



HP 48SX

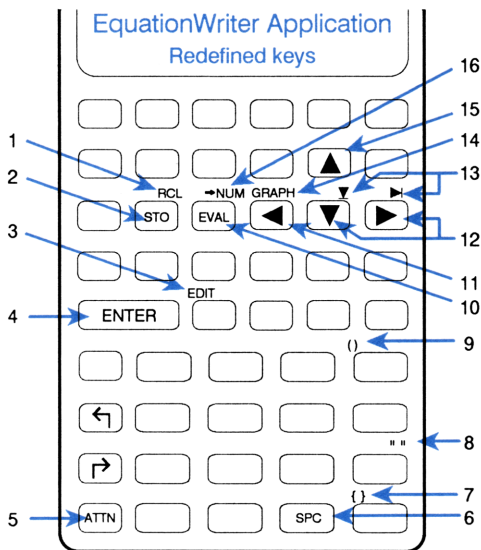
Scientific Expandable

**Quick Reference
Guide**

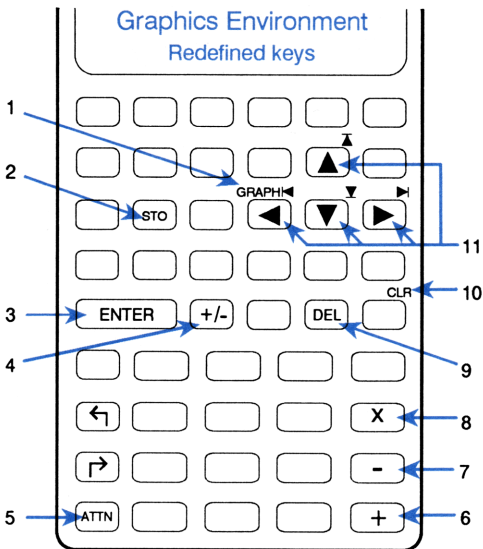
Contents

- EquationWriter Application Keyboard
- Graphics Environment Keyboard
- Interactive Stack Keyboard
- No Command Line Keyboard
- System Flags
- System Operations
- PPAR Parameters
- Σ PAR Parameters
- Printing and PRTPAR Parameters
- Data Transfer and IOPAR Parameters
- EquationWriter Application and Plotting Example
- HP Solve Application Example
- Program Branch Structures
- Object Types and Delimiters
- Command Reference

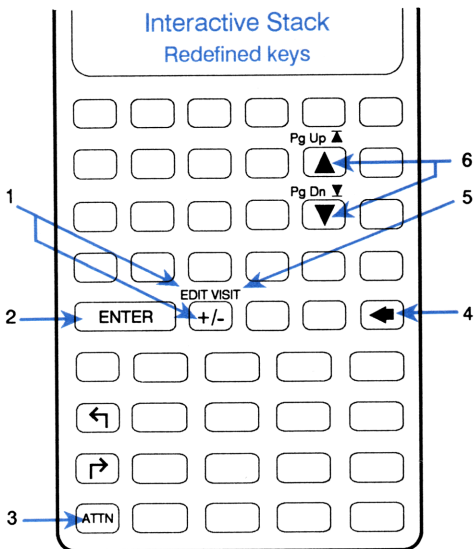
© Hewlett-Packard Co. 1990. All rights reserved.



