

Name	Description	Access	Inputs	Outputs
Σ ⁻	Returns a vector of <i>m</i> real numbers (or one number <i>x</i> if <i>m</i> = 1) corresponding to the coordinate values of the last data point entered by Σ ⁺ into the current statistics matrix.			→ <i>x</i>
Σ ⁺	Adds one or more data points to the current statistics matrix (reserved variable ΣDAT).			<i>x</i> →
∫	Integrates an <i>integrand</i> from <i>lower limit</i> to <i>upper limit</i> with respect to a specified variable of integration.*		<i>lower limit</i> <i>upper limit</i> <i>integrand</i> <i>'name'</i>	→ <i>'symb_{negal}'</i>

* = function

HP 49G Pocket Guide

CONTENTS

1	Quick Reference Chart	2
2	Function Key Guide	3
3	Reserved Names and Constants	6
4	Units	7
5	Error and Status Messages	8
6	System Information	12
7	System Flags	12
8	Object Types	17
9	Character Keys	18
10	Command Reference	20

Version 2.0



1 Quick Reference Chart

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

Name	Description	Access	Inputs	Outputs
	Where command: substitutes values for names in an expression.*		'symbol', { name ₁ , 'symbol', name ₂ , 'symbol', ... }	'symbol' _{new} *
<	Tests whether one object is less than another object.*		x y → 0/1	x y → 0/1
>	Tests whether one object is greater than another object.*		x y → 0/1	x y → 0/1
≥	Tests whether one object is greater than or equal to another object.*		x y → 0/1	x y → 0/1
≤	Tests whether one object is less than or equal to another object.*		x y → 0/1	x y → 0/1
=	Returns an equation formed from the two arguments.*		$z_1, z_2 \rightarrow z_1 = z_2$	
==	Tests if two objects are equal.*		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.*		$z \rightarrow \sqrt{z}$	
∂	Gives the derivative of an expression, number, or unit object with respect to a specified variable of differentiation.*		'symbol', 'name' → 'symbol' _i	
↑	Creates local variables in a program.		obj ₁ ... obj _n →	
π	Returns the symbolic constant 'π' or its numerical representation, 3.14159265359.*			→ 'π'
Σ	Calculates the value of a finite series.*		'indx' x _{init} x _{final} smnd →	x _{sum}

* = function

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

* = function

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 $\boxed{\text{ENTER}}$ 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 $\boxed{\text{ENTER}}$ 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 $\boxed{\text{EVAL}}$.
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 $\boxed{\text{ENTER}}$) 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 $\boxed{\text{ON}}$.
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.

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

* = function

Name	Description	Access	Inputs	Outputs
WSLOG	Returns four strings recording the date, time, and cause of the four most recent warmstart events.			\rightarrow "log ₄ " ... "log ₁ "
ΣX	Sums the values in the independent-variable column of the current statistical matrix (reserved variable ΣDAT).			\rightarrow x_{sum}
ΣX2	Sums the squares of the values in the independent-variable column of the current statistical matrix.			\rightarrow x_{sum}
XCOL	Specifies the independent-variable column of the current statistics matrix (reserved variable ΣDAT).			$\eta_{col} \rightarrow$
XGET	Retrieves a file by XMODEM from another calculator.			'name' \rightarrow
XMIT	Sends a string serially without using Kermit and then indicates whether the transmission was successful.			"string" \rightarrow 1
XNUM	Converts an object or a list of objects to approximate numeric format.			$obj_1 \rightarrow obj_2$
XOR	Returns the logical exclusive OR of two arguments.*		$\#n_1, \#n_2 \rightarrow$	$\#n_3$
XPON	Returns the exponent of the argument.*		$x \rightarrow$	η_{egon}
XPUT	Sends a file by XMODEM to another calculator.			'name' \rightarrow
XQ	Converts a number, or a list of numbers in decimal format, to rational format.			$z_1 \rightarrow z_2$
XRECV	Prepares the HP 49 to receive an object via XModem.*			'name' \rightarrow


* = function












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 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 .
+/-	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 , , or 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.

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
ODETYPE	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.

Name	Description	Access	Inputs	Outputs
VAR	Calculates the sample variance of the coordinate values in each of the <i>m</i> columns in ΣDAT.			→ <i>X</i> _{variance}
VARS	Returns a list of the names of all variables in the VAR menu for the current directory.	 MEMORY DIRECTORY		→ { <i>global</i> ₁ , ... <i>global</i> _n }
VER	Returns the Computer Algebra System version number, and date of release.			→ "string"
VERSION	Displays the software version and copyright message.			→ "version number" "copyright message"
VISIT	Places the contents of a variable on the command line.		'name' →	
VISITB	Opens the contents of a variable in the most suitable editing environment for the particular type of object.		'name' →	
VTYPE	Returns the type number of the object in the variable.	 TYPE	'name' → <i>n</i> _{type}	
V→	Separates a vector or complex number into its component elements.	 VECTOR	[<i>x y</i>] → <i>x y</i>	
WAIT	Suspends program execution for a specified time, or until a key is pressed.	 IN	<i>x</i> →	
WHILE	Starts a WHILE ... REPEAT ... END indefinite loop structure.	 BRANCH		
WIREFRAME	Sets the plot type to WIREFRAME.			

* = function

Name	Description	Access	Inputs	Outputs
UNTIL	Starts the test clause in a DO ... UNTIL ... END indefinite loop structure.	BRANCH		
UPDIR	Makes the parent of the current directory the new current directory.			
UTPC	Returns the probability that a chi-square random variable is greater than x given n degrees of freedom.	PROBABILITY	n $x \rightarrow upc(n,x)$	
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.	PROBABILITY	n_1 n_2 $x \rightarrow upf(n_1,n_2,x)$	
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.	PROBABILITY	m v $x \rightarrow upn(m,v,x)$	
UTPT	Returns the probability that a Student's t random variable is greater than x , where n is the degrees of freedom.	PROBABILITY	n $x \rightarrow upt(n,x)$	
UVAL	Returns the numerical part of a unit object.*	TOOLS	$x_unit \rightarrow x$	
$\rightarrow V2$	Converts two numbers into a vector or complex number.	VECTOR	x_1 x_2 $x_3 \rightarrow [x_1\ x_2\ x_3]$	
$\rightarrow V3$	Converts three numbers into a vector.	VECTOR		
VANDERMONDE	Builds a Vandermonde matrix from a list of objects.	CREATE	$\{ list \} \rightarrow [[matrix]]$	

* = function

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					
Mpc	CM	MM	yd	ft	in
nmi	pc	lyr	au	km	Mi
Mil	miUS	chain	rd	fath	ftUS
	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	ftxlb
therm	MeV	eV			
Power					
W	hp				
Pressure					
Pa	atm	bar	psi	torr	mmHg
inHg	inH2O				
Temperature					
°C	°F	K	°R		
Electric Current					
V	A	C	Ω	F	W
Fdy	H	mho	S	T	Wb
Angle					
°	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 MESSAGES	
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

Name	Description	Access	Inputs	Outputs
TVMBEG	Specifies that TVM calculations treat payments as being made at the beginning of the compounding periods.			
TVMEND	Specifies that TVM calculations treat payments as being made at the end of the compounding periods.			
TVMROOT	Solves for the specified TVM variable using values from the remaining TVM variables.		'TVM variable' → $x_{TVM\ variable}$	
TYPE	Returns the type number of an object.	TEST	$obj \rightarrow n_{type}$	
UBASE	Converts a unit object to SI base units.*	UNITS TOOLS	$x_unit \rightarrow y_base-units$	
UFACT	Factors the level 1 unit from the unit expression of the level 2 unit object.	UNITS TOOLS	$x_1_unit_1\ x_2_unit_2 \rightarrow x_3_unit_2 * unit_3$	
UFL1→MINIF	Converts a UFL1 (universal font library) fontset to a HP 49G minifont.		$obj_{fontset}\ n_{ID} \rightarrow$	
→UNIT	Creates a unit object from a real number and a unit expression.		$x\ y_unit \rightarrow x_unit$	
UNPICK	Replaces the object at level <i>n</i> +2 with the object at level 2 and deletes the objects at levels 1 and 2.*	STACK	$obj_{n+2} \dots obj_k\ obj_j\ n \rightarrow obj_2 \dots obj_k\ obj_j$	
UNROT	Changes the order of the first three objects on the stack.*	STACK	$obj_j\ obj_k\ obj_l \rightarrow obj_l\ obj_j\ obj_k$	

* = function

Name	Description	Access	Inputs	Outputs
TRIG	Converts complex logarithmic and exponential terms into their equivalent trigonometric terms.	$\left[\begin{smallmatrix} \text{F6} \\ \text{F7} \end{smallmatrix} \right]$		'symb ₁ ' → 'symb ₂ '
TRIGCOS	Simplifies a trigonometric expression into cosine terms.	$\left[\begin{smallmatrix} \text{F6} \\ \text{F7} \end{smallmatrix} \right]$		'symb ₁ ' → 'symb ₂ '
TRIGO	Displays a menu of trigonometry commands.	$\left[\begin{smallmatrix} \text{GAT} \end{smallmatrix} \right]$		
TRIGSIN	Simplifies a trigonometric expression into sine terms.	$\left[\begin{smallmatrix} \text{F6} \\ \text{F7} \end{smallmatrix} \right]$		'symb ₁ ' → 'symb ₂ '
TRIGTAN	Replaces sin() and cos() terms with tan() terms.	$\left[\begin{smallmatrix} \text{F6} \\ \text{F7} \end{smallmatrix} \right]$		'symb ₁ ' → 'symb ₂ '
TRN	Returns the conjugate transpose of a matrix.	$\left[\begin{smallmatrix} \text{F6} \\ \text{F7} \end{smallmatrix} \right]$ MATRIX MAKE		[[matrix]] → [[matrix]] _{transpose}
TRNC	Truncates an object to a set number of decimal places or significant digits, or to fit the current display format.*	$\left[\begin{smallmatrix} \text{F6} \\ \text{F7} \end{smallmatrix} \right]$ REAL	z_1 $n_{truncate}$ → z_2	
TRUNC	Truncates a series expansion.	$\left[\begin{smallmatrix} \text{GAT} \end{smallmatrix} \right]$	'symb ₁ ' 'symb ₂ ' → 'symb ₃ '	
TRUTH	Sets the plot type to TRUTH.	$\left[\begin{smallmatrix} \text{GAT} \end{smallmatrix} \right]$		
TSIMP	Simplifies exponential and logarithmic expressions.	$\left[\begin{smallmatrix} \text{G6} \\ \text{F6} \\ \text{F7} \end{smallmatrix} \right]$	'symb ₁ ' → 'symb ₂ '	
TSTR	Returns a string derived from the date and time.	$\left[\begin{smallmatrix} \text{F6} \\ \text{F7} \end{smallmatrix} \right]$ TOOLS	date time → "DOW DATE TIME"	
TVARS	Lists all global variables in the current directory that contain objects of a specified type.	$\left[\begin{smallmatrix} \text{F6} \\ \text{F7} \end{smallmatrix} \right]$ MEMORY DIRECTORY	n_{type} → { global ... }	
TVM	Displays the TVM Solver menu.	$\left[\begin{smallmatrix} \text{GAT} \end{smallmatrix} \right]$		

* = function

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 PTYYPE
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

