z*
LOGFIT	Stores LOGFIT in Σ*PAR*. Subsequent executions of LR will use the log curve-fitting model.	CAT		

* = function

Name	Description	Access	Inputs	Outputs
LQ	Returns the LQ factorization of an $m \times n$ matrix.	↰ MATRICES FACTORIZATION	[[*matrix*]]$_A$	→ [[*matrix*]]$_L$ [[*matrix*]]$_Q$ [[*matrix*]]$_P$
LR	Uses currently selected statistical model to calculate the linear regression coefficients (intercept and slope).	CAT		→ *Intercept:* x_1 *Slope:* x_2
LSQ	Returns the minimum norm least squares solution to any system of linear equations where $A \times X = B$.	↰ MATRICES OPERATIONS	[*array*]$_B$ [[*matrix*]]$_A$	→ [*array*]$_x$
LU	Returns the LU decomposition of a square matrix.	↰ MATRICES FACTORIZATION	[[*matrix*]]$_A$	→ [[*matrix*]]$_L$ [[*matrix*]]$_U$ [[*matrix*]]$_P$
LVAR	Returns a list of variables in an algebraic object.	CAT	obj_1	→ obj_2 [$vector_1$]
MAD	Returns details of a square matrix.	↰ MATRICES OPERATIONS	[[$array_1$]]	→ '$symb_1$' '$symb_2$' [[$matrix_1$]] '$symb_3$'
MAIN	Displays a menu of CAS categories.	CAT		
MANT	Returns the mantissa of the argument.*	↰ MTH REAL	x	→ y_{mant}
MAP	Applies a specified program to a list of objects or values.	CAT	{ $list_1$ } «*program*»	→ { $list_2$ }
↑MATCH	Rewrites an expression that matches a specified pattern.	CAT	'$symb_1$' { '$symb_{\text{pat}}$', '$symb_{\text{repl}}$' }	→ '$symb_2$' 0/1
↓MATCH	Like ↑MATCH, but works top-down not bottom-up.	CAT	'$symb_1$' { '$symb_{\text{pat}}$', '$symb_{\text{repl}}$' }	→ '$symb_2$' 0/1
MATR	Displays a menu of matrix commands.	CAT		
MAX	Returns the greater of two inputs.*	↰ MTH REAL	x y	→ *max*(x,y)

* = function

Name	Description	Access	Inputs	Outputs
MAXR	Returns the symbolic constant MAXR or its numerical representation 9.99999999999E499.*	↰ MTH CONSTANTS		→ '*MAXR*'
MAXΣ	Finds the maximum coordinate value in each of the *m* columns of the current statistical matrix.	CAT		→ x_{max}
MCALC	Designates a variable as a calculated variable for the multiple-equation solver.	CAT	*'name'*	→
MEAN	Returns the mean of each of the *m* columns of coordinate values in the current statistics matrix.	CAT		→ x_{mean}
MEM	Returns the number of bytes of available RAM.	↰ PRG MEMORY		→ *x*
MENU	Displays a built-in menu or a library menu, or defines and displays a custom menu.	CAT	x_{menu}	→
MENUXY	Displays a function key menu of the computer algebra commands in the specified range.	CAT	n_1 n_2	→ "$string_1$"
MIN	Returns the lesser of two inputs.*	↰ MTH REAL	*x* *y*	→ *min*(*x*,*y*)
MINIFONT→	Returns the font used as the minifont.	CAT		→ *obj*
→MINIFONT	Sets the font used as the minifont.	CAT	*obj*	→
MINIT	Creates the reserved variable *MPAR*, which includes the equations in *EQ* and the variables in these equations. (Used by the multiple-equation solver.)	CAT		

* = function

Name	Description	Access	Inputs	Outputs
MINR	Returns the symbolic constant MINR or its numerical representation, 1.00000000000E–499.*	↰ MTH CONSTANTS		→ '*MINR*'
MINΣ	Finds the minimum coordinate value in each of the *m* columns of the current statistics matrix.	CAT		→ x_{min}
MITM	Changes multiple equation menu titles and order.	CAT	*"title"* { *list* }	→
MOD	Returns a remainder where *x* mod *y* = *x* – *y* floor (*x*/*y*).*	↰ MTH REAL	*x y*	→ *x* mod *y*
MODSTO	Changes the modulo setting to the specified number.	↰ ARITH MODULO	z_1	→ z_2
MROOT	Uses the multiple-equation solver to solve for one or more variables using the equations in *EQ*.	CAT	*'name'*	→ *x*
MSGBOX	Creates a user-defined message box.	↰ PRG OUT	"*message*"	→
MSOLVR	Displays the multiple-equation solver variable menu for the set of equations stored in *EQ*.	CAT		
MULTMOD	Performs modular multiplication of two objects, modulo the current modulus.*	↰ ARITH MODULO	obj_1 obj_2	→ obj_3
MUSER	Designates a variable as user-defined for the multiple-equation solver.	CAT	*'name'*	→
→NDISP	Sets the number of lines over which an object is displayed.	CAT	*n*	→
NDIST	Returns the normal probability distribution at *x* based on the mean *m* and variance *v* of the normal distribution.	↰ MTH PROBABILITY	*m v x*	→ *ndist(m, v, x)*

* = function

Name	Description	Access	Inputs	Outputs
NDUPN	Duplicates an object *n* times, and returns *n.*	CAT	*obj n*	→ *obj … obj n*
NEG	Changes the sign or negates an object.*	↱ CMPLX	*z*	→ –*z*
NEWOB	Creates a new copy of the specified object.	↰ PRG MEMORY	obj_1	→ obj_1
NEXT	Ends definite loop structures.	↰ PRG BRANCH		
NEXTPRIME	Returns the next prime number greater than a specified integer.*	↰ ARITH INTEGER	n_1	→ n_2
NIP	Drops the item on level 2 of the stack.	↰ PRG STACK	obj_1 obj_2	→ obj_2
NOT	Returns the one's complement or logical inverse of the argument.*	↰ PRG TEST	$\#n_1$	→ $\#n_2$
NOVAL	Place holder for reset and initial values in user-defined dialog boxes. NOVAL is returned when a field is empty.	↰ PRG IN		→ NOVAL
NΣ	Returns the number of rows in the current statistical matrix.	CAT		→ n_{rows}
NSUB	Provides access to the current sub-list position during an iteration of a program or command applied using DOSUBS.	↰ PRG LIST PROCEDURES		→ n_{position}
NUM	Returns the code of the first character in a string.	↰ PRG TYPE	"*string*"	→ *n*
→NUM	Converts an exact value to its approximate equivalent.	↱ →NUM	n_1	→ n_2