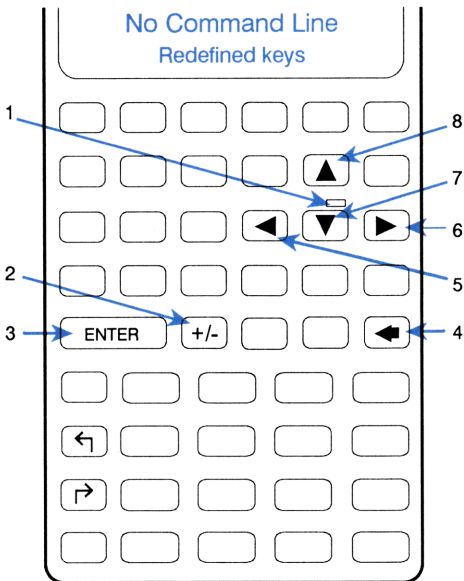
1. Inserts level 1 algebraic or string into expression.
2. Returns expression to stack as graphics object.
3. Edits expression.
4. Returns expression to stack and exits application.
5. Exits application.
6. Enters , or ; separator.
7. Switches implicit parentheses on and off.
8. Returns expression to stack as string.
9. Begins parenthesized term.
10. Exits application; performs EVAL on expression.
11. Invokes Selection environment.
12. Ends subexpression.
13. Ends all subexpressions.
14. Invokes scrolling mode.
15. Starts numerator.
16. Exits application; performs →NUM on expression.



1. Invokes scrolling mode.
2. Copies *PICT* to stack as graphics object.
3. Puts cursor-position coordinates on stack.
4. Switches cursor style.
5. Exits Graphics environment.
6. Switches cursor coordinates on or off.
7. Switches menu labels on or off.
8. Sets mark.
9. Erases rectangular region defined by mark and cursor.
10. Erases *PICT*.
11. Moves Graphics cursor.



1. Edits object in current level.
2. Exits Interactive Stack.
3. Exits Interactive Stack.
4. Deletes object in current level.
5. Edits object stored in variable in current level.
6. Moves stack-level pointer.



1. Right- shifted key views object stored in level 1 variable.
2. Negate key.
3. Duplicates object in level 1.
4. Drops level 1 object.
5. Selects Graphics environment.
6. Swaps levels 1 and 2 objects.
7. Views level 1 object.
8. Selects Interactive Stack.

System Flags

Flag	Description	Clear	Set	Default
Symbolic Math Flags				
- 1	Principal Solution	Returns general solutions	Returns principal solution	Clear
- 2	Symbolic Constants	Symbolic evaluation	Numeric evaluation	Clear
- 3	Numeric Results	Symbolic results	Numeric results	Clear
- 4 is not used.				
Binary Integer Math Flags				64
- 5 thru - 10	Binary Integer Wordsize Combined states of flags - 5 through - 10 set the wordsize from 1 to 64 bits. Flag - 10 is the most significant bit.			Set

Flag	Description	Clear	Set	Default
Binary Integer Base				
– 11 and – 12	DEC HEX OCT BIN	– 11, – 12 – 12 – 11	– 11, – 12 – 11 – 12	Clear
– 13 and – 14 are not used.				
Coordinate System Flags				
– 15 and – 16	Rectangular mode Polar/Cylindrical mode Polar/Spherical mode	– 15, – 16 – 15	– 16 – 15, – 16	Clear

Flag	Description	Clear	Set	Default
Trigonometric Angle Mode Flags				
– 17 and – 18	Degrees mode Radians mode Grads mode	– 17, – 18 – 18 – 17	– 17 – 18	Clear
– 19	Complex Mode	Creates 2D vector	Creates complex number	Clear
Math Exception-Handling Flags				
– 20	Underflow Exception	Returns 0	Errors	Clear
– 21	Overflow Exception	Returns $\pm 9.999999999999999E499$	Errors	Clear
– 22	Infinite Result Exception	Errors	Returns $\pm 9.999999999999999E499$	Clear
– 23	Negative Underflow Indicator	No occurrence	Occurrence	Clear

Flag	Description	Clear	Set	Default
- 24	Positive Underflow Indicator	No occurrence	Occurrence	Clear
- 25	Overflow Indicator	No occurrence	Occurrence	Clear
- 26	Infinite Result Indicator	No occurrence	Occurrence	Clear
- 27 through - 29 are not used.				
Plotting and Graphics Flags				
- 30	Function Plotting ($y=f(x)$)	One plot is drawn: $f(x)$	Separate plots drawn: y and $f(x)$	Clear
- 31	Curve Filling	Curve filling on	Curve filling off	Clear
- 32	Graphics Cursor	Black cursor	Inverting cursor	Clear
I/O and Printing Flags				
- 33	I/O Device	Selects serial port	Selects IR port	Clear
- 34	Printing Device	Selects IR port	Selects serial port	Clear