Name	Description	Access	Inputs	Outputs
THEN	Starts the true-clause in a conditional or error-trapping structure.	PIC BRANCH		
TICKS	Returns the system time as a binary integer.	TIME TOOLS TICKS	$\rightarrow \#t_{line}$	
TIME	Returns the system time in HH.MMSSs format.	TIME TOOLS	$\rightarrow time$	
\rightarrow TIME	Sets the system time.	TIME TOOLS	$time \rightarrow$	
TINC	Calculates a temperature increment.*	CAI	$x_{initial}, y_{delta} \rightarrow x_{final}$	
TLIN	Linearizes and simplifies a trigonometric expression.	TRG	$'symbol_1' \rightarrow 'symbol_2'$	
TLINE	For each pixel along the line in <i>PICT</i> defined by the specified coordinates, <i>TLINE</i> turns on/off every pixel that is on/off.	PIC PICT	$(x_1, y_1) (x_2, y_2) \rightarrow$	
TMENU	Displays a built-in menu, library menu, or user-defined menu.	CAI	$x_{menu} \rightarrow$	
TOT	Computes the sum of each of the <i>m</i> columns of coordinate values in ΣDAT .	CAI	$\rightarrow x_{sum}$	
TRACE	Returns the trace of a square matrix.	MAT ES OPERATIONS	$[[matrix]]_{n \times n} \rightarrow x_{trace}$	
TRAN	Returns the transpose of a matrix.	MAT ES OPERATIONS	$[[matrix]] \rightarrow [[matrix]]_{transpose}$	
TRANSIO	Specifies a character translation option in data transfer.	CAI	$n_{option} \rightarrow$	

* = function

Name	Description	Access	Inputs	Outputs
TAN	Returns the tangent of the argument.*	TAN		$z \rightarrow \tan z$
TAN2SC	Replaces $\tan(x)$ terms with $\sin(x)$ and $\cos(x)$ terms.	TRG		'symb ₁ ' \rightarrow 'symb ₂ '
TAN2SC2	Replaces $\tan(x)$ terms with $\sin(x)$ and $\cos(x)$ terms.	TRG		'symb ₁ ' \rightarrow 'symb ₂ '
TANH	Returns the hyperbolic tangent of the argument.*	TRG		$z \rightarrow \tanh z$
TAYLOR0	Performs a fourth-order Taylor expansion of an expression at $x = 0$.*	TRG		'symb ₁ ' \rightarrow 'symb ₂ '
TAYLR	Calculates the <i>n</i> th order Taylor polynomial of <i>symb</i> in the variable <i>global</i> .	TRG	'symb' 'global' $n_{order} \rightarrow$ 'symb ₁ '	
TCHEBYCHEFF	Returns the <i>n</i> th Tchebycheff polynomial.*	TRG	$n_1 \rightarrow$ 'symb ₁ '	
TCOLLECT	Linearizes products in a trigonometric expression by collecting and combining sine and cosine terms.	TRG	'symb ₁ ' \rightarrow 'symb ₂ '	
TDELTA	Calculates a temperature change.*	TRG	$x \ y \rightarrow x_{delta}$	
TEVAL	For the specified operation, performs the same function as EVAL, and returns the time taken to perform the evaluation.	TRG	$obj_1 \rightarrow obj_2 \ hms$	
TEXPAND	Expands transcendental functions.	TRG	'symb ₁ ' \rightarrow 'symb ₂ '	
TEXT	Displays the stack.	TRG		

* = function

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 \blacktriangle	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.

1 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.

Name	Description	Access	Inputs	Outputs
SUBTMOD	Performs a subtraction, modulo the current modulus.*	MOD MODULO		
SVD	Returns the singular value decomposition of an $m \times n$ matrix.	SVD FACTORIZATION	$obj_1 \text{ } obj_2 \rightarrow [[matrix]]_A \rightarrow [[matrix]]_U [[matrix]]_V [vector]_S$	
SVL	Returns the singular values of an $m \times n$ matrix.	SVD FACTORIZATION	$[[matrix]] \rightarrow [vector]$	
SWAP	Swaps the position of the two objects.	SWAP STACK	$obj_1 \text{ } obj_2 \rightarrow obj_2 \text{ } obj_1$	
SYLVESTER	For a symmetric matrix A, returns D and P where D is a diagonal matrix and $A = P \cdot D \cdot P$.	SYLVESTER	$[[matrix]]_A \rightarrow [[matrix]]_D [[matrix]]_P$	
SYSEVAL	Evaluates unnamed operating system objects specified by their memory addresses.	SYSEVAL	$\#n_{\text{address}} \rightarrow$	
%T	Returns the percent of the first argument that is represented by the second argument.*	PCT REAL	$x \text{ } y \rightarrow 100y/x$	
TABVAL	For an expression and a list of values, returns the results of substituting the values for the default variable in the expression.	TABVAL	$'symbol', \{ list_1 \} \rightarrow 'symbol', \{ list_2 \}$	
TABVAR	For a rational function, computes the turning points and where the function increases or decreases.	TABVAR	$'symbol', \{ list_1 \} \rightarrow 'symbol', \{ list_2 \} \text{ } grab_1$	
→TAG	Combines objects to create a tagged object.	TAG	$obj \text{ } 'tag' \rightarrow :tag: obj$	
TAIL	Returns all but the first element of a list or string.	TAIL CHARS	$\{ obj_1, \dots, obj_n \} \rightarrow \{ obj_2, \dots, obj_n \}$	

* = function

Name	Description	Access	Inputs	Outputs
STO-	Calculates the difference between an object and a variable and stores the object in the variable.	MEMORY ARITHMETIC	<i>obj</i> 'name' →	
STO*	Multiplies the contents of a specified variable by a number or other object.	MEMORY ARITHMETIC	<i>obj</i> 'name' →	
STO/	Calculates the quotient of a number and the contents of a specified variable. Stores new value in the specified variable.	MEMORY ARITHMETIC	<i>obj</i> 'name' →	
STO+	Adds a number or other object to a variable.	MEMORY ARITHMETIC	<i>obj</i> 'name' →	
STO:	Stores <i>obj</i> in the reserved variable ΣDAT.	MEMORY	<i>obj</i> →	
→STR	Converts any object to string form.	MEMORY	<i>obj</i> → "string"	
STR→	Evaluates the text of a string as if the text were entered from the command line.	MEMORY	<i>obj</i> ₁ → <i>obj</i> ₂	
STREAM	Repeatedly executes <i>obj</i> on the first two elements in a list until the list is exhausted. Returns the final result.	MEMORY LIST PROCEDURES	{ <i>list</i> } <i>obj</i> → <i>result</i>	
STWS	Sets the current binary integer wordsize to <i>n</i> bits, where <i>n</i> is a value from 1 through 64 (the default is 64).	MEMORY BASE	<i>n</i> →	
SUB	Returns the specified portion of an object.	MEMORY LIST	"string," <i>n</i> _{start position} <i>n</i> _{end position} →	"string"
SUBST	Substitutes a value or expression for a variable in an expression.*	MEMORY	' <i>symbol</i> ,' <i>Z</i> ₁ → ' <i>symbol</i> ₂ '	

* = function

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 \square (USER) once for user mode lock.
Clear.*	Press \square (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 GET1 or PUT1 wrapped index (to 1).
Clear.*	The last GET1 or PUT1 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.

Name	Description	Access	Inputs	Outputs
SRB	Shifts a binary integer one byte to the right.	\square (01H) BASE BYTE	$\#n_1 \rightarrow \#n_2$	
SRECV	Returns up to n characters from the serial input buffer, with an error digit if an error occurred.	\square (CAT)	$n \rightarrow \text{'string'}$ 0/1	
SREPL	Finds and replaces a string in a text object.	\square (CAT)	"string," "string ₂ " "string ₃ " \rightarrow "string ₄ "	
START	Begins START ... NEXT and START ... STEP definite loop structures.	\square (PRG) BRANCH	START x_{start} $x_{finish} \rightarrow$	
STD	Sets the number display format to standard mode.	\square (CAT)		
STEP	Defines the increment (step) value, and ends definite loop structure.	\square (PRG) BRANCH		
SEQ	Stores an object into the reserved variable EQ in the current directory.	\square (CAT)	$obj \rightarrow$	
STIME	Specifies the period that SRECV (serial reception) and XMIT (serial transmission) wait before timing out.*	\square (CAT)	$x_{seconds} \rightarrow$	
STO	Stores an object into a specified variable or object.	\square (STO)	obj 'name' \rightarrow	
STOALARM	Stores an alarm in the system alarm list and returns its alarm index number.	\square (IME) TOOLS ALRM	$x_{line} \rightarrow n_{index}$	
STOF	Sets the states of the system flags or the system and user flags.	\square (CAT)	$\#n_{system} \rightarrow$	
STOKEYS	Assigns objects to specified keys on the user keyboard.	\square (CAT)	$\{obj_1, x_{key1}, \dots, obj_n, x_{keyn}\} \rightarrow$	

* = function

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
















* = function

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:	\rightarrow GROB can accept multi-line strings.
Clear:*	\rightarrow 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.

Name	Description	Access	Inputs	Outputs
SIGMA	Calculates the discrete antiderivative of a function with respect to a specified variable.*			'symb ₁ ' z ₁ → 'symb ₂ '
SIGMAVX	Calculates the discrete antiderivative of a function with respect to the current variable.*			'symb ₁ ' → 'symb ₂ '
SIGN	Returns the sign of a real number.*		REAL	z ₁ → z ₂
SIGNTAB	Returns the sign table of a rational function of one variable.			'symb ₁ ' → { /list _i }
SIMP2	Simplifies two objects by dividing them by their greatest common divisor.			'symb ₁ ' 'symb ₂ ' → 'symb ₃ ' 'symb ₄ '
SIN	Returns the sine of the argument.*			z → sin z
SINCOS	Converts complex logarithmic and exponential expressions to expressions with trigonometric terms.			'symb ₁ ' → 'symb ₂ '
SINH	Returns the hyperbolic sine of the argument.*			z → sinh z
SINV	Replaces the contents of a variable with its inverse.*		HYPERBOLIC	'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.		MEMORY ARITHMETIC	"string" → n
SL	Shift a binary integer one bit to the left.*		BASE BIT	#n ₁ → #n ₂

* = function

Name	Description	Access	Inputs	Outputs
SCONJ	Conjugates the contents of a named object.	  MEMORY ARITHMETIC	'name' →	
SCROLL	Displays the contents of a named object.		'name' →	
SDEV	Calculates the sample standard deviation of each of the <i>m</i> columns of coordinate values in <i>ZDATA</i> .		→ <i>X_{dev}</i>	
SEND	Sends a copy of the named objects to a Kermit device.		'name' →	
SEQ	Returns a list of results generated by repeatedly executing <i>obj_{exec}</i> using <i>/index</i> between <i>X_{start}</i> to <i>X_{end}</i> , in steps of <i>X_{incr}</i> .	  LIST PROCEDURES	<i>obj_{exec}</i> <i>index</i> <i>X_{start}</i> <i>X_{end}</i> <i>X_{incr}</i> → { <i>/list</i> }	
SERIES	For a given function, computes Taylor series; asymptotic development and limit at finite and infinite points.	  LIMITS & SERIES	' <i>symb₁</i> ' ' <i>symb₂</i> ' <i>Z₁</i> → { <i>/list</i> } ' <i>symb₃</i> '	
SERVER	Starts Kermit Server mode.			
SEVAL	Evaluates the variables in an expression and substitutes the values into the expression.*		' <i>symb₁</i> ' → ' <i>symb₂</i> '	
SF	Sets a specified user or system flag.	  TEST	<i>n_{flag number}</i> →	
SHOW	Returns ' <i>symb₂</i> ', which is equivalent to ' <i>symb₁</i> ', but with all implicit references to the variable <i>name</i> made explicit.		' <i>symb₁</i> ' ' <i>name</i> ' → ' <i>symb₂</i> '	
SIDENS	Calculates the intrinsic density of silicon as a function of temperature, <i>x_T</i> .		<i>x_T</i> → <i>X_{density}</i>	