* = function

Name	Description	Access	Inputs	Outputs
NUMX	Sets the number of *x*-steps for each *y*-step in 3D perspective plots.	CAT	n_x	→
NUMY	Sets the number of *y*-steps across the view volume in 3D perspective plots.	CAT	n_y	→
OBJ→	Separates an object into its components.	↰ PRG TYPE	(x, y)	→ x y
OCT	Selects octal base for binary integer operations.	CAT		
OFF	Turns off the calculator.	↱ OFF		
OPENIO	Opens a serial port using the I/O parameters in the reserved variable *IOPAR.*	CAT		
OR	Returns the logical OR of two arguments.*	↱ BASE BASE LOGIC	$\#n_1$ $\#n_2$	→ $\#n_3$
ORDER	Reorders the variables in the current directory (shown in the VAR menu) to the order specified.	↰ PRG MEMORY DIRECTORY	{ $global_1$ ... $global_n$ }	→
OVER	Returns a copy to level 1 of the object on level 2.	↰ PRG STACK	obj_1 obj_2	→ obj_1 obj_2 obj_1
PA2B2	Takes a prime number and returns a Gaussian integer.	↰ ARITH INTEGER	z_1	→ z_2
PARAMETRIC	Sets the plot type to PARAMETRIC.	CAT		
PARITY	Sets the parity value in the reserved variable *IOPAR.*	CAT	n_{parity}	→
PARSURFACE	Sets plot type to PARSURFACE.	CAT		

* = function

Name	Description	Access	Inputs	Outputs
PARTFRAC	Performs partial fraction decomposition on a partial fraction.	↰ ARITH POLYNOMIAL	'$symb_1$'	→ '$symb_2$'
PATH	Returns a list specifying the path to the current directory.	↰ PRG MEMORY DIRECTORY		→ { *HOME directory-name*$_n$ … *directory-name*$_n$ }
PCAR	Returns the characteristic polynomial of an $n \times n$ matrix.	↰ MATRICES EIGENVECTORS	[[$matrix_1$]]	→ '$symb_1$'
PCOEF	Returns the coefficients of a monic polynomial having specific roots.	↰ ARITH POLYNOMIAL	[*array*]$_{\text{roots}}$	→ [*array*]$_{\text{coefficients}}$
PCONTOUR	Sets the plot type to PCONTOUR.	CAT		
PCOV	Returns the population covariance of the independent and dependent data columns in the current statistics matrix.	CAT		→ $x_{\text{pcovariance}}$
PDIM	Replaces *PICT* with a blank *PICT* of the specified dimensions.	↰ PRG PICT	$(x_{\text{min}}, y_{\text{min}})$ $(x_{\text{max}}, y_{\text{max}})$	→
PERM	Returns the number of possible permutations of *n* items taken *m* at a time.*	↰ MTH PROBABILITY	*n* *m*	→ $P_{n,m}$
PEVAL	Evaluates an *n*-degree polynomial at *x.*	CAT	[*array*]$_{\text{coefficients}}$ *x*	→ $p(x)$
PGDIR	Purges the named directory.	↰ PRG MEMORY DIRECTORY	'*global*'	→
PICK	Copies the contents of a specified level to level 1.	↰ PRG STACK	obj_n … obj_1 *n*	→ obj_n … obj_1 obj_n

* = function

Name	Description	Access	Inputs	Outputs
PICK3	Duplicates the object on level 3 of the stack.	CAT	obj_1 obj_2 obj_3	→ obj_1 obj_2 obj_3 obj_1
PICT	Puts the name PICT on the stack.	⬅ PRG PICT		→ *PICT*
PICTURE	Selects the Picture environment.	CAT		
PINIT	Initializes all currently active ports.	CAT		
PIXOFF	Turns off the pixel at the specified coordinate in *PICT.*	⬅ PRG PICT	(*x*,*y*)	→
PIXON	Turns on the pixel at the specified coordinate in *PICT.*	⬅ PRG PICT	(*x*,*y*)	→
PIX?	Tests whether the specified pixel in *PICT* is on.	⬅ PRG PICT	(*x*,*y*)	→ 0/1
PKT	Used to send command “packets” (and receive requested data) to a Kermit server.	CAT	“*data*” “*type*”	→ “*response*”
PLOTADD	Adds a function to the plot function list.	CAT	‘$symb_1$’	→
PMAX	Specifies (*x*, *y*) as the coordinates of the upper right corner of the display.	CAT	(*x*,*y*)	→
PMIN	Specifies (*x*, *y*) as the coordinates of the lower left corner of the display.	CAT	(*x*,*y*)	→
POLAR	Sets the plot type to POLAR.	CAT		
POS	Returns the position of a substring within a string or the position of an object within a list.	⬅ PRG CHARS	“*string*” “*substring*”	→ *n*

Name	Description	Access	Inputs	Outputs
POWMOD	Raises an object (number or expression) to the specified power, and expresses the result modulo the current modulus.*	↰ ARITH MODULO	obj_1 z_1	→ obj_2
PR1	Prints an object in multiline printer format.	CAT		
PREDV	Returns the predicted dependent-variable value $y_{\text{dependent}}$, based on $x_{\text{independent}}$, the selected statistical model, and the current regression coefficients in Σ*PAR.*	CAT	$x_{\text{independent}}$	→ $y_{\text{dependent}}$
PREDX	Returns the predicted independent-variable value $x_{\text{independent}}$ based on: $y_{\text{dependent}}$, the selected statistical model, and the current regression coefficients in Σ*PAR.*	CAT	$y_{\text{dependent}}$	→ $x_{\text{independent}}$
PREDY	Returns the predicted dependent-variable value based on $x_{\text{independent}}$, the selected statistical model, and the current regression coefficients in Σ*PAR.* Same as PREDV.	CAT	$x_{\text{independent}}$	→ $y_{\text{dependent}}$
PREVAL	Relative to the current default variable, returns the difference between the values of a function at two specified values.*	↰ CALC DERIV. & INTEG	'$symb_1$' z_1 z_2	→ '$symb_2$'
PREVPRIME	Given an integer, finds the closest prime number less than the integer.*	↰ ARITH INTEGER	n_1	→ n_2
PRLCD	Prints a pixel-by-pixel image of the current display (excluding the annunciators).	CAT		

* = function

Name	Description	Access	Inputs	Outputs
PROMPT	Displays the contents of “*prompt*” in the status area, and halts program execution.	⬅ PRG IN	“*prompt*”	→
PROMPTSTO	Creates a variable with the specified name, prompts for a value, and stores the value you enter in the variable.	CAT	“*global*”	→
PROOT	Returns all roots of an *n*-degree polynomial having real or complex coefficients.	⬅ ARITH POLYNOMIAL	[*array*]$_{\text{coefficients}}$	→ [*array*]$_{\text{roots}}$
PROPFRAC	Splits an improper fraction into an integer and a fraction.	⬅ ARITH	‘$symb_1$’	→ ‘$symb_2$’
PRST	Prints all objects on the stack, starting with the object on the highest level.	CAT		
PRSTC	Prints in compact form all objects on the stack, starting with the object on the highest level.	CAT		
PRVAR	Searches the current directory path or port for the specified variables and prints the name and contents of each variable.	CAT	‘*name*’	→
PSDEV	Calculates the population standard deviation of each of the *m* columns of coordinate values in Σ*DAT.*	CAT		→ x_{psdev}
Psi	Calculates the digamma function in one point.*	CAT	‘$symb_1$’ n	→ ‘$symb_2$’
PSI	Calculates the polygamma function in one point.*	CAT	‘$symb_1$’	→ ‘$symb_2$’
PTAYL	Returns the Taylor polynomial for a specified polynomial.*	⬅ ARITH POLYNOMIAL	‘$symb_1$’ z_1	→ ‘$symb_2$’