Flag	Description	Clear	Set	Default
– 35	I/O Data Format	ASCII form	Memory image form	Clear
– 36	RECV Overwrite	Creates new variable	Overwrites existing variable	Clear
– 37	Double-Spaced Printing	Single-spaced printing	Double-spaced printing	Clear
– 38	Linefeed	Adds linefeeds	Does not add linefeeds	Clear
– 39	I/O Messages	Displayed	Suppressed	Clear
Time Management Flags				
– 40	Clock Display	Displayed with TIME menu	Displayed always	Clear
– 41	Clock Format	12-hour clock	24-hour clock	Clear
– 42	Date Format	MM/DD/YY	DD.MM.YY	Clear

Flag	Description	Clear	Set	Default
- 43	Repeat Alarms Not Rescheduled	Alarms rescheduled	Alarms not rescheduled	Clear
- 44	Acknowledged Alarms Saved	Alarms deleted	Alarms saved	Clear
Display Format Flags				
- 45 thru - 48	Number of Decimal Digits Combined states of flags - 45 through - 48 set number of decimal digits in Fix, Scientific, and Engineering modes.			Clear
- 49 and - 50	Number Display Format Standard Fix Scientific Engineering	- 49, - 50 - 50 - 49	- 49 - 50 - 49, - 50	Clear
- 51	Fraction Mark	Period	Comma	Clear
- 52	Single-Line Display	Multiline display of level 1	Single-line display of level 1	Clear

Flag	Description	Clear	Set	Default
- 53	Precedence	Some parentheses displayed	All parentheses displayed	Clear
- 54 is not used.				
Miscellaneous Flags				
- 55	Last Argument	Arguments saved	Arguments not saved	Clear
- 56	Error Beep	Beep on	Beep suppressed	Clear
- 57	Alarm Beep	Beep on	Beep suppressed	Clear
- 58	Verbose Messages	Messages displayed	Messages suppressed	Clear
- 59	Fast Catalog Display	Shows name and equation	Shows name only	Clear
- 60	Alpha Lock	☐ twice to lock	☐ once to lock	Clear
- 61	User-Mode Lock	☐ twice to lock	☐ once to lock	Clear

Flag	Description	Clear	Set	Default
-62	User Mode	Not active	Active	Clear
-63	Vectored ENTER	Normal ENTER functionality	User-defined ENTER active	Clear
-64	Index Wrap Indicator	No wrap occurred	Wrap occurred	Clear

System Operations

- To *adjust the display contrast*, simultaneously press **ON** and either **+** or **-**.
- To perform a *system halt* (halts running programs and applications, erases stack and *PICT* only), simultaneously press **ON** and the third key from the left in the top row. Release keys.
- To *clear all memory* (erases *ALL* stored data and merged memory), simultaneously press **ON** and both outer keys in the top row. Release keys.
- To *print the contents of the current display*, simultaneously press **ON** and **MTH**. Release keys.

PPAR Parameters $\{(x_{\min}, y_{\min}) (x_{\max}, y_{\max}) indep res axes ptype depend\}$

Parameter	Description	Parameter	Description
$(x_{\min}, y_{\min})$	Lower-left pixel coordinates. Default is $(-6.5, -3.1)$.	<i>axes</i>	Axes intersection coordinates. Default is $(0,0)$.
$(x_{\max}, y_{\max})$	Upper-right pixel coordinates. Default is $(6.5, 3.2)$.	<i>ptype</i>	Plot type (FUNCTION, CONIC, POLAR, PARAMETRIC, TRUTH, BAR, HISTOGRAM, or SCATTER). Default is FUNCTION.
<i>indep res</i>	Independent variable. Default is X. Resolution (point spacing). Default is 0 (point plotted in every pixel column).	<i>depend</i>	Dependent variable. Default is Y.