* = function


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 x 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

3 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	TAN
(sp)	32	SPC	V	86	CEX
!	33	2	W	87	
"	34	X	X	88	X
#	35	3	Y	89	
\$	36	4	Z	90	
%	37	1		91	
&	38	ENTER	\	92	5
'	39	EQW]	93	
(40		^	94	
)	41		-	95	
*	42	X	.	96	EQW
+	43		a	97	F1
,	44	SPC	b	98	F2
-	45		c	99	F3
.	46		d	100	F4
/	47		e	101	F5
0	48	0	f	102	F6
1	49	1	g	103	APPS
2	50	2	h	104	MODE
3	51	3	i	105	TOOL
4	52	4	j	106	VAR
5	53	5	k	107	STD
6	54	6	l	108	NXT
7	55	7	m	109	HIST
8	56	8	n	110	CAT
9	57	9	o	111	EQW
:	58		p	112	SYMB
;	59	2	q	113	
<	60	X	r	114	
=	61		s	115	SIN
>	62		t	116	COS
?	63	3	u	117	TAN
@	64	ENTER	v	118	CEX
A	65	F1	w	119	
B	66	F2	x	120	X
C	67	F3	y	121	
D	68	F4	z	122	
E	69	F5	{	123	
F	70	F6		124	TOOL
G	71	APPS	}	125	
H	72	MODE	~	126	1
I	73	TOOL		127	CHARS
J	74	VAR	Δ	128	6
K	75	STD	Σ	129	CHARS
L	76	NXT	√	130	CHARS
M	77	HIST	1/x	131	
N	78	CAT	∫	132	TAN
O	79	EQW	Σ	133	SIN
P	80	SYMB	▶	134	STD
Q	81		π	135	SPC
R	82		δ	136	COS
S	83	SIN	Δ	137	X
T	84	COS	>	138	

Name	Description	Access	Inputs	Outputs
R→I	Converts a real number to an integer.*	CAT		$z_1 \rightarrow r_1$
SAME	Compares two objects, and returns a true result (1) if they are identical, and a false result (0) if they are not.	TEST		$obj_1 \rightarrow 0/1$
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		$x_{scale} \rightarrow y_{scale} \rightarrow$
SCALEH	Multiplies the vertical plot scale by x_{factor} .	CAT		$x_{factor} \rightarrow$
SCALEW	Multiplies the horizontal plot scale by y_{factor} .	CAT		$y_{factor} \rightarrow$
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.	QUANTILES FACTORIZATION		$[[matrix]]_A \rightarrow [[matrix]]_L [[matrix]]_T$
SCI	Sets the number display to scientific mode; one digit left of the fraction mark and n significant digits to the right.	CAT		$n \rightarrow$
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
rref	Reduces a matrix to row-reduced echelon form and returns pivot points.*		$[[\text{matrix}_1]] \rightarrow \{ \text{list} \}$ $[[\text{matrix}_2]]$	
RREF	Reduces a matrix to row-reduced echelon form.*	LINEAR SYSTEMS	$[[\text{matrix}_1]] \rightarrow [[\text{matrix}_2]]$	
RREFMOD	Performs modular row-reduction to echelon form on a matrix, modulo the current modulus.		$[[\text{matrix}_1]] \rightarrow [[\text{matrix}_2]]$	
RRK	Computes the solution to an initial value problem for a differential equation with known partial derivatives.		$\{ \text{list} \} \text{ } x_{\text{sol}} \text{ } x_{\text{t final}} \rightarrow \{ \text{list} \} \text{ } x_{\text{sol}}$	
RRKSTEP	Computes the next solution step to an initial value problem for a differential equation, and displays method used.		$\{ \text{list} \} \text{ } x_{\text{sol}} \text{ } h \text{ fast} \rightarrow \{ \text{list} \} \text{ } x_{\text{sol}} \text{ } h_{\text{next}} \text{ current}$	
RSBERR	Returns an error estimate for a given step h when solving an initial values problem for a differential equation.		$\{ \text{list} \} \text{ } h \rightarrow \{ \text{list} \} \text{ } h \text{ } y_{\text{dels}} \text{ error}$	
RSD	Computes the residual B – AZ of the arrays B, A, and Z.	OPERATIONS	$[[\text{matrix}]]_B \text{ } [[\text{matrix}]]_A \text{ } [[\text{matrix}]]_Z \rightarrow [[\text{matrix}]]_{B-AZ}$	
RSWP	Swaps rows i and j of a matrix and returns the modified matrix.	CREATE ROW	$[[\text{matrix}]]_i \text{ } n_{\text{row}} \text{ } n_{\text{row}} \rightarrow [[\text{matrix}]]_j$	
R→B	Converts a positive real to its binary integer equivalent.		$n \rightarrow \#n$	
R→C	Combines two real numbers or real arrays into a single complex number or complex array.	TYPE	$x \text{ } y \rightarrow (x,y)$	
R→D	Converts a real number expressed in radians to its equivalent in degrees.*	REAL	$x \rightarrow (180/\pi)x$	

* = function

Char.	No.	Key(s)	Char.	No.	Key(s)
≠	139		Æ	198	
α	140		Ç	199	
→	141		È	200	
↱	142		É	201	
↓	143		Ê	202	
↑	144		Ë	203	
γ	145		ì	204	
δ	146		í	205	
ε	147		Î	206	
η	148		Ï	207	
θ	149		Ð	208	
λ	150		Ñ	209	
ρ	151		Ò	210	
σ	152		Ó	211	
τ	153		Ô	212	
ω	154		Õ	213	
Δ	155		Ö	214	
Π	156		×	215	
Ω	157		Ø	216	
•	158		Ù	217	
∞	159		Ú	218	
∞	160		Û	219	
i	161		Ü	220	
ε	162		Ý	221	
£	163		ß	222	
¤	164		ß	223	
¥	165		à	224	
!	166		á	225	
§	167		â	226	
•	168		ã	227	
•	169		ä	228	
•	170		å	229	
«	171		æ	230	
»	172		ç	231	
•	173		è	232	
•	174		é	233	
•	175		ê	234	
•	176		ë	235	
±	177		ì	236	
±	178		í	237	
±	179		î	238	
•	180		ï	239	
μ	181		ð	240	
•	182		ñ	241	
•	183		ñ	242	
•	184		ó	243	
•	185		ô	244	
•	186		ö	245	
»	187		ø	246	
¼	188		÷	247	
½	189		ø	248	
¾	190		ù	249	
¿	191		ú	250	
À	192		û	251	
Á	193		ü	252	
Â	194		ý	253	
Ã	195		ÿ	254	
Ä	196		ÿ	255	
Å	197				

19 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 see 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
'symp'	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
ROLLD	Moves the contents of level 2 to a specified level, n , and rolls downward the portion of the stack beneath the specified level.	PFR STACK	$obj_n \dots obj_1\ n \rightarrow$	$obj_1\ obj_2 \dots$
ROMULOAD	Transfers the operating system to another calculator.	CAL		
ROOT	Returns the value of the specified variable <i>global</i> for which the specified program or algebraic object most nearly evaluates to zero or a local extremum.	CAL	'program' <i>global</i> <i>guess</i> $\rightarrow x_{root}$	
ROT	Rotates the first three objects on the stack, moving the object on level 3 to level 1.	PFR STACK	$obj_3\ obj_2\ obj_1 \rightarrow$	$obj_2\ obj_1\ obj_3$
→ROW	Transforms a matrix into a series of row vectors and returns the vectors and a row count.	VAR STACK CREATE ROW	[[matrix]]	\rightarrow [vector] _{row1} ... [vector] _{rown} 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).	VAR STACK DELETE ROW	[[matrix]] ₁ $n_{row} \rightarrow$	[[matrix]] ₂ [vector] _{low}
ROW+	Inserts an array into a matrix at the position indicated by n_{index} , and returns the modified matrix.	VAR STACK CREATE ROW	[[matrix]] ₁ [[matrix]] ₂ $n_{index} \rightarrow$	[[matrix]] ₂
ROW→	Transforms a series of row vectors and a row count into a matrix containing those rows.	VAR STACK CREATE ROW	[vector] _{row1} ... [vector] _{rown} $n \rightarrow$	[[matrix]]
RR	Rotates a binary integer one bit to the right.	BASE BIT	$\#n_1 \rightarrow$	$\#n_2$
RRB	Rotates a binary integer one byte to the right.	BASE BYTE	$\#n_1 \rightarrow$	$\#n_2$

* = function

Name	Description	Access	Inputs	Outputs
REVLIST	Reverses the order of the elements in a list.	LIST PROCEDURES	$\{ obj_h, \dots obj_i \} \rightarrow \{ obj_h, \dots obj_i \}$	
RISCH	Performs symbolic integration on a function using the Risch algorithm.*	DERIV. & INTEG	'symb ₁ ' $z_1 \rightarrow$ 'symb ₂ '	
RKF	Computes solution to an initial value problem for a differential equation using the Runge–Kutta–Fehlberg (4,5) method.		$\{ list \} \ X_{old} \ X_{Tfinal} \rightarrow \{ list \} \ X_{old}$	
RKFERR	Returns the absolute error estimate for a given step h when solving an initial value problem for a differential equation.		$\{ list \} \ h \rightarrow \{ list \} \ h \ y_{data} \ error$	
RKFSTEP	Computes the next solution step (h_{next}) to an initial value problem for a differential equation.		$\{ list \} \ X_{old} \ h \rightarrow \{ list \} \ X_{old} \ h_{next}$	
RL	Rotates a binary integer one bit to the left.	BASE BIT	$\#n_1 \rightarrow \#n_2$	
RLB	Rotates a binary integer one byte to the left.	BASE BYTE	$\#n_1 \rightarrow \#n_2$	
RND	Rounds an object to a specified number of decimal places or significant digits, or to fit the current display format.*	REAL	$z_1 \ n_{round} \rightarrow z_2$	
RNRM	Returns the row norm (infinity norm) of an array.	OPERATIONS	$[array] \rightarrow X_{low \ norm}$	
ROLL	Moves the contents of a specified level to level 1, and rolls up the portion of the stack beneath the specified level.	STACK	$obj_h \dots obj_i \ n \rightarrow obj_{h-1} \dots obj_i \ obj_h$	

* = function

Name	Description	Access	Inputs	Outputs
ABCVU	Returns a solution in polynomials u and v of $au+bv=c$ where a and b are polynomials, and c is a value.	POLYNOMIAL	'symb ₁ ' 'symb ₂ ' $z \rightarrow$ 'symb ₃ ' 'symb ₄ '	
ABS	Returns the absolute value of its argument.*		$x \rightarrow x $	
ACK	Acknowledges the oldest past-due alarm.	TOOLS ALRM		
ACKALL	Acknowledges all past-due alarms.	TOOLS ALRM		
ACOS	Returns the value of the angle with the given cosine.		$z \rightarrow \arccos z$	
ACOS2S	Replaces cos() terms with equivalent asin() terms.*		'symb ₁ ' \rightarrow 'symb ₂ '	
ACOSH	Returns the inverse hyperbolic cosine of the argument.*	HYPERBOLIC	$z \rightarrow \cosh z$	
ADD	Adds corresponding elements of two lists or adds a number to each of the elements of a list.		$\{ list_1 \} \{ list_2 \} \rightarrow \{ list_{result} \}$	
ADDTMOD	Adds two expressions or values, modulo the current modulus.*		'symb ₁ ' 'symb ₂ ' \rightarrow 'symb ₃ '	
ADDTOREAL	Adds the specified global name to the reserved variable REALASSUME.*	MODULO	'global' \rightarrow	