* = function

Name	Description	Access	Inputs	Outputs
PURGE	Purges the named variables or empty subdirectories from the current directory.	↰ PRG MEMORY	'*global*'	→
PUT	Replaces the object at a specified position in an array.	↰ PRG LIST ELEMENTS	$[[\ matrix\ ]]_1\ \ n_{position}\ \ z_{put}$	→ $[[\ matrix\ ]]_2$
PUTI	As for PUT (see above) but also increments the position.	↰ PRG LIST ELEMENTS	$[[\ matrix\ ]]_1\ \ n_{position1}\ \ z_{put}$	→ $[[\ matrix\ ]]_2\ \ n_{position2}$
PVAR	Calculates the population variance of the coordinate values in each of the *m* columns in Σ*DAT*.	CAT		→ $x_{pvariance}$
PVARS	Returns a list of the backup objects and library objects in a specified port, and the available memory.	CAT	n_{port}	→ { $:n_{port}:\ name_{backup}$ … } *memory*
PVIEW	Displays *PICT* with the specified coordinates at the upper left corner of the graphics display.	↰ PRG PICT	(*x*,*y*)	→
PWRFIT	Stores PWRFIT in Σ*PAR*, so that subsequent executions of LR use the power curve fitting model.	CAT		
PX→C	Converts the specified pixel coordinates to user-unit coordinates.	↰ PRG PICT	{ $\#_n$, $\#_s$ }	→ (*x*,*y*)
→Q	Returns a rational form of the argument.	CAT	*x*	→ '*a*/*b*'
QR	Returns the QR factorization of an *m* × *n* matrix.	↰ MATRICES FACTORIZATION	$[[\ matrix\ ]]_A$	→ $[[\ matrix\ ]]_Q\ \ [[\ matrix\ ]]_R\ \ [[\ matrix\ ]]_P$
QUAD	Finds zeros of an expression equated to 0, or solves an equation. Same as SOLVE.	CAT	'$symb_1$' '*global*'	→ '$symb_2$'

* = function

Name	Description	Access	Inputs	Outputs
QUOT	Returns the quotient part of the Euclidean division of two polynomials.	↰ ARITH POLYNOMIAL	‘$symb_1$’ ‘$symb_2$’	→ ‘$symb_3$’
QUOTE	Returns unevaluated arguments.*	CAT	obj_1	→ obj_2
QXA	Expresses a quadratic form in matrix form.	CAT	‘$symb_1$’ [$vector_1$]	→ ‘$symb_2$’ [$vector_2$]
→Qπ	Returns a rational form of the argument, *or* a rational form of the argument with π factored out.	CAT	x	→ ‘a/b*π’
RAD	Sets Radians angle mode.	CAT		
RAND	Returns a pseudo-random number generated using a seed value, and updates the seed value.	↰ MTH PROBABILITY		→ x_{random}
RANK	Returns the rank of a rectangular matrix.	↰ MATRICES OPERATIONS	[[*matrix*]]	→ n_{rank}
RANM	Returns a matrix of specified dimensions that contains random integers in the range –9 to 9.	↰ MATRICES CREATE	{ m, n }	→ [[*random matrix*]]$_{\text{m×n}}$
RATIO	Prefix form of / (divide).*	CAT	z_1 z_2	→ z_1/z_2
RCEQ	Returns the unevaluated contents of the reserved variable *EQ* from the current directory.	CAT		→ obj_{EQ}
RCI	Multiplies row n of a matrix (or element n of a vector) by a constant x_{factor}, and returns the modified matrix.	↰ MATRICES CREATE ROW	[[*matrix*]]$_1$ x_{factor} $nv_{\text{row number}}$	→ [[*matrix*]]$_3$

* = function

Name	Description	Access	Inputs	Outputs
RCIJ	Multiplies row *i* of a matrix by a constant x_{factor}, adds this product to row *j* of the matrix, and returns the modified matrix.*	↰ MATRICES CREATE ROW	[[*matrix*]]$_1$ x_{factor} $n_{\text{row i}}$ $n_{\text{row j}}$	→ [[*matrix*]]$_2$
RCL	Returns the unevaluated contents of a specified variable.	↰ RCL	'*name*'	→ *obj*
RCLALARM	Recalls a specified alarm.	↱ TIME TOOLS ALRM	n_{index}	→ { *date time* obj_{action} x_{repeat} }
RCLF	Returns a list of integers representing the states of the system and user flags respectively.	CAT		→ { $\#n_{\text{system}}$ $\#n_{\text{user}}$ $\#n_{\text{system2}}$ $\#n_{\text{user2}}$ }
RCLKEYS	Returns the current user key assignments.	CAT		→ { obj_1, x_{key1}, … obj_{n}, x_{keyn} }
RCLMENU	Returns the number of the currently displayed menu.	CAT		→ x_{menu}
RCLΣ	Returns the statistical matrix from the current directory.	CAT		→ [[*matrix*]]
RCWS	Returns the current wordsize in bits (1 through 64).	↱ BASE		→ *n*
RDM	Rearranges the elements of the argument according to specified dimensions.	↰ MATRICES CREATE	[*vector*]$_1$ { n_{elements} }	→ [*vector*]$_2$
RDZ	Specifies the seed for the RAND command.	↰ MTH PROBABILITY	x_{seed}	→
RE	Returns the real part of the argument.*	↱ CMPLX	(*x*, *y*)	→ *x*
RECN	Prepares the HP 49 to receive a file from another Kermit server device, and to store the file in a specified variable.	CAT	'*name*'	→