ΣPAR Parameters {*indep depend intercept slope model*}

Parameter	Description	Parameter	Description
<i>indep</i> <i>depend</i> <i>intercept</i>	Independent column number. Default is 1. Dependent column number. Default is 2. Intercept of current regression model. Default is 0.	<i>slope</i> <i>model</i>	Slope of current regression model. Default is 0. LINFIT, LOGFIT, EXPFIT, or PWRFIT. Default is LINFIT.

Printing

Printing with the HP 82240B

1. Position HP 48 and HP 82240B less than 18 inches apart and with IR ports aligned. See printer owner's manual for additional information.
2. Select PRINT menu ( ).
3. Execute desired command by pressing corresponding menu key.

Printing to the Serial Port

1. Connect HP 48 to serial printer.
2. Set flag -34; clear flag -33.
3. Set parameters in I/O SETUP menu (  .
4. If printer uses XON/XOFF handshaking, set *IOPAR* parameter transmit-pacing $\neq 0$.
5. If printer requires line termination sequence other than carriage-return/line-feed, enter sequence as last list item in *PRTPAR*.

PRTPAR Parameters {*delay remap linelength lineterm*}

Parameter	Description	Parameter	Description
<i>delay</i>	Time (≤ 6.9 seconds) delay between printed lines. Default is 1.8 seconds.	<i>linelength</i>	Print line length. Does not affect IR printing. Default is 80 characters.
<i>remap</i>	String that specifies remapping character set. Default is null string.	<i>lineterm</i>	String containing line terminating character sequence. Does not affect IR printing. Default is carriage-return/line-feed.

Data Transfer

Transferring Data Between Two HP 48s. Before beginning transfer, set IR and binary transfer modes in I/O SETUP menu (  **SETUP**).

- On sender, go to directory where variables are located.
- On receiver, go to directory that will receive information.
- Position calculators less than 2 inches apart, with IR ports aligned (the ▲ symbols near the HP logo on top of the HP 48 cases should be lined up).

Local/Local Configuration

1. On receiver, do either of the following:
 - Press   **RECV** to store variable under sender name.
 - Press   **RECN** to select new name and store variable.
2. On sender, enter variable name and press   **SEND**.
3. Repeat steps 1 and 2 to send additional information.

Local/Server Configuration

1. On server, press   **SERVE** or  .
2. On local HP 48,
 - To send file to server, enter variable name and press   **SEND**.
 - To receive file from server, enter variable name and press   **KGET**.
3. Repeat step 2 to send additional information.
4. Press **FINIS** on local or press **ATTN** on server.

Transferring Data Between a Personal Computer and the HP 48. Before beginning the transfer,

- Connect computer and HP 48 via serial cable.
- Using the I/O SETUP menu (  SETUP), set wire transmission mode, ASCII or binary, transfer rate, parity, and checksum.
- Go to the sending or receiving HP 48 directory.
- Press   OPENI to open HP 48 serial port. (After transmission is complete, press   CLOSE to conserve battery power.)

Local/Local Configuration	Local/Server Configuration
1. Execute "receive" command: – For HP 48 receiver, press   RECV; or, to rename variable, enter variable name and press   RECN. – For computer receiver, execute "receive" command. 2. Execute "send" command: – For HP 48 sender, enter variable and press   SEND. – For computer sender, issue "send" <i>filespec</i> command. 3. Repeat steps 1 and 2 to transfer additional information.	1. Set server operation: – For HP 48 server, press   SERV. – For computer server, execute "server" command. 2. On local device, – To send file to server, execute "send" command. – To receive file from server, execute "get" command. 3. Repeat step 2 to send additional information. 4. Execute "finish" on local device to end session.