* = function

Name	Description	Access	Inputs	Outputs
ALOG	Returns the common antilogarithm; that is, 10 raised to the given power.*	\square (U) ³		$z \rightarrow 10^z$
AMORT	Amortizes a loan or investment based upon the current amortization settings.	\square (FIVE)		$n \rightarrow$ principal interest balance
AND	Returns the logical AND of two arguments.*	\square (BASE) LOGIC	$\#n_1 \ \#n_2 \rightarrow \#n_3$	
ANIMATE	Displays graphic objects in sequence.	\square (PRE) GROB	$grob_1 \dots grob_n \ \eta_{grabs} \rightarrow$ same stack	
ANS	Recalls the <i>n</i> th answer from history.	\square (ANS)	$n \rightarrow obj_n$	
APPLY	Creates an expression from the specified function name and arguments.*	\square (AT)	$\{ symb_1, \dots symb_n \}$ 'name' \rightarrow 'name (symb ₁ ... symb _n)'	
ARC	Draws an arc in PICT counterclockwise.	\square (PRE) PICT	$(x, y) \ X_{radius} \ X_{q1} \ X_{q2} \rightarrow$	
ARCHIVE	Creates a backup copy of the HOME directory.	\square (PRE) MEMORY	η_{port} 'name' \rightarrow	
ARG	Returns the (real) polar angle of a complex number.*	\square (PRE)	$(x, y) \rightarrow \theta$	
ARIT	Displays a menu of arithmetic commands.	\square (AT)		
→ARRY	Returns a vector of <i>n</i> real or complex elements or a matrix of <i>n</i> x <i>m</i> real or complex elements.	\square (AT)	$z_1 \dots z_n \ \eta_{element} \rightarrow$ [vector]	
ARRY →	Takes an array and returns its elements as separate real or complex numbers.	\square (AT)	[vector] $\rightarrow z_1 \dots z_n \ \{ \eta_{element} \}$	
ASIN	Returns the value of the angle with the given sine.*	\square (ASIN)	$z \rightarrow asin \ z$	

* = function

Name	Description	Access	Inputs	Outputs
RECT	Sets the coordinate mode to rectangular.	\square (AT)		
RECV	Instructs the HP 49 to look for a named file on another Kermit server device.	\square (AT)		
REF	Reduces a matrix to echelon form.	\square (ADDRESS) LINEAR SYSTEMS	[[matrix ₁]] \rightarrow [[matrix ₂]]	
REMAINDER	Returns the remainder of the Euclidean division of two polynomials.*	\square (ARITH) POLYNOMIAL	'symb ₁ ' 'symb ₂ ' \rightarrow 'symb ₃ '	
RENAME	Renames a variable as specified.	\square (AT)	'name _{new} ' 'name _{old} ' \rightarrow	
REORDER	Given a polynomial and variable, reorders the variables in the order of powers set in the CAS modes.*	\square (AT)	'symb ₁ ' $z_1 \rightarrow$ 'symb ₂ '	
REPEAT	Starts a loop clause in a WHILE ... REPEAT ... END indefinite loop structure.	\square (PRE) BRANCH		
REPL	Replaces a portion of the target object with a specified object, beginning at a specified position.	\square (PRE) LIST	[[matrix]], $\eta_{position}$ [[matrix]] ₂ \rightarrow [[matrix]] ₃	
RES	Specifies the resolution of mathematical and statistical plots.	\square (AT)	$\eta_{interval} \rightarrow$	
RESTORE	Replaces the current HOME directory with the specified backup copy previously created by ARCHIVE.	\square (AT)	η_{port} 'name _{backup} ' \rightarrow	
RESULTANT	Returns resultant of two polynomials of the current variable.*	\square (AT)	'symb ₁ ' 'symb ₂ ' $\rightarrow z_1$	

* = function

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

* = function

Name	Description	Access	Inputs	Outputs
QUOT	Returns the quotient part of the Euclidean division of two polynomials.	\square QMTH POLYNOMIAL	'symb ₁ ' 'symb ₂ ' \rightarrow 'symb ₃ '	
QUOTE	Returns unevaluated arguments. *	QAT		$obj_1 \rightarrow obj_2$
QXA	Expresses a quadratic form in matrix form.	QAT	'symb ₁ ' [vector ₁] \rightarrow 'symb ₂ ' [vector ₂]	
$\rightarrow Q\pi$	Returns a rational form of the argument, or a rational form of the argument with π factored out.	QAT	$x \rightarrow$ 'alb' π	
RAD	Sets Radians angle mode.	QAT		
RAND	Returns a pseudo-random number generated using a seed value, and updates the seed value.	\square MTH PROBABILITY	$\rightarrow x_{\text{random}}$	
RANK	Returns the rank of a rectangular matrix.	\square QARTES OPERATIONS	[[matrix]] $\rightarrow \rho_{\text{rank}}$	
RANM	Returns a matrix of specified dimensions that contains random integers in the range -9 to 9.	\square QARTES CREATE	{ m, n } \rightarrow [[random matrix]] _{mn}	
RATIO	Prefix form of / (divide). *	QAT	$z_1, z_2 \rightarrow z_1/z_2$	
RCEQ	Returns the unevaluated contents of the reserved variable EQ from the current directory.	QAT	$\rightarrow obj_{\text{eq}}$	
RCI	Multiplies row n of a matrix (or element n of a vector) by a constant x_{factor} , and returns the modified matrix.	\square QARTES CREATE ROW	[[matrix]] ₁ $x_{\text{factor}} \quad n_{\text{row number}} \rightarrow$ [[matrix]] ₂	

* = function

Name	Description	Access	Inputs	Outputs
PURGE	Purges the named variables or empty subdirectories from the current directory.	\square $\overline{\text{PRG}}$ MEMORY		'global' \rightarrow
PUT	Replaces the object at a specified position in an array.	\square $\overline{\text{PRG}}$ LIST ELEMENTS	$[[\text{matrix}]]_1, \eta_{\text{position}} \text{ } z_{\text{put}} \rightarrow [[\text{matrix}]]_2$	
PUT1	As for PUT (see above) but also increments the position.	\square $\overline{\text{PRG}}$ LIST ELEMENTS	$[[\text{matrix}]]_1, \eta_{\text{position1}} \text{ } z_{\text{put}} \rightarrow [[\text{matrix}]]_2 \text{ } \eta_{\text{position2}}$	
PVAR	Calculates the population variance of the coordinate values in each of the m columns in ΣDAT .	\square $\overline{\text{A1}}$		$\rightarrow X_{\text{variance}}$
PVARS	Returns a list of the backup objects and library objects in a specified port, and the available memory.	\square $\overline{\text{A1}}$	$\eta_{\text{port}} \rightarrow \{ \text{' } \eta_{\text{port}} \text{' name}_{\text{backup}} \dots \}$	memory
PVIEW	Displays <i>PICT</i> with the specified coordinates at the upper left corner of the graphics display.	\square $\overline{\text{PRG}}$ PICT	$(x,y) \rightarrow$	
PWRFIT	Stores <i>PWRFIT</i> in ΣPAR , so that subsequent executions of LR use the power curve fitting model.	\square $\overline{\text{A1}}$		
PX→C	Converts the specified pixel coordinates to user-unit coordinates.	\square $\overline{\text{PRG}}$ PICT	$\{ \#_n, \#_s \} \rightarrow (x,y)$	
→Q	Returns a rational form of the argument.	\square $\overline{\text{A1}}$	$x \rightarrow \text{'a1b'}$	
QR	Returns the QR factorization of an $m \times n$ matrix.	\square $\overline{\text{VALUES}}$ FACTORIZATION	$[[\text{matrix}]]_A \rightarrow [[\text{matrix}]]_Q \text{ } [[\text{matrix}]]_R$	$[[\text{matrix}]]_p$
QUAD	Finds zeros of an expression equated to 0, or solves an equation. Same as SOLVE.	\square $\overline{\text{A1}}$	'sybm1' 'global' \rightarrow 'sybm2'	

* = 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.	\square $\overline{\text{A1}}$		$\rightarrow \eta_{\text{chars}} \text{ } 0/1$
BYTES	Returns the number of bytes and the checksum for the given object.	\square $\overline{\text{PRG}}$ MEMORY	$obj \rightarrow \# \eta_{\text{checksum}} \text{ } x_{\text{size}}$	
B→R	Converts a binary integer to its floating-point equivalent.	\square $\overline{\text{BASE}}$	$\#n \rightarrow n$	
CASCFG	Restores the default CAS mode settings.	\square $\overline{\text{A1}}$		
CASE	Starts CASE ... END conditional structure.	\square $\overline{\text{PRG}}$ BRCH		
CEIL	Returns the smallest integer greater than or equal to the argument.*	\square $\overline{\text{MTH}}$ REAL	$x \rightarrow n$	
CENTR	Adjusts first two parameters in <i>PPAR</i> , (x_{min} , y_{min}) and (x_{max} , y_{max}), so that point (x , y) is plot center.	\square $\overline{\text{A1}}$	$(x,y) \rightarrow$	
CF	Clears the specified user or system flag.	\square $\overline{\text{PRG}}$ TEST	$\eta_{\text{flagnumber}} \rightarrow$	
%CH	Returns the percent change from x to y as a percentage of x .*	\square $\overline{\text{MTH}}$ REAL	$x \text{ } y \rightarrow 100(y - x)/x$	
CHINREM	Solves a system of simultaneous polynomial congruences in the ring $\mathbb{Z}[x]$.	\square $\overline{\text{PRG}}$ POLYNOMIAL	$[\text{vector}_1 \text{ } [\text{vector}_2] \rightarrow [\text{vector}_3]$	
CHOOSE	Creates a user-defined choose box.	\square $\overline{\text{PRG}}$ IN	"prompt" { $c_1 \dots c_n$ } $\eta_{\text{pos}} \rightarrow obj \text{ or result "r"}$	

* = function

Name	Description	Access	Inputs	Outputs
CHR	Returns a string representing the character corresponding to the character code n .	\square $\overline{\text{PRG}}$ TYPE	$n \rightarrow$ "string"	
CKSM	Specifies the error-detection scheme.	\square $\overline{\text{CAT}}$	$n_{\text{checksum}} \rightarrow$	
CLEAR	Removes all objects from the stack or history.	\square $\overline{\text{QEP}}$	$obj_1 \dots obj_i \rightarrow$	
CLKADJ	Adjusts the system time by x clock ticks, where 8192 clock ticks equal 1 second.	\square $\overline{\text{TIME}}$ TOOLS	$x \rightarrow$	
OLLCD	Clears (blanks) the stack display.	\square $\overline{\text{PRG}}$ OUT		
CLOSEIO	Closes the serial port, and clears the input buffer and any error messages for KERRM.	\square $\overline{\text{CAT}}$		
QLΣ	Purges the current statistics matrix.	\square $\overline{\text{CAT}}$		
CLVAR	Purges all variables and empty subdirectories in the current directory.	\square $\overline{\text{CAT}}$		
CMPLX	Displays a menu of commands pertaining to complex numbers.	\square $\overline{\text{CAT}}$		
CNRM	Returns the column norm (one-norm) of the array argument.	\square $\overline{\text{MATHS}}$ OPERATIONS	$[array] \rightarrow$ $x_{\text{columnnorm}}$	
COL−	Deletes column n of a matrix, and returns the modified matrix (or vector) and the deleted column (or element).	\square $\overline{\text{MTH}}$ MATRIX COL	$[[matrix]], n_{\text{column}} \rightarrow [[matrix]]_2$ [vector] _{column}	
COL+	Inserts an array into a matrix at the position indicated by n_{index} , and returns the modified array.	\square $\overline{\text{MTH}}$ MATRIX COL	[vector], n_{element} $n_{\text{index}} \rightarrow$ [vector] ₂	