* = function

Name	Description	Access	Inputs	Outputs
RECT	Sets the coordinate mode to rectangular.	CAT		
RECV	Instructs the HP 49 to look for a named file on another Kermit server device.	CAT		
REF	Reduces a matrix to echelon form.	↰ MATRICES LINEAR SYSTEMS	[[$matrix_1$]]	→ [[$matrix_2$]]
REMAINDER	Returns the remainder of the Euclidean division of two polynomials.*	↰ ARITH POLYNOMIAL	‘$symb_1$’ ‘$symb_2$’	→ ‘$symb_3$’
RENAME	Renames a variable as specified.	CAT	‘$name_{new}$‘ ‘$name_{old}$‘	→
REORDER	Given a polynomial and variable, reorders the variables in the order of powers set in the CAS modes.*	CAT	‘$symb_1$’ z_1	→ ‘$symb_2$’
REPEAT	Starts a loop clause in aWHILE … REPEAT … END indefinite loop structure.	↰ PRG BRANCH		
REPL	Replaces a portion of the target object with a specified object, beginning at a specified position.	↰ PRG LIST	[[$matrix$]]$_1$ $n_{position}$ [[$matrix$]]$_2$	→ [[$matrix$]]$_3$
RES	Specifies the resolution of mathematical and statistical plots.	CAT	$n_{interval}$	→
RESTORE	Replaces the current *HOME* directory with the specified backup copy previously created by ARCHIVE.	CAT	$:n_{port}: name_{backup}$	→
RESULTANT	Returns resultant of two polynomials of the current variable.*	CAT	‘$symb_1$’ ‘$symb_2$’	→ z_1

* = function

Name	Description	Access	Inputs	Outputs
REVLIST	Reverses the order of the elements in a list.	↰ PRG LIST PROCEDURES	{ obj_n, ... obj_1 }	→ { ob_{j1}, … obj_n }
RISCH	Performs symbolic integration on a function using the Risch algorithm.*	↰ CALC DERIV. & INTEG	'$symb_1$' z_1	→ '$symb_2$'
RKF	Computes solution to an initial value problem for a differential equation using the Runge–Kutta–Fehlberg (4,5) method.	CAT	{ *list* } x_{tol} $x_{T\ final}$	→ { *list* } x_{tol}
RKFERR	Returns the absolute error estimate for a given step *h* when solving an initial value problem for a differential equation.	CAT	{ *list* } *h*	→ { *list* } *h* y_{delta} *error*
RKFSTEP	Computes the next solution step (h_{next}) to an initial value problem for a differential equation.	CAT	{ *list* } x_{tol} *h*	→ { *list* } x_{tol} h_{next}
RL	Rotates a binary integer one bit to the left.	↰ MTH BASE BIT	$\#n_1$	→ $\#n_2$
RLB	Rotates a binary integer one byte to the left.	↰ MTH BASE BYTE	$\#n_1$	→ $\#n_2$
RND	Rounds an object to a specified number of decimal places or significant digits, or to fit the current display format.*	↰ MTH REAL	z_1 n_{round}	→ z_2
RNRM	Returns the row norm (infinity norm) of an array.	↰ MATRICES OPERATIONS	[*array*]	→ $x_{row\ norm}$
ROLL	Moves the contents of a specified level to level 1, and rolls up the portion of the stack beneath the specified level.	↰ PRG STACK	obj_n ... obj_1 *n*	→ obj_{n-1} … obj_1 obj_n

* = function

Name	Description	Access	Inputs		Outputs
ROLLD	Moves the contents of level 2 to a specified level, *n*, and rolls downward the portion of the stack beneath the specified level.	↰ PRG STACK	obj_n ... obj_1 n	→	obj_1 obj_n … obj_2
ROMUPLOAD	Transfers the operating system to another calculator.	CAT			
ROOT	Returns the value of the specified variable *global* for which the specified program or algebraic object most nearly evaluates to zero or a local extremum.	CAT	«*program*» ‘*global*’ *guess*	→	x_{root}
ROT	Rotates the first three objects on the stack, moving the object on level 3 to level 1.	↰ PRG STACK	obj_3 obj_2 obj_1	→	obj_2 obj_1 obj_3
→ROW	Transforms a matrix into a series of row vectors and returns the vectors and a row count.	↰ MATRICES CREATE ROW	[[*matrix*]]	→	$[\ vector\ ]_{row1}$ … $[\ vector\ ]_{row\,n}$ n
ROW–	Deletes row *n* of a matrix (or element *n* of a vector), and returns the modified matrix (or vector) and the deleted row (or element).	↰ MATRICES CREATE ROW	$[[\ matrix\ ]]_1$ n_{row}	→	$[[\ matrix\ ]]_2$ $[\ vector\ ]_{row}$
ROW+	Inserts an array into a matrix at the position indicated by n_{index}, and returns the modified matrix.	↰ MATRICES CREATE ROW	$[[\ matrix\ ]]_1$ $[[\ matrix\ ]]_2$ n_{index}	→	$[[\ matrix\ ]]_3$
ROW→	Transforms a series of row vectors and a row count into a matrix containing those rows.	↰ MATRICES ROW	$[\ vector\ ]_{row1}$…$[\ vector\ ]_{row\,n}$ n	→	[[*matrix*]]
RR	Rotates a binary integer one bit to the right.	↱ BASE BIT	$\#n_1$	→	$\#n_2$
RRB	Rotates a binary integer one byte to the right.	↱ BASE BYTE	$\#n_1$	→	$\#n_2$

* = function

Name	Description	Access	Inputs	Outputs
rref	Reduces a matrix to row-reduced echelon form and returns pivot points.*	CAT	$[[\ \mathit{matrix}_1\ ]]$	→ { *list* } $[[\ \mathit{matrix}_2\ ]]$
RREF	Reduces a matrix to row-reduced echelon form.*	↰ MATRICES LINEAR SYSTEMS	$[[\ \mathit{matrix}_1\ ]]$	→ $[[\ \mathit{matrix}_2\ ]]$
RREFMOD	Performs modular row-reduction to echelon form on a matrix, modulo the current modulus.	CAT	$[[\ \mathit{matrix}_1\ ]]$	→ $[[\ \mathit{matrix}_2\ ]]$
RRK	Computes the solution to an initial value problem for a differential equation with known partial derivatives.	CAT	{ *list* } x_{tol} $x_{\mathrm{T\ final}}$	→ { *list* } x_{tol}
RRKSTEP	Computes the next solution step to an initial value problem for a differential equation, and displays method used.	CAT	{ *list* } x_{tol} *h* *last*	→ { *list* } x_{tol} h_{next} *current*
RSBERR	Returns an error estimate for a given step *h* when solving an initial values problem for a differential equation.	CAT	{ *list* } *h*	→ { *list* } *h* y_{delta} *error*
RSD	Computes the residual B – AZ of the arrays B, A, and Z.	↰ MATRICES OPERATIONS	$[[\ \mathit{matrix}\ ]]_B$ $[[\ \mathit{matrix}\ ]]_A$ $[[\ \mathit{matrix}\ ]]_Z$	→ $[[\ \mathit{matrix}\ ]]_{B-AZ}$
RSWP	Swaps rows *i* and *j* of a matrix and returns the modified matrix.	↰ MATRICES CREATE ROW	$[[\ \mathit{matrix}\ ]]_1$ $n_{\mathrm{row\ i}}$ $n_{\mathrm{row\ j}}$	→ $[[\ \mathit{matrix}\ ]]_2$
R→B	Converts a positive real to its binary integer equivalent.	↱ BASE	*n*	→ *#n*
R→C	Combines two real numbers or real arrays into a single complex number or complex array.	↰ PRG TYPE	*x* *y*	→ (*x*,*y*)
R→D	Converts a real number expressed in radians to its equivalent in degrees.*	↰ MTH REAL	*x*	→ $(180/\pi)x$