IOPAR Parameters {*baud parity receive-pacing transmit-pacing checksum translate-code*}

Parameter	Description	Parameter	Description
<i>baud</i>	Select 1200, 2400, 4800, or 9600. Default is 9600.	<i>transmit-pacing</i>	Not used by Kermit protocol. Non-zero value stops HP 48 transmission if XOFF signal is received. Default is 0.
<i>parity</i>	Select odd, even, mark, space, or no parity. Default is 0 (no parity).	<i>checksum</i>	Select the 1, 2, or 3 error detection option. Default is 3.
<i>receive-pacing</i>	Not used by Kermit protocol. Non-zero value sends XOFF signal if HP 48 buffer is full. Default is 0.	<i>translate-code</i>	Select translate code of 0, 1, 2, or 3. Default is 1.

EquationWriter Application and Plotting Example

Keystrokes	Description
$\leftarrow$ EQUATION X $\left[y^x \right]$ 2 $\rightarrow$ + 4X - 2 $\left[\text{ENTER} \right]$ $\leftarrow$ PLOT NEW P1 $\left[\text{ENTER} \right]$ PTYPE FUNC PLOT $\left[\text{NXT} \right]$ RESET $\leftarrow$ PREV AUTO FCN ROOT $\left[+ \right]$ $\rightarrow$ $\leftarrow$ ROOT $\left[+ \right]$ EXTR ATTN ATTN	Selects EquationWriter application. Enters expression $x^2 + 4x - 2$ onto the stack. Selects Plot application and prompts for name for new expression. Names expression $P1$; stores as current equation. Selects FUNCTION plot type. Selects PLOT menu, resets plot parameters to default state, and erases $P1CT$ (leaving blank $P1CT$ of default size). Draws plot, autoscaling the y-axis. Returns first root (.449489742783) and redisplayes menu keys. Moves cursor to far left and returns second root (-4.44948974278). Redisplays menu and calculates extremum ($-2, -6$). Exits application.

HP Solve Application Example

Keystrokes	Description
$\frac{1}{x}$ $\pm/\mp$ K $\times$ T $\leftarrow$ = $\rightarrow$ LN N $\div$ I ENTER $\leftarrow$ SOLVE NEW DK ENTER	Enters equation for radioactive decay. Selects HP Solve application; names equation; stores equation as current equation.
SOLVR 1.2 EEX $\pm/\mp$ 4 K	Selects SOLVR menu, enters value for K .
10.9 N	Enters value for N .
15.3 I	Enters value for I .
$\leftarrow$ T	Solves for T (2825.75032636).
8.5 N	Enters new value for N .
$\leftarrow$ K	Solves for K (2.08010827927E – 4).

Program Branch Structures

Conditional Structures	Typing Aid (PRG BRCH)
CASE <i>test-clause</i> ₁ THEN <i>true-clause</i> ₁ END ... <i>test-clause</i> _{<i>n</i>} THEN <i>true-clause</i> _{<i>n</i>} END END IF <i>test-clause</i> THEN <i>true-clause</i> END IF <i>test-clause</i> THEN <i>true-clause</i> ELSE <i>false-clause</i> END	↩ CASE ↩ IF ↪ IF
Definite Loop Structures	Typing Aid (PRG BRCH)
<i>start finish</i> FOR <i>counter</i> <i>loop-clause</i> NEXT <i>start finish</i> FOR <i>counter</i> <i>loop-clause</i> <i>increment</i> STEP <i>start finish</i> START <i>loop-clause</i> NEXT <i>start finish</i> START <i>loop-clause</i> <i>increment</i> STEP	↩ FOR ↪ FOR ↩ START ↪ START

Program Branch Structures (continued)