* = function

Name	Description	Access	Inputs	Outputs
PROMPT	Displays the contents of "prompt" in the status area, and halts program execution.	\square $\overline{\text{PRG}}$ IN	"prompt" \rightarrow	
PROMPTSTO	Creates a variable with the specified name, prompts for a value, and stores the value you enter in the variable.	\square $\overline{\text{CAT}}$	"global" \rightarrow	
PROOT	Returns all roots of an n -degree polynomial having real or complex coefficients.	\square $\overline{\text{MTH}}$ POLYNOMIAL	[array] _{coefficients} \rightarrow [array] _{roots}	
PROPFRACTION	Splits an improper fraction into an integer and a fraction.	\square $\overline{\text{MTH}}$	'symp', \rightarrow 'symp' ₂	
PRST	Prints all objects on the stack, starting with the object on the highest level.	\square $\overline{\text{CAT}}$		
PRSTC	Prints in compact form all objects on the stack, starting with the object on the highest level.	\square $\overline{\text{CAT}}$	'name' \rightarrow	
PRVAR	Searches the current directory path or port for the specified variables and prints the name and contents of each variable.	\square $\overline{\text{CAT}}$		
PSDEV	Calculates the population standard deviation of each of the m columns of coordinate values in ΣDAT.	\square $\overline{\text{CAT}}$	\rightarrow x_{psdev}	
Psi	Calculates the digamma function in one point.*	\square $\overline{\text{CAT}}$	'symp', $n \rightarrow$ 'symp' ₂	
PSI	Calculates the polygamma function in one point.*	\square $\overline{\text{CAT}}$	'symp', \rightarrow 'symp' ₂	
PTAYL	Returns the Taylor polynomial for a specified polynomial.*	\square $\overline{\text{MTH}}$ POLYNOMIAL	'symp', $z_1 \rightarrow$ 'symp' ₂	

* = function

Name	Description	Access	Inputs	Outputs
POWMOD	Raises an object (number or expression) to the specified power, and expresses the result modulo the current modulus.	\square $\overline{A}(\overline{H})$ MODULO		$obj_1\ z_1 \rightarrow obj_2$
PR1	Prints an object in multiline printer format.	\square $\overline{A}(\overline{L})$		
PREDV	Returns the predicted dependent-variable value $y_{dependent}$ based on $x_{independent}$, the selected statistical model, and the current regression coefficients in ΣPAR .	\square $\overline{A}(\overline{L})$	$x_{independent} \rightarrow y_{dependent}$	
PREDX	Returns the predicted independent-variable value $x_{independent}$ based on: $y_{dependent}$, the selected statistical model, and the current regression coefficients in ΣPAR .	\square $\overline{A}(\overline{L})$	$y_{dependent} \rightarrow x_{independent}$	
PREDY	Returns the predicted dependent-variable value based on $x_{independent}$, the selected statistical model, and the current regression coefficients in ΣPAR . Same as PREDV.	\square $\overline{A}(\overline{L})$	$x_{independent} \rightarrow y_{dependent}$	
PREVAL	Relative to the current default variable, returns the difference between the values of a function at two specified values.*	\square $\overline{A}(\overline{L})$ $\overline{D}ERIV$. & $\overline{I}NTEG$	$'symbol_1'\ z_1\ z_2 \rightarrow 'symbol_2'$	
PREVPRIME	Given an integer, finds the closest prime number less than the integer.*	\square $\overline{A}(\overline{H})$ $\overline{I}NTEG$	$n_1 \rightarrow n_2$	
PRLCD	Prints a pixel-by-pixel image of the current display (excluding the annunciators).	\square $\overline{A}(\overline{L})$		

* = function

Name	Description	Access	Inputs	Outputs
$\rightarrow COL$	Transforms a matrix into a series of column vectors and returns the vectors and a column count.	\square $\overline{A}(\overline{H})$ $\overline{M}ATRX$ $\overline{C}OL$	$[[\ matrix]]$ $\rightarrow [\ vector]_{col1}$	$[\ vector]_{join}\ n_{colcount}$
$COL \rightarrow$	Transforms a series of column vectors and a column count into a matrix containing those columns.	\square $\overline{A}(\overline{H})$ $\overline{M}ATRX$ $\overline{C}OL$	$[\ vector]_{col1}\ [vector]_{join}\ n_{colcount} \rightarrow [[\ matrix]]$	
$COLCT$	Factorizes a polynomial or integer. Identical to FACTOR.	\square $\overline{A}(\overline{L})$	$'symbol_1' \rightarrow 'symbol_2'$	
$COL\S$	Specifies the independent-variable and dependent-variable columns of the current statistics matrix.	\square $\overline{A}(\overline{L})$	$x_{col}\ y_{col} \rightarrow$	
COMB	Returns the number of possible combinations of n items taken m at a time.*	\square $\overline{A}(\overline{H})$ $\overline{P}ROBABILITY$	$n\ m \rightarrow C_{n,m}$	
CON	Returns a constant array, defined as an array whose elements all have the same value.	\square $\overline{A}(\overline{H})$ $\overline{M}ATRX$ $\overline{M}AKE$	$\{ n_{columns} \}\ z_{constant} \rightarrow [vector_{constant}]$	
COND	Returns the 1-norm (column norm) condition number of a square matrix.	\square $\overline{A}(\overline{H})$ $\overline{M}ATRX$ $\overline{N}ORMALIZE$	$[[\ matrix]]_{new} \rightarrow x_{condition\ number}$	
CONIC	Sets the plot type to CONIC.	\square $\overline{A}(\overline{L})$		
CONJ	Conjugates a complex number or a complex array.*	\square $\overline{A}(\overline{M}P\overline{X})$	$x \rightarrow x$	
CONLIB	Opens the Constants Library catalog.	\square $\overline{A}(\overline{F})$ $\overline{C}ONSTANTS$ $\overline{L}IB$		
CONST	Returns the value of a constant.*	\square $\overline{A}(\overline{L})$	$'name' \rightarrow x$	
CONT	Resumes execution of a halted program.	\square $\overline{A}(\overline{N})$		

* = function

Name	Description	Access	Inputs	Outputs
CONVERT	Converts a source unit object to the dimensions of a target unit.		$x1_units$ $x2_units_{target}$ \rightarrow $x3_units_{target}$	
CORR	Returns the correlation coefficient of the independent and dependent data columns in the current statistics matrix.		\rightarrow $X_{correlation}$	
COS	Returns the cosine of the argument.*		$Z \rightarrow \cos Z$	
COSH	Returns the hyperbolic cosine of the argument.*		$Z \rightarrow \cosh Z$	
COV	Returns the sample covariance of the independent and dependent data columns in the current statistics matrix.		\rightarrow $X_{covariance}$	
CR	Prints the contents, if any, of the printer buffer.			
ORDIR	Creates an empty subdirectory with the specified name in the current directory.		'global' \rightarrow	
CROSS	Returns the cross product $C = A \times B$ of vectors A and B.		$[vector]_A$ $[vector]_B \rightarrow [vector]_{A \times B}$	
CSWP	Swaps columns <i>j</i> and <i>j</i> of the argument matrix and returns the modified matrix.		$[[matrix]]_n$ n_{column} $n_{column} \rightarrow [[matrix]]_2$	
CURL	Returns the curl of a three-dimensional vector function.		$[vector]_r$ $[[array]] \rightarrow 'symp_{,i}'$	
CYLIN	Sets Cylindrical coordinate mode.			
C→PX	Converts the specified user-unit coordinates to pixel coordinates.		$(x, y) \rightarrow \{ \#n, \#m \}$	

* = function

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

* = function

Name	Description	Access	Inputs	Outputs
PARTFRAC	Performs partial fraction decomposition on a partial fraction.	POLYNOMIAL	'syml_1' → 'syml_2'	
PATH	Returns a list specifying the path to the current directory.	MEMORY DIRECTORY		→ { HOME directory-name ₁ , ... directory-name _n }
PCAR	Returns the characteristic polynomial of an $n \times n$ matrix.	EIGENVECTORS	[[matrix ₁]] → 'syml_1'	
PCOEF	Returns the coefficients of a monic polynomial having specific roots.	POLYNOMIAL	[array] _{roots} → [array] _{coefficients}	
PCONTOUR	Sets the plot type to PCONTOUR.			
PCOV	Returns the population covariance of the independent and dependent data columns in the current statistics matrix.			→ $x_{population}$
PDIM	Replaces PICT with a blank PICT of the specified dimensions.	PICT	(x_{min} , y_{min}) (x_{max} , y_{max}) →	
PERM	Returns the number of possible permutations of n items taken m at a time.*	PROBABILITY	n m → $P_{n,m}$	
PEVAL	Evaluates an n -degree polynomial at x .		[array] _{coefficients} x → $p(x)$	
PGDIR	Purges the named directory.	MEMORY DIRECTORY	'global' →	
PICK	Copies the contents of a specified level to level 1.	STACK	obj_h ... obj_i n → obj_h ... obj_i obj_i	

* = function

Name	Description	Access	Inputs	Outputs
C→R	Separates the real and imaginary parts of a complex number or complex array.	TYPE	(x , y) → x y	
DARCY	Calculates the Darcy friction factor of certain fluid flows.*		xe/D yRe → x_{Darcy}	
→DATE	Sets the system date to date.	TOOLS	date →	
DATE	Returns the system date.	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.	TOOLS	date ₁ x_{days} → date _{new}	
DEBUG	Starts program execution, then suspends it as if HALT were the first program command.		« program » or 'program name' →	
DDAYS	Returns the number of days between two dates.	TOOLS	date ₁ date ₂ → x_{days}	
DEC	Selects decimal base for binary integer operations. (The default base is decimal).			
DECR	Takes a variable, subtracts 1, stores the new value back into the original variable, and returns the new value.	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.		'name=exp' →	
DEG	Sets the angle mode to degrees.			
DELALARM	Deletes the specified alarm.	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]		$x_{delay} \rightarrow$
DELKEYS	Clears user-defined key assignments.	[CAT]		$x_{key} \rightarrow$
DEPND	Specifies the dependent variable (and its plotting range for TRUTH plots).	[CAT]		'global' \rightarrow
DEPTH	Returns a real number representing the number of objects present on the stack (before DEPTH was executed).	[PRG] STACK		$\rightarrow n$
DERIV	Returns the partial derivatives of a function, with respect to the specified variables.*	[CAT] DERIV AND INTEG		'symp ₁ ' $z \rightarrow$ 'symp ₂ '
DERVX	Returns the derivative of a function with respect to the current variable.*	[CAT] DERIV AND INTEG		'symp ₁ ' \rightarrow 'symp ₂ '
DESOLVE	Solves certain first-order ordinary differential equations with respect to the current variable.	[SOLV]	'symp ₁ ' 'symp ₂ ' \rightarrow 'symp ₃ '	
DET	Returns the determinant of a square matrix.	[MATHS] OPERATIONS	[[matrix]] \rightarrow $x_{determinant}$	
DETACH	Detaches the library with the specified number from the current directory.	[CAT]	$n_{library} \rightarrow$	
-DIAG	Returns a vector that contains the major diagonal elements of a matrix.	[MATHS] CREATE	[[matrix]] \rightarrow [vector] _{diagonals}	