* = function

Name	Description	Access	Inputs	Outputs
R→I	Converts a real number to an integer.*	CAT	z_1 →	n_1
SAME	Compares two objects, and returns a true result (1) if they are identical, and a false result (0) if they are not.	↰ PRG TEST	obj_1 obj_2 →	0/1
SBRK	Interrupts serial transmission or reception	CAT		
SCALE	Adjusts first two parameters in *PPAR*, (x_{min}, y_{min}) and (x_{max}, y_{max}), so that x_{scale} and y_{scale} are the new plot horizontal and vertical scales.	CAT	x_{scale} y_{scale} →	
SCALEH	Multiplies the vertical plot scale by $x_{factor.}$	CAT	x_{factor} →	
SCALEW	Multiplies the horizontal plot scale by $x_{factor.}$	CAT	x_{factor} →	
SCATRPLOT	Draws a scatterplot of (*x*, *y*) data points from the specified columns of the current statistics matrix.	CAT		
SCATTER	Sets the plot type to SCATTER.	CAT		
SCHUR	Returns the Schur decomposition of a square matrix.	↰ MATRICES FACTORIZATION	[[*matrix*]]$_A$ →	[[*matrix*]]$_Q$ [[*matrix*]]$_T$
SCI	Sets the number display to scientific mode: one digit left of the fraction mark and *n* significant digits to the right.	CAT	*n* →	
SCLΣ	Adjusts (x_{min}, y_{min}) and (x_{max}, y_{max}) in *PPAR* so that a subsequent scatter plot exactly fills *PICT.*	CAT		

* = function

Name	Description	Access	Inputs	Outputs
SCONJ	Conjugates the contents of a named object.	↰ PRG MEMORY ARITHMETIC	'*name*'	→
SCROLL	Displays the contents of a named object.	CAT	'*name*'	→
SDEV	Calculates the sample standard deviation of each of the *m* columns of coordinate values in Σ*DAT*.	CAT		→ x_{sdev}
SEND	Sends a copy of the named objects to a Kermit device.	CAT	'*name*'	→
SEQ	Returns a list of results generated by repeatedly executing obj_{exec} using *index* between x_{start} to x_{end}, in steps of $x_{incr.}$	↰ PRG LIST PROCEDURES	obj_{exec} *index* x_{start} x_{end} x_{incr}	→ { *list* }
SERIES	For a given function, computes Taylor series, asymptotic development and limit at finite and infinite points.	↰ CALC LIMITS & SERIES	'$symb_1$' '$symb_2$' z_1	→ { $list_1$ } '$symb_3$'
SERVER	Starts Kermit Server mode.	CAT		
SEVAL	Evaluates the variables in an expression and substitutes the values into the expression.*	CAT	'$symb_1$'	→ '$symb_2$'
SF	Sets a specified user or system flag.	↰ PRG TEST	$n_{flag\ number}$	→
SHOW	Returns $symb_2$, which is equivalent to $symb_1$ but with all implicit references to the variable *name* made explicit.	CAT	'$symb_1$' '*name*'	→ '$symb_2$'
SIDENS	Calculates the intrinsic density of silicon as a function of temperature, $x_{T.}$*	CAT	x_T	→ $x_{density}$

* = function

Name	Description	Access	Inputs	Outputs
SIGMA	Calculates the discrete antiderivative of a function with respect to a specified variable.*	CAT	‘$symb_1$’ z_1	→ ‘$symb_2$’
SIGMAVX	Calculates the discrete antiderivative of a function with respect to the current variable.*	CAT	‘$symb_1$’	→ ‘$symb_2$’
SIGN	Returns the sign of a real number.*	↰ MTH REAL	z_1	→ z_2
SIGNTAB	Returns the sign table of a rational function of one variable.	CAT	‘$symb_1$’	→ { $list_1$ }
SIMP2	Simplifies two objects by dividing them by their greatest common divisor.	↰ ARITH	‘$symb_1$’ ‘$symb_2$’	→ ‘$symb_3$’ ‘$symb_4$’
SIN	Returns the sine of the argument.*	SIN	z	→ sin z
SINCOS	Converts complex logarithmic and exponential expressions to expressions with trigonometric terms.	↱ TRIG	‘$symb_1$’	→ ‘$symb_2$’
SINH	Returns the hyperbolic sine of the argument.*	↱ TRIG HYPERBOLIC	z	→ sinh z
SINV	Replaces the contents of a variable with its inverse.*	↰ PRG MEMORY ARITHMETIC	‘*name*’	→
SIZE	Returns the number of characters in a string, elements in a list, dimensions of an array, objects in a unit object or algebraic object, or the dimensions of a graphics object.	↰ PRG CHARS	“*string*”	→ n
SL	Shift a binary integer one bit to the left.*	↰ MTH BASE BIT	$\#n_1$	→ $\#n_2$

* = function

Name	Description	Access	Inputs	Outputs
SLB	Shifts a binary integer one byte to the left.	↰ MTH BASE BYTE	$\#n_1$ →	$\#n_2$
SLOPEFIELD	Sets the plot type to SLOPEFIELD.	CAT		
SNEG	Replaces the contents of a variable with its negative.	↰ PRG MEMORY ARITHMETIC	'*name*' →	
SNRM	Returns the spectral norm of an array.	↰ MATRICES OPERATIONS	[*array*] →	$x_{\text{spectralnorm}}$
SOLVE	Finds zeros of an expression equated to 0, or solves an equation.	↰ S.SLV	'$symb_1$' z_1 →	{ $list_1$ }
SOLVER	Displays a menu of commands used in solving equations.	CAT		
SOLVEVX	Finds zeros of an expression with respect to the current variable.	↰ S.SLV	'$symb_1$' →	{ $list_1$ }
SORT	Sorts the elements in a list in ascending order.	↰ MTH LIST	{ *list* }$_1$ →	{ *list* }$_2$
SPHERE	Sets the coordinate mode to spherical.	CAT		
SQ	Returns the square of the argument.*	↰ x^2	z →	z^2
SR	Shifts a binary integer one bit to the right.	↰ MTH BASE BIT	$\#n_1$ →	$\#n_2$
SRAD	Returns the spectral radius of a square matrix.	↰ MATRICES OPERATIONS	[[*matrix*]]$_{n\times n}$ →	$x_{\text{spectral radius}}$