Indefinite Loop Structures	Typing Aid (PRG BRCH)
DO <i>loop-clause</i> UNTIL <i>test-clause</i> END WHILE <i>test-clause</i> REPEAT <i>loop-clause</i> END	↩ DO ↩ WHILE

Object Types and Delimiters

No.	Type	No.	Type	No.	Type
0	Real number	7	Local name	14	XLIB name
1	⟨Complex number⟩	8	«Program»	15	Directory (DIR ... END)
2	"String"	9	'Algebraic object'	16	Library
3	[Real array]	10	#Binary integer	17	Backup object
4	[Complex array]	11	Graphics object (GROB $n \times m$)	18	Built-in function
5	⟨List⟩	12	Tagged object (: tag: object)	19	Built-in command
6	Global name	13	Unit object (x_{unit})		

Command Reference Legend

Term	Description	Term	Description
[array]	Real or complex vector or matrix.	<i>PICT</i>	Current graphics object.
date	Date in form MM.DDYYYY or DD.MMYYYY.	point	Pixel $\langle \#n \#m \rangle$ or user-unit $\langle x,y \rangle$
$\langle dim \rangle$	List of 1 or 2 array dimensions.		coordinate.
'global'	Global name.	rc.p	Row, column, and plane (shift) of key.
grob	Graphics object.	$\langle r,\theta \rangle$	Complex number in polar form.
hms	Hours minutes seconds format (H.MMSSs).	"string"	Character <i>string</i> .
index	Real number specifying element in list or array.	'symb'	Expression, equation, or name treated as algebraic.
LID	Library ID number.	T/F	Test result: non-zero (true) or 0 (false).
$\langle list \rangle$	List of objects.	time	Time in form HH.MMSSs.

Term	Description	Term	Description
m or n	Positive integer real number.	[vector]	Real or complex vector.
[matrix]	Real or complex matrix.	x or y	Real number.
# n	Binary integer.	x_unit	Unit object.
'name'	Local or global name.	$\langle x,y \rangle$	Complex number in rectangular form.
obj	Any object.	z	Real or complex number.

For additional syntax possibilities for the following commands, please refer to the stack diagrams in the *Programmer's Reference Manual* for the HP 48 (HP part number 00048-90054).

Command Reference Example

Name	Description	Level 3	Level 2	Level 1	Command executed		
ROT	Rotates level 3 object into level 1.	obj ₁	obj ₂	obj ₃	→	obj ₂	obj ₃ obj ₁

Command Reference

Name	Description	In	Out
ABS	Absolute value.	z	$ z $
ACK	Acknowledges displayed past due alarm.		
ACKALL	Acknowledges all past due alarms.		
ACOS	Arc cosine.	z	$\text{acos}(z)$
ACOSH	Arc hyperbolic cosine.	z	$\text{acosh}(z)$
ALOG	Common (base 10) antilogarithm.	z	10^z
AND	Logical or binary AND.	$\#n_1 \ \#n_2$	$\#n_1 \text{ AND } \#n_2$
APPLY	Returns evaluated expressions as argument to unevaluated local name.	$'name' \ \{symb \dots\}$	$'name(symb \dots)'$

ARC	Draws arc in <i>PICT</i> with specified center point, radius, and beginning and ending angles.	$point \quad x_{radius} \quad x_{\theta 1} \quad x_{\theta 2} \rightarrow$
ARCHIVE	Makes backup copy of HOME directory.	$:n:name \rightarrow$
ARG	Returns polar angle θ .	$z \rightarrow x_{\theta}$
ARRY $\rightarrow$	Separates array.	$[array] \rightarrow z_1 \dots z_n \quad \{dim\}$
$\rightarrow$ ARRY	Combines numbers into array.	$z_1 \dots z_n \quad \{dim\} \rightarrow [array]$
ASIN	Arc sine.	$z \rightarrow \text{asin}(z)$
ASINH	Arc hyperbolic sine.	$z \rightarrow \text{asinh}(z)$
ASN	Makes a single user-key assignment.	$obj \quad rc.p \rightarrow$
ASR	1-bit arithmetic shift right.	$\#n_1 \rightarrow \#n_2$
ATAN	Arc tangent.	$z \rightarrow \text{atan}(z)$
ATANH	Arc hyperbolic tangent.	$z \rightarrow \text{atanh}(z)$