* = function

Name	Description	Access	Inputs	Outputs
NUMX	Sets the number of x-steps for each y-step in 3D perspective plots.	[CAT]		$n_x \rightarrow$
NUMY	Sets the number of y-steps across the view volume in 3D perspective plots.	[CAT]		$n_y \rightarrow$
OBJ \rightarrow	Separates an object into its components.	[PRG] TYPE	(x y) \rightarrow 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	$\#n_1 \#n_2 \rightarrow \#n_3$	
ORDER	Reorders the variables in the current directory (shown in the VAR menu) to the order specified.	[PRG] MEMORY DIRECTORY	{ global ₁ ... global _n } \rightarrow	
OVER	Returns a copy to level 1 of the object on level 2.	[PRG] STACK	obj ₁ obj ₂ \rightarrow obj ₁ obj ₂ obj ₁	
PAZB2	Takes a prime number and returns a Gaussian integer.	[ARITH] INTEGER	$z_1 \rightarrow z_2$	
PARAMETRIC	Sets the plot type to PARAMETRIC.	[CAT]		
PARITY	Sets the parity value in the reserved variable IOPAR.	[CAT]	$n_{parity} \rightarrow$	
PARSURFACE	Sets plot type to PARSURFACE.	[CAT]		

* = function

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

* = 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.	VARIES CREATE	$[array]_{diagonals} \{dim\} \rightarrow [[matrix]]$	
DIFF	Displays a menu of calculus commands.	CAL		
DIFFEQ	Sets the plot type to DIFFEQ.	CAL		
DISP	Displays obj in the n th display line.	PRG OUT	$obj\ n \rightarrow$	
DIV	Returns the divergence of a vector function.	CAL CALD DERIV AND INTEG	$[[array_1]] \ [[array_2]] \rightarrow 'symb_1'$	
DIV2	Performs euclidean division on two expressions. Step-by-step mode is available with this command.	ARITH POLYNOMIAL	$'symb_1' \ 'symb_2' \rightarrow 'symb_3'$	
DIV2MOD	Performs euclidean division on two expressions modulo the current modulus.	ARITH MODULO	$'symb_1' \ 'symb_2' \rightarrow 'symb_3'$	
DIVIS	Returns a list of the divisors of a polynomial or an integer.	ARITH	$'symb_1' \rightarrow \{list\}$	
DIVMOD	Divides two expressions modulo the current modulus.*	ARITH MODULO	$'symb_1' \ z \rightarrow 'symb_2'$	
DIVPC	Returns a Taylor polynomial for the quotient of two expressions.	CAL LIMITS & SERIES	$'symb_1' \ 'symb_2' \ z \rightarrow 'symb_3'$	
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	$n_{error} \rightarrow$	

* = function

Name	Description	Access	Inputs	Outputs
DOLIST	Applies commands, programs, or user-defined functions to lists.	\square $\overline{\text{PROC}}$ LIST \square $\overline{\text{PROC}}$ PROCEDURES	$\{ \text{list}'_1, \dots, \text{list}'_n \} \ll \text{program} \gg \rightarrow \{ \text{results}' \}$	
DOSUBS	Applies a program or command to groups of elements in a list.	\square $\overline{\text{PROC}}$ LIST \square $\overline{\text{PROC}}$ PROCEDURES	$\{ \text{list}'_1, n \ll \text{program} \gg \rightarrow \{ \text{list}'_2 \}$	
DOT	Returns the dot product A·B of two arrays A and B.	\square $\overline{\text{MATH}}$ VECTOR	$[\text{array}_A] [\text{array}_B] \rightarrow x$	
DRAW	Plots the mathematical data in the reserved variable EQ.	\square $\overline{\text{CAT}}$		
DRAW3DMATRIX	Draws a 3D plot from the values in a specified matrix.	\square $\overline{\text{CAT}}$	$[[\text{matrix}]] \quad v_{\min} \quad v_{\max} \rightarrow$	
DRAX	Draws axes in <i>PLOT</i> .	\square $\overline{\text{CAT}}$		
DROP	Removes the level 1 object from the stack.	\square $\overline{\text{PROC}}$ STACK	$\text{obj} \rightarrow$	
DROP2	Removes the first two objects from the stack.	\square $\overline{\text{PROC}}$ STACK	$\text{obj}_1 \quad \text{obj}_2 \rightarrow$	
DROPN	Removes the first $n + 1$ objects from the stack (the first n objects excluding the integer n itself).	\square $\overline{\text{PROC}}$ STACK	$\text{obj}_1 \dots \text{obj}_n \quad n \rightarrow$	
DTAG	Removes all tags (labels) from an object.	\square $\overline{\text{PROC}}$ TYPE	$\text{tag.obj} \rightarrow \text{obj}$	
DUP	Returns a copy of the argument (or the object on level 1).	\square $\overline{\text{PROC}}$ STACK	$\text{obj} \rightarrow \text{obj} \quad \text{obj}$	
DUP2	Returns copies of the two arguments (or the objects on levels 1 and 2 of the stack).	\square $\overline{\text{PROC}}$ STACK	$\text{obj}_2 \quad \text{obj}_1 \rightarrow \text{obj}_2 \quad \text{obj}_1 \quad \text{obj}_2 \quad \text{obj}_1$	
DUPDUP	Duplicates an object twice.	\square $\overline{\text{CAT}}$	$\text{obj} \rightarrow \text{obj} \quad \text{obj} \quad \text{obj}$	

* = function

Name	Description	Access	Inputs	Outputs
MINR	Returns the symbolic constant MINR or its numerical representation, 1.0000000000E-499.*	\square $\overline{\text{MTH}}$ CONSTANTS	$\rightarrow \text{'MINR'}$	
MINΣ	Finds the minimum coordinate value in each of the m columns of the current statistics matrix.	\square $\overline{\text{CAT}}$	$\rightarrow x_{\min}$	
MITM	Changes multiple equation menu titles and order.	\square $\overline{\text{CAT}}$	$\text{'title' } \{ \text{list}' \} \rightarrow$	
MOD	Returns a remainder where $x \bmod y = x - y \text{floor}(x/y)$.*	\square $\overline{\text{MTH}}$ REAL	$x \quad y \rightarrow x \bmod y$	
MODSTO	Changes the modulo setting to the specified number.	\square $\overline{\text{MTH}}$ MODULO	$z_1 \rightarrow z_2$	
MROOT	Uses the multiple-equation solver to solve for one or more variables using the equations in EQ.	\square $\overline{\text{CAT}}$	$\text{'name' } \rightarrow x$	
MSGBOX	Creates a user-defined message box.	\square $\overline{\text{PROC}}$ OUT	$\text{'message' } \rightarrow$	
MSOLVR	Displays the multiple-equation solver variable menu for the set of equations stored in EQ.	\square $\overline{\text{CAT}}$		
MULTMOD	Performs modular multiplication of two objects, modulo the current modulus.*	\square $\overline{\text{MTH}}$ MODULO	$\text{obj}_1 \quad \text{obj}_2 \rightarrow \text{obj}_3$	
MUSER	Designates a variable as user-defined for the multiple-equation solver.	\square $\overline{\text{CAT}}$	$\text{'name' } \rightarrow$	
—NDISP	Sets the number of lines over which an object is displayed.	\square $\overline{\text{CAT}}$	$n \rightarrow$	
NDIST	Returns the normal probability distribution at x based on the mean m and variance v of the normal distribution.	\square $\overline{\text{MTH}}$ PROBABILITY	$m \quad v \quad x \rightarrow \text{ndist}(m, v, x)$	

* = function

Name	Description	Access	Inputs	Outputs
ENG	Sets the number display format to engineering mode.			
EPSX0	Replaces with zero those coefficients in a polynomial that have an absolute value less than the variable EPS.		$n \rightarrow$	'symbol' ₁ \rightarrow 'symbol' ₂
EQW	Opens Equation Writer, where you can edit an expression.			$exp_1 \rightarrow exp_2$
EQ→	Separates an equation into its left and right sides.		TYPE	'symbol'=symbol' ₂ \rightarrow 'symbol' ₁ 'symbol' ₂
ERASE	Erases <i>PICIT</i> , leaving a blank <i>PICIT</i> of the same dimensions.			
ERR0	Clears the last error number (and message) so that a subsequent execution of ERRN returns # 0h.		ERROR	
ERRM	Returns a string containing the error message of the most recent calculator error.		ERROR	\rightarrow "error message"
ERRN	Returns the error number of the most recent calculator error.		ERROR	\rightarrow #n _{error}
EULER	Returns the number of integers less than a specified integer that are co-prime with the integer.*		INTEGER	$Z_1 \rightarrow Z_2$
EVAL	Evaluates the object.			obj \rightarrow
EXLR	Returns the left- and right-hand sides of an equation as discrete expressions.			'symbol' ₁ \rightarrow 'symbol' ₂ 'symbol' ₃
EXP	Returns the exponential, or natural antilogarithm, of the argument; that is, <i>e</i> raised to the given power.*		e^z	$z \rightarrow e^z$

* = function

Name	Description	Access	Inputs	Outputs
LQ	Returns the LQ factorization of an $m \times n$ matrix.		FACTORIZATION	$[[matrix]]_A \rightarrow [[matrix]]_L [[matrix]]_Q [[matrix]]_P$
LR	Uses currently selected statistical model to calculate the linear regression coefficients (intercept and slope).			\rightarrow Intercept: x_1 Slope: x_2
LSQ	Returns the minimum norm least squares solution to any system of linear equations where $A \times X = B$.		OPERATIONS	$[array]_B [[matrix]]_A \rightarrow [array]_X$
LU	Returns the LU decomposition of a square matrix.		FACTORIZATION	$[[matrix]]_A \rightarrow [[matrix]]_L [[matrix]]_U [[matrix]]_P$
LVAR	Returns a list of variables in an algebraic object.			$obj_1 \rightarrow obj_2 [vector,]$
MAD	Returns details of a square matrix.		OPERATIONS	$[[array_1]] \rightarrow$ 'symbol' ₁ 'symbol' ₂ 'symbol' ₃
MAIN	Displays a menu of CAS categories.			$x \rightarrow y_{mant}$
MANT	Returns the mantissa of the argument.*		REAL	$\{list, \}$ «program» $\rightarrow \{list, \}$
MAP	Applies a specified program to a list of objects or values.			'symbol', { 'symbol' _{part} , 'symbol' _{rep} } \rightarrow 'symbol' ₂ 0/1
↑MATCH	Rewrites an expression that matches a specified pattern.			'symbol', { 'symbol' _{part} , 'symbol' _{rep} } \rightarrow 'symbol' ₂ 0/1
↓MATCH	Like ↑MATCH, but works top-down not bottom-up.			
MATR	Displays a menu of matrix commands.			
MAX	Returns the greater of two inputs.*		REAL	$x \ y \rightarrow \max(x,y)$