* = function

Name	Description	Access	Inputs	Outputs
SRB	Shifts a binary integer one byte to the right.	↰ MTH BASE BYTE	$\#n_1$ →	$\#n_2$
SRECV	Returns up to *n* characters from the serial input buffer, with an error digit if an error occurred.	CAT	*n* →	'*string*' 0/1
SREPL	Finds and replaces a string in a text object.	CAT	"$string_1$" "$string_2$" "$string_3$" →	"$string_4$"
START	Begins START … NEXT and START … STEP definite loop structures.	↰ PRG BRANCH	*START* x_{start} x_{finish} →	
STD	Sets the number display format to standard mode.	CAT		
STEP	Defines the increment (step) value, and ends definite loop structure.	↰ PRG BRANCH		
STEQ	Stores an object into the reserved variable *EQ* in the current directory.	CAT	*obj* →	
STIME	Specifies the period that SRECV (serial reception) and XMIT (serial transmission) wait before timing out.*	CAT	x_{seconds} →	
STO	Stores an object into a specified variable or object.	STO▶	*obj* '*name*' →	
STOALARM	Stores an alarm in the system alarm list and returns its alarm index number.	↱ TIME TOOLS ALRM	x_{time} →	n_{index}
STOF	Sets the states of the system flags or the system and user flags.	CAT	$\#n_{\text{system}}$ →	
STOKEYS	Assigns objects to specified keys on the user keyboard.	CAT	{ obj_1, $x_{\text{key 1}}$, ... obj_n, $x_{\text{key n}}$ } →	

* = function

Name	Description	Access	Inputs	Outputs
STO–	Calculates the difference between an object and a variable and stores the object in the variable.	↰ PRG MEMORY ARITHMETIC	*obj* ‘*name*’ →	
STO*	Multiplies the contents of a specified variable by a number or other object.	↰ PRG MEMORY ARITHMETIC	*obj* ‘*name*’ →	
STO/	Calculates the quotient of a number and the contents of a specified variable. Stores new value in the specified variable.	↰ PRG MEMORY ARITHMETIC	*obj* ‘*name*’ →	
STO+	Adds a number or other object to a variable.	↰ PRG MEMORY ARITHMETIC	*obj* ‘*name*’ →	
STOΣ	Stores *obj* in the reserved variable Σ*DAT.*	CAT	*obj* →	
→STR	Converts any object to string form.	CAT	*obj* →	“*string*”
STR→	Evaluates the text of a string as if the text were entered from the command line.	CAT	obj_1 →	obj_2
STREAM	Repeatedly executes *obj* on the first two elements in a list until the list is exhausted. Returns the final result.	↰ PRG LIST PROCEDURES	{ *list* } *obj* →	*result*
STWS	Sets the current binary integer wordsize to *n* bits, where *n* is a value from 1 through 64 (the default is 64).	↰ MTH BASE	*n* →	
SUB	Returns the specified portion of an object.	↰ PRG LIST	“$string_1$” $n_{\text{start position}}$ $n_{\text{end position}}$ →	“$string_2$”
SUBST	Substitutes a value or expression for a variable in an expression.*	↱ ALG	‘$symb_1$’ z_1 →	‘$symb_2$’

* = function

Name	Description	Access	Inputs	Outputs
SUBTMOD	Performs a subtraction, modulo the current modulus.*	↰ ARITH MODULO	obj_1 obj_2	→ obj_3
SVD	Returns the singular value decomposition of an $m \times n$ matrix.	↰ MATRICES FACTORIZATION	[[*matrix*]]$_A$	→ [[*matrix*]]$_U$ [[*matrix*]]$_V$ [*vector*]$_S$
SVL	Returns the singular values of an $m \times n$ matrix.	↰ MATRICES FACTORIZATION	[[*matrix*]]	→ [*vector*]
SWAP	Swaps the position of the two objects.	↰ PRG STACK	obj_1 obj_2	→ obj_2 obj_1
SYLVESTER	For a symmetric matrix A, returns D and P where D is a diagonal matrix and $A = P^T DP$.	CAT	[[*matrix*]]$_A$	→ [[*matrix*]]$_D$ [[*matrix*]]$_P$
SYSEVAL	Evaluates unnamed operating system objects specified by their memory addresses.	CAT	$\#n_{address}$	→
%T	Returns the percent of the first argument that is represented by the second argument.*	↰ MTH REAL	x y	→ $100y/x$
TABVAL	For an expression and a list of values, returns the results of substituting the values for the default variable in the expression.	CAT	'$symb_1$' { $list_1$ }	→ '$symb_1$' { $list_2$ }
TABVAR	For a rational function, computes the turning points and where the function increases or decreases.	CAT	'$symb_1$'	→ '$symb_1$' { $list_1$ } $grob_1$
→TAG	Combines objects to create a tagged object.	CAT	*obj* "*tag*"	→ :*tag*: *obj*
TAIL	Returns all but the first element of a list or string.	↰ PRG CHARS	{ obj_1 ... obj_n }	→ { ob_{j2} … obj_n }

* = function

Name	Description	Access	Inputs	Outputs
TAN	Returns the tangent of the argument.*	TAN	z	→ tan z
TAN2SC	Replaces tan(x) terms with sin(x) and cos(x) terms.	↱ TRIG	'$symb_1$'	→ '$symb_2$'
TAN2SC2	Replaces tan(x) terms with sin(x) and cos(x) terms.	↱ TRIG	'$symb_1$'	→ '$symb_2$'
TANH	Returns the hyperbolic tangent of the argument.*	↱ TRIG HYPERBOLIC	z	→ tanh z
TAYLOR0	Performs a fourth-order Taylor expansion of an expression at $x = 0$.*	↰ CALC LIMITS & SERIES	'$symb_1$'	→ '$symb_2$'
TAYLR	Calculates the nth order Taylor polynomial of *symb* in the variable *global.*	↰ CALC LIMITS & SERIES	'*symb*' '*global*' n_{order}	→ '$symb_{Taylor}$'
TCHEBYCHEFF	Returns the nth Tchebycheff polynomial.*	CAT	n_1	→ '$symb_1$'
TCOLLECT	Linearizes products in a trigonometric expression by collecting and combining sine and cosine terms.	↱ TRIG	'$symb_1$'	→ '$symb_2$'
TDELTA	Calculates a temperature change.*	CAT	x y	→ x_{delta}
TEVAL	For the specified operation, performs the same function as EVAL, and returns the time taken to perform the evaluation.	CAT	obj_1	→ obj_2 *hms*
TEXPAND	Expands transcendental functions.	↰ TRIG	'$symb_1$'	→ '$symb_2$'
TEXT	Displays the stack.	↰ PRG OUT		