Name	Description	In	Out
ATTACH	Attaches library to current directory.		$LID \rightarrow$
AUTO	Scales y-axis.		
AXES	Sets coordinates of axes intersection.		$point \rightarrow$
BAR	Selects BAR plot.		
BARPLOT	Plots bar chart of ΣDAT data.		
BAUD	Sets transfer rate.		$n_{baud} \rightarrow$
BEEP	Sounds a beep.	$n_{frequency}$	$n_{duration} \rightarrow$
BESTFIT	Selects stat model that yields largest absolute value for correlation coefficient, and executes LR.		
BIN	Sets binary base.		

BINS	Sorts elements in independent variable column of ΣDAT into $n + 2$ bins.	$x_{\min} \ x_{width} \ n_{bins} \rightarrow [[b_1] \dots [b_n]] [b_{<} b_{>}]$
BLANK	Creates blank graphics object.	$\#n_{width} \ \#n_{height} \rightarrow grob$
BOX	Draws box with opposite corners defined by coordinates.	$point_1 \ point_2 \rightarrow$
BUFLEN	Returns number of characters in serial buffer.	$\rightarrow n_{length} \ T/F$
BYTES	Returns checksum and byte size of object.	$obj \rightarrow \#n_{checksum} \ x_{size}$
B→R	Binary-to-real conversion.	$\#n \rightarrow n$
CASE	Begins CASE ... THEN ... END ... END structure.	
CEIL	Returns next greater integer.	$x \rightarrow n$

Name	Description	In	Out
CENTR	Sets center of plot display at coordinates.		$point \rightarrow$
CF	Clears specified flag.		$\pm m \rightarrow$
%CH	Computes percent change.	$x_{initial}$	$y_{final} \rightarrow x_{\Delta\%}$
CHR	Converts character code to one-character string.		$n \rightarrow "string"$
CKSM	Sets I/O checksum type.		$n_{checksum} \rightarrow$
CLEAR	Clears stack.	$obj_1 \dots obj_n$	$\rightarrow$
CLKADJ	Adds clock ticks to time.		$\pm n_{ticks} \rightarrow$
CLLCD	Blanks stack display.		

CLOSEIO	Closes serial port.
CLUSR	Same as CLVAR.
CLVAR	Purges all user variables in current directory.
CL Σ	Purges ΣDAT .
CNRM	Computes column norm of array.
COLCT	Collects like terms in expression.
COL Σ	Selects independent and dependent statistics columns.
COMB	Returns combinations of n objects taken m at a time.
CON	Creates constant array.
CONIC	Selects conic plot.

$[array]$	$\rightarrow$	$x_{column - norm}$
' $symb_1$ '	$\rightarrow$	' $symb_2$ '
n_{indep} n_{depen}	$\rightarrow$	
n m	$\rightarrow$	$C_{n,m}$
$\{dim\}$ z	$\rightarrow$	$[array]$

Name	Description	In	Out
CONJ	Returns complex conjugate.		$[array] \rightarrow [array^*]$
CONT	Continues a halted program.		
CONVERT	Performs unit conversion.	$x_1_unit_1$	$x_2_unit_2 \rightarrow x_3_unit_2$
CORR	Calculates correlation coefficient.		$\rightarrow x_{correlation}$
COS	Cosine.		$z \rightarrow \cos(z)$
COSH	Hyperbolic cosine.		$z \rightarrow \cosh(z)$
COV	Calculates covariance.		$\rightarrow x_{covariance}$
CR	Printer carriage return/line feed.		
CRDIR	Creates directory.		$'global' \rightarrow$
CROSS	Cross product of vectors.	$[vector_1]$	$[vector_2] \rightarrow [vector_1 \times vector_2]$
C→PX	User-unit to pixel coordinate conversion.		$(x,y) \rightarrow \{\#n \#m\}$