* = function

Name	Description	Access	Inputs	Outputs
LININ	Tests whether an algebraic is structurally linear for a given variable.*	\square $\overline{\text{PRG}}$ TEST	'sybm' 'name' \rightarrow 0/1	
LINSOLVE	Solves a system of linear equations.	\square $\overline{\text{SOL}}$	$[[\text{array}_1,]] [\text{vector}_1] \rightarrow \text{'sybm}_1' \{ \text{list}_1 \} \text{'sybm}_2' \{ \text{list}_2 \} \rightarrow z$	
Σ LIST	Returns the sum of the elements in a list.	\square $\overline{\text{MTH}}$ LIST	$\{ \text{list} \} \rightarrow z$	
Δ LIST	Returns the first differences of the elements in a list.	\square $\overline{\text{MTH}}$ LIST	$\{ \text{list} \} \rightarrow \{ \text{differences} \}$	
Π LIST	Returns the product of the elements in a list.	\square $\overline{\text{MTH}}$ LIST	$\{ \text{list} \} \rightarrow z$	
\rightarrow LIST	Takes n specified objects and returns a list of those objects.	\square $\overline{\text{CAT}}$	$\text{obj}_1 \dots \text{obj}_n \rightarrow \{ \text{obj}_1 \dots \text{obj}_n \}$	
LIST \rightarrow	Takes a list of n objects and returns each object separately, and returns the total number of objects to item.	\square $\overline{\text{CAT}}$	$\{ \text{obj}_1 \dots \text{obj}_n \} \rightarrow \text{obj}_1 \dots \text{obj}_n \ n$	
LN	Returns the natural (base e) logarithm of the argument.*	\square $\overline{\text{LN}}$	$z \rightarrow \ln z$	
LNAME	Returns the variable names in a symbolic expression.	\square $\overline{\text{CAT}}$	'sybm ₁ ' $\rightarrow [\text{vector}_1]$	
LNCOLLECT	Simplifies an expression by collecting logarithmic terms.	\square $\overline{\text{ALG}}$	'sybm ₁ ' $\rightarrow \text{'sybm}_2'$	
LNPI	Returns $\ln (x + 1)$.*	\square $\overline{\text{MTH}}$ HYPERBOLIC	$x \rightarrow \ln (x + 1)$	
LOG	Returns the common logarithm (base 10) of the argument.*	\square $\overline{\text{LOG}}$	$z \rightarrow \log z$	
LOGFIT	Stores LOGFIT in Σ PAR. Subsequent executions of LR will use the log curve-fitting model.	\square $\overline{\text{CAT}}$		

* = function

Name	Description	Access	Inputs	Outputs
EXPAN	Expands and simplifies an algebraic expression.	\square $\overline{\text{CAT}}$	'sybm ₁ ' $\rightarrow \text{'sybm}_2'$	
EXPAND	Expands and simplifies an algebraic expression.	\square $\overline{\text{ALG}}$	'sybm ₁ ' $\rightarrow \text{'sybm}_2'$	
EXPANDMOD	Expands and simplifies an expression, modulo the current modulus.*	\square $\overline{\text{ARITH}}$ MODULO	'sybm ₁ ' $\rightarrow \text{'sybm}_2'$	
EXPFIT	Stores EXPFIT in Σ PAR, thus subsequent executions of LR will use the exponential curve fitting model.	\square $\overline{\text{CAT}}$		
EXPLN	Transforms the trigonometric terms in an expression to exponential and logarithmic terms.	\square $\overline{\text{QWERT}}$	'sybm ₁ ' $\rightarrow \text{'sybm}_2'$	
EXPM	Returns $e^x - 1$.*	\square $\overline{\text{MTH}}$ HYPERBOLIC	$x \rightarrow e^x - 1$	
EYEPT	Specifies the coordinates of the eye point in a perspective plot.	\square $\overline{\text{CAT}}$	$x_{\text{point}} \ y_{\text{point}} \ z_{\text{point}} \rightarrow$	
F0 λ	Returns the fraction of total black-body emissive power at temperature x_T between wavelengths 0 and y_{lambda} .*	\square $\overline{\text{CAT}}$ F0 λ	$y_{\text{lambda}} \ x_T \rightarrow x_{\text{power}}$	
FACT	FACT is the same as !. See !.	\square $\overline{\text{MTH}}$ PROBABILITY	$n \rightarrow n!$	
FACTOR	Factorizes a polynomial or an integer.	\square $\overline{\text{ALG}}$	'sybm ₁ ' $\rightarrow \text{'sybm}_2'$	
FACTORMOD	Factorizes a polynomial modulo the current modulus. The modulus must be less than 100, and a prime number.*	\square $\overline{\text{ARITH}}$ MODULO	'sybm ₁ ' $\rightarrow \text{'sybm}_2'$	

* = function

Name	Description	Access	Inputs	Outputs
FACTORS	For a value or expression, returns a list of prime factors and their multiplicities.	[A]RITH		$z \rightarrow \{ /list \}$
FANNING	Calculates the Fanning friction factor of certain fluid flows.*	[C]A	$x_{\text{fID}} \ y_{\text{Re}} \rightarrow x_{\text{fanning}}$	
FAST3D	Sets the plot type to FAST 3D.	[C]A		
FCOEF	From an array of roots and multiplicities/poles, returns a rational polynomial with a leading coefficient of 1.	[A]RITH POLYNOMIAL	$[/array_1] \rightarrow 'symbol_1'$	
FC?	Tests whether the specified system or user flag is clear, and returns a corresponding test result.	[P]B TEST	$n_{\text{flagnumber}} \rightarrow 0/1$	
FC?C	Tests whether the specified system or user flag is clear, returns a corresponding test result, and then clears the flag.	[P]B TEST	$n_{\text{flagnumber}} \rightarrow 0/1$	
FFT	Computes the one- or two-dimensional discrete Fourier transform of an array.	[V]H FFT	$[/array]_1 \rightarrow [/array]_2$	
FILER	Opens File Manager.	[F]ES		
FINDALARM	Returns the alarm index n_{index} of the first alarm due after the specified time.	[T]IME TOOLS ALRM	$date \rightarrow n_{\text{index}}$	
FINISH	Terminates Kermit Server mode in a connected device.	[C]A		
FIX	Sets the number display format to fix mode, which rounds the display to n decimal places.	[C]A	$n \rightarrow$	

* = function

Name	Description	Access	Inputs	Outputs
LXCM	From a program with two arguments, builds a matrix with the specified number of rows and columns, with $a_{ij} = i/j_1$.	[C]A	$n_1 \ n_2$ «program» $\rightarrow [/matrix;]$	
LDEC	Solves a linear differential equation with constant coefficients.	[S]LV	$'symbol_1' \ 'symbol_2' \rightarrow 'symbol_3'$	
LEGENDRE	Returns the n th degree Legendre polynomial.*	[A]RITH POLYNOMIAL	$n_1 \rightarrow 'symbol_1'$ $\{ /list \} \rightarrow \{ /list \} \ z_1$	
LGCD	Returns the greatest common divisor of a list of expressions or values.*	[A]RITH	$\#n_{\text{function}} \rightarrow$ $\rightarrow ['title' \ n_{\text{lb}} \ n_{\text{point}} \ ... 'title' \ n_{\text{lb}} \ n_{\text{point}}]$	
LIBEVAL	Evaluates unnamed library functions.	[C]A		
LIBS	Lists the title, number, and port of each library attached to the current directory.	[C]A	$'symbol_1' \ 'symbol_2' \rightarrow 'symbol_3'$	
LIMIT	Returns the limit of a function as it approaches a specified value.*	[C]A LIMITS & SERIES	$'symbol_1' \rightarrow 'symbol_2'$ $(x_1, y_1) \ (x_2, y_2) \rightarrow$ $\rightarrow 'symbol_{\text{formula}}'$	
LIN	Linearizes expressions involving exponential terms.	[P]VAL		
LINE	Draws a line in <i>P/ICT</i> between the input coordinates.	[P]B PICT		
ΣLINE	Returns an expression representing the best fit line according to the current statistical model.	[C]A		
LINFIT	Stores LINFIT in the reserved variable Σ <i>PAR</i> . Subsequent executions of LR will use the linear curve fitting model.	[C]A		

* = function

Name	Description	Access	Inputs	Outputs
KILL	Cancels all currently halted programs. If KILL is executed within a program, that program is also canceled.	RUN & DEBUG		
LABEL	Labels axes in <i>PICT</i> with variable names and with the minimum and maximum values of the display ranges.			
LAGRANGE	Returns the interpolating polynomial of minimum degree for a pair of values.	POLYNOMIAL	$[[matrix_1]] \rightarrow 'symbol_1'$	
LANGUAGE→	Returns a value indicating the message language.		$\rightarrow z$	
→LANGUAGE	Sets the language used in messages.		$z \rightarrow$	
LAP	Performs a Laplace transform on an expression with respect to the current default variable.*	DIFFERENTIAL EONS	$'symbol_1' \rightarrow 'symbol_2'$	
LAPL	Returns the Laplacian of a function with respect to a vector of variables.	DERIV & INTEG	$'symbol_1' [vector_1] \rightarrow 'symbol_2'$	
LASTARG	Returns copies of the arguments of the most recently executed command.*	ERROR	$\rightarrow obj_1 \dots obj_1$	
→LCD	Displays the specified graphics object with its upper left pixel in the upper left corner of the display.		$grob \rightarrow$	
LCD→	Returns the current stack and menu display as a 131 x 64 graphics object.	GROB	$\rightarrow grob$	
LCM	Returns the least common multiple of two objects.*	POLYNOMIAL	$'symbol_1' 'symbol_2' \rightarrow 'symbol_3'$	

* = function

Name	Description	Access	Inputs	Outputs
FLASHEVAL	Evaluates unnamed flash functions.		$\#n_{function} \rightarrow$	
FLOOR	Returns the greatest integer that is less than or equal to the argument.*	REAL	$x \rightarrow n$	
→FONT	Returns the current system font.		$\rightarrow obj$	
FONT6	Returns the system FONT 6 object.		$\rightarrow obj$	
FONT7	Returns the system FONT 7 object.		$\rightarrow obj$	
FONT8	Returns the system FONT 8 object.		$\rightarrow obj$	
FONT→	Sets the system font.*		$obj \rightarrow$	
FOR	Starts FOR ... NEXT and FOR ... STEP definite loop structures.	BRANCH	$FOR x_{start} x_{finish} \rightarrow$	
FOURIER	Returns the n^{th} coefficient of a complex Fourier series expansion.*	DERIV. & INTEG	$'symbol_1' z_1 \rightarrow z_2$	
FP	Returns the fractional part of the argument.*	REAL	$x \rightarrow y$	
FREEZE	Freezes the specified part of the display so that it is not updated until a key is pressed.	OUT	$n_{display area} \rightarrow$	
FROOTS	For a rational polynomial, returns an array of its roots and poles, with their corresponding multiplicities.	POLYNOMIAL	$'symbol_1' \rightarrow [[array_1]]$	