* = function

Name	Description	Access	Inputs	Outputs
THEN	Starts the true-clause in a conditional or error-trapping structure.	↰ PRG BRANCH		
TICKS	Returns the system time as a binary integer.	↱ TIME TOOLS TICKS		→ $\#n_{\text{time}}$
TIME	Returns the system time in HH.MMSSs format.	↱ TIME TOOLS		→ *time*
→TIME	Sets the system time.	↱ TIME TOOLS	*time*	→
TINC	Calculates a temperature increment.*	CAT	x_{initial} y_{delta}	→ x_{final}
TLIN	Linearizes and simplifies a trigonometric expression.	↱ TRIG	'$symb_1$'	→ '$symb_2$'
TLINE	For each pixel along the line in *PICT* defined by the specified coordinates, TLINE turns off/on every pixel that is on/off.	↰ PRG PICT	(x_1,y_1) (x_2,y_2)	→
TMENU	Displays a built-in menu, library menu, or user-defined menu.	CAT	x_{menu}	→
TOT	Computes the sum of each of the *m* columns of coordinate values in ΣDAT.	CAT		→ x_{sum}
TRACE	Returns the trace of a square matrix.	↰ MATRICES OPERATIONS	[[*matrix*]]$_{n\times n}$	→ x_{trace}
TRAN	Returns the transpose of a matrix.	↰ MATRICES OPERATIONS	[[*matrix*]]	→ [[*matrix*]]$_{\text{transpose}}$
TRANSIO	Specifies a character translation option in data transfer.	CAT	$n_{\textit{option}}$	→

* = function

Name	Description	Access	Inputs	Outputs
TRIG	Converts complex logarithmic and exponential terms into their equivalent trigonometric terms.	↱ TRIG	‘$symb_1$’	→ ‘$symb_2$’
TRIGCOS	Simplifies a trigonometric expression into cosine terms.	↱ TRIG	‘$symb_1$’	→ ‘$symb_2$’
TRIGO	Displays a menu of trigonometry commands.	CAT		
TRIGSIN	Simplifies a trigonometric expression into sine terms.	↱ TRIG	‘$symb_1$’	→ ‘$symb_2$’
TRIGTAN	Replaces sin() and cos() terms with tan() terms.	↱ TRIG	‘$symb_1$’	→ ‘$symb_2$’
TRN	Returns the conjugate transpose of a matrix.	↰ MTH MATRIX MAKE	[[*matrix*]]	→ [[*matrix*]]$_{\text{transpose}}$
TRNC	Truncates an object to a set number of decimal places or significant digits, or to fit the current display format.*	↰ MTH REAL	z_1 n_{truncate}	→ z_2
TRUNC	Truncates a series expansion.	CAT	‘$symb_1$’ ‘$symb_2$’	→ ‘$symb_3$’
TRUTH	Sets the plot type to TRUTH.	CAT		
TSIMP	Simplifies exponential and logarithmic expressions.	↰ EXP&LN	‘$symb_1$’	→ ‘$symb_2$’
TSTR	Returns a string derived from the date and time.	↱ TIME TOOLS	*date time*	→ “*DOW DATE TIME*“
TVARS	Lists all global variables in the current directory that contain objects of a specified type.	↰ PRG MEMORY DIRECTORY	n_{type}	→ { *global* … }
TVM	Displays the TVM Solver menu.	CAT		

* = function

Name	Description	Access	Inputs	Outputs
TVMBEG	Specifies that TVM calculations treat payments as being made at the beginning of the compounding periods.	CAT		
TVMEND	Specifies that TVM calculations treat payments as being made at the end of the compounding periods.	CAT		
TVMROOT	Solves for the specified TVM variable using values from the remaining TVM variables.	CAT	‘*TVM variable*’	→ $x_{\text{TVM variable}}$
TYPE	Returns the type number of an object.	↰ PRG TEST	obj	→ n_{type}
UBASE	Converts a unit object to SI base units.*	↰ CONVERT UNITS TOOLS	x_unit	→ $y\_base\text{-}units$
UFACT	Factors the level 1 unit from the unit expression of the level 2 unit object.	↰ CONVERT UNITS TOOLS	$x_1\_unit_1$ $x_2\_unit_2$	→ $x_3\_unit_2{*}unit_3$
UFL1→MINIF	Converts a UFL1 (universal font library) fontset to a HP 49G minifont.	CAT	obj_{fontset} n_{ID}	→
→UNIT	Creates a unit object from a real number and a unit expression.	CAT	x y_unit	→ x_unit
UNPICK	Replaces the object at level *n*+2 with the object at level 2 and deletes the objects at levels 1 and 2.*	↰ PRG STACK	obj_{n+2} … obj_4 obj_3 obj_2 n	→ obj_2 … obj_4 obj_3
UNROT	Changes the order of the first three objects on the stack.*	↰ PRG STACK	obj_3 obj_2 obj_1	→ obj_1 obj_3 obj_2

* = function

Name	Description	Access	Inputs	Outputs
UNTIL	Starts the test clause in a DO … UNTIL … END indefinite loop structure.	↰ PRG BRANCH		
UPDIR	Makes the parent of the current directory the new current directory.	↰ UPDIR		
UTPC	Returns the probability that a chi-square random variable is greater than x given n degrees of freedom.	↰ MTH PROBABILITY	$n\ \ x$	$\to utpc(n,x)$
UTPF	Returns the probability that a Snedecor's F random variable is greater than x. n_1 and n_2 are the numerator and denominator degrees of freedom of the F distribution.	↰ MTH PROBABILITY	$n_1\ \ n_2\ \ x$	$\to utpf(n_1,n_2,x)$
UTPN	Returns the probability that a normal random variable is greater than x, where m and v are the mean and variance of the normal distribution.	↰ MTH PROBABILITY	$m\ \ v\ \ x$	$\to utpn(m,v,x)$
UTPT	Returns the probability that a Student's t random variable is greater than x, where n is the degrees of freedom.	↰ MTH PROBABILITY	$n\ \ x$	$\to utpt(n,x)$
UVAL	Returns the numerical part of a unit object.*	↱ UNITS TOOLS	x_unit	$\to x$
→V2	Converts two numbers into a vector or complex number.	↰ MTH VECTOR	$x\ \ y$	$\to [\ x\ y\ ]$
→V3	Converts three numbers into a vector.	↰ MTH VECTOR	$x_1\ \ x_2\ \ x_3$	$\to [\ x_1\ \ x_2\ \ x_3\ ]$
VANDERMONDE	Builds a Vandermonde matrix from a list of objects.	↰ MATRICES CREATE	$\{\ list\ \}$	$\to [[\ matrix\ ]]$

* = function

Name	Description	Access	Inputs	Outputs
VAR	Calculates the sample variance of the coordinate values in each of the *m* columns in Σ*DAT*.	CAT		→ x_{variance}
VARS	Returns a list of the names of all variables in the VAR menu for the current directory.	↰ PRG MEMORY DIRECTORY		→ { $global_1$ … $global_n$ }
VER	Returns the Computer Algebra System version number, and date of release.	CAT		→ "$string_1$"
VERSION	Displays the software version and copyright message.	CAT		→ "*version number*" "*copyright message*"
VISIT	Places the contents of a variable on the command line.	CAT	'*name*'	→
VISITB	Opens the contents of a variable in the most suitable editing environment for the particular type of object.	CAT	'*name*'	→
VTYPE	Returns the type number of the object in the variable.	↰ PRG TYPE	'*name*'	→ n_{type}
V→	Separates a vector or complex number into its component elements.	↰ MTH VECTOR	[*x y*]	→ *x* *y*
WAIT	Suspends program execution for a specified time, or until a key is pressed.	↰ PRG IN	*x*	→
WHILE	Starts a WHILE … REPEAT … END indefinite loop structure.	↰ PRG BRANCH		
WIREFRAME	Sets the plot type to WIREFRAME.	CAT		