* = function

Name	Description	Access	Inputs	Outputs
FS?	Tests whether the specified system or user flag is set, and returns a corresponding test result.	\square $\overline{\text{PFS}}$ TEST	$n_{\text{flagnumber}} \rightarrow 0/1$	
FS7C	Tests whether the specified system or user flag is set, returns a corresponding test result, then clears the flag.	\square $\overline{\text{PFS}}$ TEST	$n_{\text{flagnumber}} \rightarrow 0/1$	
FUNCTION	Sets the plot type to FUNCTION.	\square $\overline{\text{CAT}}$		
FXND	Splits an object into a numerator and a denominator.	\square $\overline{\text{CAT}}$	'symb ₁ ' \rightarrow 'symb ₂ ' 'symb ₃ '	
GAUSS	Returns the diagonal representation of a quadratic form.	\square $\overline{\text{GAUSS}}$ QUADRATIC FORM	'symb ₁ ' [vector ₁] \rightarrow [[array ₁]] ['symb ₂ ' { list ₁ }	
GCD	Returns the greatest common divisor of two objects.	\square $\overline{\text{ARITH}}$ POLYNOMIAL	'symb ₁ ' 'symb ₂ ' \rightarrow z	
GCDMOD	Finds the greatest common divisor of two polynomials modulo the current modulus.*	\square $\overline{\text{ARITH}}$ MODULO	'symb ₁ ' 'symb ₂ ' \rightarrow 'symb ₃ '	
GET	Retrieves the specified object from a matrix, a list, or an array.	\square $\overline{\text{PFS}}$ LIST ELEMENTS	[[matrix]] $n_{\text{position}} \rightarrow z_{\text{get}}$	
GETI	Retrieves the specified object from a matrix, a list, or an array, and the index of the next object.	\square $\overline{\text{PFS}}$ LIST ELEMENTS	[[matrix]] $n_{\text{position}1} \rightarrow$ [[matrix]] $n_{\text{position}2} z_{\text{get}}$	
GOR	Superimposes <i>grob₁</i> onto <i>grob_{target}</i> .	\square $\overline{\text{PFS}}$ GROB	<i>grob_{target}</i> { #n #m } <i>grob₁</i> \rightarrow <i>grob_{result}</i>	
GRAD	Sets Grads angle mode.	\square $\overline{\text{MODE}}$		
GRIDMAP	Sets the plot type to GRIDMAP.	\square $\overline{\text{CAT}}$		

* = function

Name	Description	Access	Inputs	Outputs
IP	Returns the integer part of the argument.*	\square $\overline{\text{MATH}}$ REAL	$x \rightarrow n$	
IQUOT	Returns the integer quotient of two integers.*	\square $\overline{\text{ARITH}}$ INTEGER	$n_1 n_2 \rightarrow n_3$	
IREMAINDER	Returns the remainder of an integer division.*	\square $\overline{\text{CAT}}$	$n_1 n_2 \rightarrow n_3$	
ISOL	Returns an algebraic <i>symb₂</i> that rearranges <i>symb₁</i> to isolate the first occurrence of variable <i>global</i> .	\square $\overline{\text{SOLV}}$	'symb ₁ ' 'global' \rightarrow 'symb ₂ '	
ISPRIME?	Tests if a number is prime.*	\square $\overline{\text{ARITH}}$ INTEGER	<i>obj₁</i> \rightarrow T/F	
I→R	Converts an integer into a real number.*	\square $\overline{\text{CAT}}$	$n \rightarrow z$	
JORDAN	Computes the eigenvalues, eigenvectors, minimum polynomial, and characteristic polynomial of a matrix.	\square $\overline{\text{GAUSS}}$ EIGENVECTORS	[[matrix ₁]] \rightarrow 'symb ₁ ' 'symb ₂ ' { list ₁ } [[array ₁]]	
KERRM	Returns the text of the most recent Kermit error packet.	\square $\overline{\text{CAT}}$	\rightarrow "error message"	
KEY	Suspends program execution until a key is pressed, then returns the row-column location <i>x_{rm}</i> of that key.	\square $\overline{\text{PFS}}$ IN	$\rightarrow x_{r,m} 1$	
KEYEVAL	Actions the specified key press.	\square $\overline{\text{CAT}}$	<i>rc</i> . <i>p₁</i> \rightarrow	
→KEYTIME	Sets a new keytime value, or the time in ticks after a keypress until another key is actioned.	\square $\overline{\text{CAT}}$	<i>time</i> \rightarrow	
KEYTIME→	Displays the current keytime value.	\square $\overline{\text{CAT}}$	\rightarrow <i>time</i>	
KGET	Used by a local Kermit to get a Kermit server to transmit the named object(s).	\square $\overline{\text{CAT}}$	'name' \rightarrow	
















* = function

Name	Description	Access	Inputs	Outputs
IFTE	Executes the <i>obj</i> in argument 2 or level 2 if <i>T/F</i> is nonzero. Executes the <i>obj</i> in argument 3 or level 1 if <i>T/F</i> is zero.*	<div>▢ PRG BRANCH</div>	<i>T/F</i> <i>obj_{true}</i> <i>obj_{false}</i> →	
ILAP	Returns the inverse Laplace transform of an expression. The expression must evaluate to a rational fraction.*	<div>▢ CAL DIFFERENTIAL EONS</div>	' <i>symb₁</i> ' → ' <i>symb₂</i> '	
IM	Returns the imaginary part of its complex argument.*	<div>▢ APLY</div>	$x \rightarrow 0$	
INCR	Takes a variable, adds 1, stores the new value back into the original variable, and returns the new value.	<div>▢ PRG MEMORY ARITHMETIC</div>	' <i>name</i> ' → <i>x_{increment}</i>	
INDEP	Specifies the independent variable and its plotting range.	<div>OUT</div>	' <i>global</i> ' →	
INFORM	Creates a user-defined input form (dialog box).	<div>▢ PRG IN</div>	" <i>title</i> " (<i>s₁</i> , <i>s₂</i> , ... <i>s_n</i>) <i>format</i> { <i>resets</i> } [<i>init</i>]	→ { <i>vals</i> } 1
INPUT	Prompts for data input to the command line and halts stack or command line operations.	<div>▢ PRG IN</div>	" <i>stack prompt</i> " " <i>command-line prompt</i> "	→ " <i>result</i> "
INT	Calculates the antiderivative of a function for a given variable, at a given point.*	<div>OUT</div>	' <i>symb₁</i> ' ' <i>symb₂</i> ' ' <i>symb₃</i> ' → ' <i>symb₄</i> '	
INTVX	Finds the antiderivative of a function symbolically, with respect to the current default variable.*	<div>▢ CAL DERIV. & INTEG</div>	' <i>symb₁</i> ' → ' <i>symb₂</i> '	
INV	Returns the reciprocal or the matrix inverse.*	<div>IN</div>	$z \rightarrow 1/z$	
INVMOD	Performs modular inversion on an object modulo the current modulus.*	<div>▢ PRG MODULO</div>	<i>obj₁</i> → <i>obj₁</i>	






























* = function

Name	Description	Access	Inputs	Outputs
→GROB	Creates a graphics object from a specified object, where the argument <i>n_{char}</i> size specifies the size of the object.	<div>OUT</div>	<i>obj</i> <i>n_{char}</i> → <i>grob</i>	
GROBADD	Combines two graphic objects.	<div>OUT</div>	<i>grob₁</i> <i>grob₂</i> → <i>grob₃</i>	
GXOR	Superimposes <i>grob₁</i> onto <i>grob_{target}</i> .	<div>▢ PRG GROB</div>	<i>grob₁</i> <i>grob₂</i> → <i>grob_{result}</i>	
HADAMARD	Performs an element by element multiplication of two matrices (Hadamard product).	<div>▢ PRG OPERATIONS</div>	<i>grob_{target}</i> { <i>#n #m</i> } [[<i>matrix₁</i>]] [[<i>matrix₂</i>]] → [[<i>matrix₃</i>]]	
HALFTAN	Replaces sin(<i>x</i>), cos(<i>x</i>) and tan(<i>x</i>) terms with tan(<i>x/2</i>) terms.	<div>▢ PRG</div>	' <i>symb₁</i> ' → ' <i>symb₂</i> '	
HALT	Halts program execution.	<div>▢ PRG RUN & DEBUG</div>		
HEAD	Returns the first element of a list or string.	<div>▢ PRG CHARS</div>	{ <i>obj₁</i> ... <i>obj_n</i> } → <i>obj₁</i>	
HEADER→	Returns the size, in lines, of the display header.	<div>OUT</div>	→ <i>z</i>	
→HEADER	Sets the size, in lines, of the display header.	<div>OUT</div>	$z \rightarrow$	
HERMITE	Returns the <i>n</i> th Hermite polynomial.*	<div>▢ PRG</div>	$z \rightarrow$ ' <i>symb₁</i> '	
HESS	Returns the Hessian matrix and the gradient of an expression with respect to the specified variables.	<div>POLYNOMIAL ▢ CAL DERIV & INTEG</div>	' <i>symb₁</i> ' [<i>vector₁</i>] → [[<i>matrix</i>]]	<i>z</i> [<i>vector₂</i>]
HEX	Sets hexadecimal base for binary integer operations.	<div>OUT</div>		

* = function

Name	Description	Access	Inputs	Outputs
HILBERT	Returns a square Hilbert matrix of the specified order.			$z \rightarrow [[\text{matrix}]]$
HISTOGRAM	Sets the plot type to HISTOGRAM.			
HISTPLOT	Plots a frequency histogram.			
→HMS	Converts a real number representing hours or degrees with a decimal fraction to hours-minutes-seconds format.	 	$x \rightarrow \text{HMS}$	
HMS-	Returns the difference of two real numbers, where the arguments and the result are interpreted in hours-minutes-seconds format.	 	$\text{HMS}_1, \text{HMS}_2 \rightarrow \text{HMS}_1 - \text{HMS}_2$	HMS_2
HMS+	Returns the sum of two real numbers, where the arguments and the result are interpreted in hours-minutes-seconds format.	 	$\text{HMS}_1, \text{HMS}_2 \rightarrow \text{HMS}_1 + \text{HMS}_2$	HMS_2
HMS→	Converts a real number in hours-minutes-seconds format to its decimal form.	 	$\text{HMS} \rightarrow x$	
HOME	Makes the HOME directory the current directory.			
HORNER	Executes a Horner scheme on a polynomial.	 	$\text{'symbol' } z_1 \rightarrow \text{'symbol' } z_2, z_3$	z_3
i	Returns the symbolic constant <i>i</i> or its numerical representation, (0, 1).*			i

* = function

Name	Description	Access	Inputs	Outputs
IABCUV	Returns a solution in integers <i>u</i> and <i>v</i> of $au + bv = c$, where <i>a</i> , <i>b</i> , and <i>c</i> are integers.	  	$n_1, n_2, n_3 \rightarrow z_1, z_2$	
IBERNOULLI	Returns the <i>n</i> th Bernoulli number for a given integer.*		$n_1 \rightarrow z_1$	
IBP	Performs integration by parts on a function.	  	$\text{'symbol' } \text{'symbol' } \rightarrow \text{'symbol' } \text{'symbol' }$	'symbol'
ICHINREM	Solves a system of two congruences in integers using the Chinese Remainder theorem.	  	$[\text{vector}_1] [\text{vector}_2] \rightarrow [\text{vector}_3]$	
IDIV2	For two integers, <i>a</i> and <i>b</i> , returns the integer part of <i>a/b</i> , and the remainder, <i>r</i> .	  	$n_1, n_2 \rightarrow n_3, n_4$	
IDN	Returns an identity matrix.		$n \rightarrow [[\text{R-matrix}_{\text{identity}}]]$	
IEGCD	Given two integers <i>x</i> and <i>y</i> , returns three integers, <i>a</i> , <i>b</i> , and <i>c</i> , such that: $ax + by = c$.	  	$n_1, n_2 \rightarrow n_3, n_4, n_5$	
IF	Starts IF ... THEN ... END and IF ... THEN ... ELSE ... END conditional structures.	  		
IFERR	Starts IFERR ... THEN ... END and IFERR ... THEN ... ELSE ... END error trapping structures.	  		
IFFT	Computes the one- or two-dimensional inverse discrete Fourier transform of an array.	  	$[\text{array}]_1 \rightarrow [\text{array}]_2$	
IFT	Executes <i>obj</i> if <i>T/F</i> is nonzero; discards <i>obj</i> if <i>T/F</i> is zero.	  	$T/F \text{ obj} \rightarrow$	

* = function