* = function

Name	Description	Access	Inputs	Outputs
WSLOG	Returns four strings recording the date, time, and cause of the four most recent warmstart events.	CAT		→ “log$_4$” … “log$_1$”
ΣX	Sums the values in the independent-variable column of the current statistical matrix (reserved variable Σ*DAT*).	CAT		→ x_{sum}
ΣX2	Sums the squares of the values in the independent-variable column of the current statistical matrix.	CAT		→ x_{sum}
XCOL	Specifies the independent-variable column of the current statistics matrix (reserved variable Σ*DAT*).	CAT	n_{col}	→
XGET	Retrieves a file by XMODEM from another calculator.	CAT	‘*name*’	→
XMIT	Sends a string serially without using Kermit and then indicates whether the transmission was successful.	CAT	“*string*”	→ 1
XNUM	Converts an object or a list of objects to approximate numeric format.	CAT	obj_1	→ obj_2
XOR	Returns the logical exclusive OR of two arguments.*	↱ BASE LOGIC	$\#n_1$ $\#n_2$	→ $\#n_3$
XPON	Returns the exponent of the argument.*	↰ MTH REAL	x	→ n_{expon}
XPUT	Sends a file by XMODEM to another calculator.	CAT	‘*name*’	→
XQ	Converts a number, or a list of numbers in decimal format, to rational format.	CAT	z_1	→ z_2
XRECV	Prepares the HP 49 to receive an object via XModem.*		‘*name*’	→

* = function

Name	Description	Access	Inputs	Outputs
XRNG	Specifies the *x*-axis display range.	CAT	x_{min} x_{max} →	
XROOT	Computes the *x*th root of a real number.*	↱ $\sqrt[x]{y}$	y x →	$\sqrt[x]{y}$
XSEND	Sends a copy of the named object via XModem.	CAT	'*name*' →	
XSERVE	Puts the calculator in XMODEM server mode.	CAT		
XVOL	Sets the width of the view volume in *VPAR* (for 3-D plotting).	CAT	x_{left} x_{right} →	
XXRNG	Specifies the *x* range of an input plane (domain) for GRIDMAP and PARSURFACE plots.	CAT	x_{min} x_{max} →	
ΣXY	Sums the products of each of the corresponding values in the independent- and dependent-variable columns of the current statistical matrix.	↱ STAT SUMMARY STATS	→	x_{sum}
ΣY	Sums the values in the dependent variable column of the current statistical matrix (reserved variable Σ*DAT*).	↱ STAT SUMMARY STATS	→	x_{sum}
ΣY2	Sums the squares of the values in the dependent-variable columns of the current statistical matrix.	↱ STAT SUMMARY STATS	→	x_{sum}
YCOL	Specifies the dependent variable column of the current statistics matrix (reserved variable Σ*DAT*).	CAT	n_{col} →	
YRNG	Specifies the *y*-axis display range.	CAT	y_{min} y_{max} →	
YSLICE	Sets the plot type to YSLICE.	CAT		

* = function

Name	Description	Access	Inputs	Outputs
YVOL	Sets the depth of the view volume in *VPAR.*	CAT	y_{near} y_{far} →	
YYRNG	Specifies the y range of an input plane (domain) for GRIDMAP and PARSURFACE plots.		y_{near} y_{far} →	
ZEROS	Returns the zeros of a function of one variable, without multiplicity.	↰ S.SLV	'$symb_1$' z_1 →	z_2
ZFACTOR	Calculates the gas compressibility correction factor for non-ideal behavior of a hydrocarbon gas.*	CAT	x_{Tr} y_{Pr} →	x_{Zfactor}
ZVOL	Sets the height of the view volume in *VPAR.*	CAT	x_{low} x_{high}	
%	Returns *x* percent of *y*.*	↰ MTH REAL	x y →	$xy/100$
+	Returns the sum of the arguments.*	+	z_1 z_2 →	$z_1 + z_2$
–	Returns the difference of the arguments.*	−	z_1 z_2 →	$z_1 - z_2$
!	Returns the factorial *n*! of a positive integer argument *n*, or the gamma function Γ(*x*+1) of a non-integer argument *x*.*	↰ MTH PROBABILITY	n →	$n!$
*	Returns the product of the arguments.*	X	z_1 z_2 →	z_1 z_2
/	Returns the quotient of the arguments: the first argument is divided by the second argument.*	÷	z_1 z_2 →	z_1 / z_2
^	Returns the value of the level 2 object raised to the power of the level 1 object.*	y^x	w z →	w^z

* = function

Name	Description	Access	Inputs	Outputs
|	Where command: substitutes values for names in an expression.*	|	'$symb_{old}$' { $name_1$, '$symb_1$', nam_{e2}, '$symb_2$' … }	→ '$symb_{new}$'
<	Tests whether one object is less than another object.*	↱ <	x y	→ 0/1
>	Tests whether one object is greater than another object.*	↱ >	x y	→ 0/1
≥	Tests whether one object is greater than or equal to another object.*	↰ ≥	x y	→ 0/1
≤	Tests whether one object is less than or equal to another object.*	↰ ≤	x y	→ 0/1
=	Returns an equation formed from the two arguments.*	↱ =	z_1 z_2	→ $z_1 = z_2$
==	Tests if two objects are equal.*	CAT	obj_1 obj_2	→ 0/1
≠	Tests if two objects are not equal.*	↰ ≠	obj_1 obj_2	→ 0/1
√	Returns the (positive) square root of the argument.*	√x	z	→ $\sqrt{z}$
∂	Gives the derivative of an expression, number, or unit object with respect to a specified variable of differentiation.*	↱ ∂	'$symb_1$' '$name$'	→ '$symb_2$'
→	Creates local variables in a program.	↱ →	obj_1 … obj_n	→
π	Returns the symbolic constant 'π' or its numerical representation, 3.14159265359.*	↰ π		→ 'π'
Σ	Calculates the value of a finite series.*	↱ Σ	'$indx$' x_{init} x_{final} $smnd$	→ x_{sum}

* = function

Name	Description	Access	Inputs	Outputs
Σ–	Returns a vector of *m* real numbers (or one number *x* if *m* = 1) corresponding to the coordinate values of the last data point entered by Σ+ into the current statistics matrix.	CAT		→ *x*
Σ+	Adds one or more data points to the current statistics matrix (reserved variable Σ*DAT*).	CAT	*x*	→
∫	Integrates an *integrand* from *lower limit* to *upper limit* with respect to a specified variable of integration.*	↱ ∫	*lower limit* *upper limit* *integrand* ‘*name*’	→ ‘$symb_{\text{integral}}$’

* = function