C→R	Complex to real number conversion.	$(x,y) \rightarrow x \ y$
DATE	Returns date.	$\rightarrow date$
DATE +	Returns new date from specified date and number of days.	$date_1 \ n_{days} \rightarrow date_2$
→DATE	Sets date.	$date \rightarrow$
DDAYS	Number of days between two dates.	$date_1 \ date_2 \rightarrow n_{\Delta days}$
DEC	Sets decimal base.	
DECR	Decrements and returns value of specified variable.	$'name' \rightarrow x$
DEFINE	Creates variable or user-defined function.	$'symb' \rightarrow$
DEG	Sets Degrees mode.	
DELALARM	Deletes alarm from system alarm list.	$n_{alarm} \rightarrow$
DELAY	Sets delay between printed lines.	$x \rightarrow$

Name	Description	In	Out
DELKEYS	Clears user-key assignments.	$rc.p$	$\rightarrow$
DEPND	Specifies name of dependent plot variable.	' $name$ '	$\rightarrow$
DEPTH	Returns number of objects on stack.		$\rightarrow n$
DET	Determinant of a matrix.	$[matrix]$	$\rightarrow x_{determinant}$
DETACH	Detaches library from current directory.	LID	$\rightarrow$
DISP	Displays object in specified display line.	obj n_{line}	$\rightarrow$
DO	Begins DO ... UNTIL ... END structure.		
DOERR	Aborts program execution; displays error message.	" $string$ "	$\rightarrow$

DOT	Dot product of two arrays.	$[array_1] \ [array_2] \rightarrow x$
DRAW	Plots contents of EQ .	
DRAX	Draws axes.	
DROP	Drops level 1 object from stack.	$obj \rightarrow$
DROPN	Drops n objects from stack.	$obj_1 \dots obj_n \ n \rightarrow$
DROP2	Drops two objects from stack.	$obj_1 \ obj_2 \rightarrow$
DTAG	Removes all tags from object.	$:tag:obj \rightarrow obj$
DUP	Duplicates object in level 1.	$obj \rightarrow obj \ obj$
DUPN	Duplicates n objects on stack.	$obj_1 \dots obj_n \ n \rightarrow obj_1 \dots obj_n \ obj_1 \dots obj_n$
DUP2	Duplicates objects in levels 1 and 2.	$obj_1 \ obj_2 \rightarrow obj_1 \ obj_2 \ obj_1 \ obj_2$
D→R	Degrees-to-radians conversion.	$x_{degrees} \rightarrow x_{radians}$
e	Symbolic constant e (2.71828182846).	
ELSE	Begins ELSE clause.	

Name	Description	In	Out
END	Ends program structures.		
ENG	Sets display mode to Engineering.		
EQ→	Separates equation into left and right sides.	n_{digits} → ' $symb_1 = symb_2$ '	→ ' $symb_1$ ' ' $symb_2$ '
ERASE	Erases <i>PICT</i> .		
ERRM	Returns last error message.		→ "error message"
ERRN	Returns last error number.		→ # n
ERR0	Clears last error message and number.		
EVAL	Evaluates object.	obj	→ ...
EXP	e raised to power of object in level 1.	z	→ e^z
EXPAN	Expands algebraic object.	' $symb_1$ '	→ ' $symb_2$ '

EXPFIT	Sets curve-fitting model to exponential.
EXPM	Natural exponential minus 1.
FC?	Tests if specified flag is clear.
FC?C	Tests if flag is clear, then clears it.
FINDALARM	Returns first alarm due after specified time or first past due alarm.
FINISH	Terminates Kermit server mode.
FIX	Selects FIX display mode.
FLOOR	Next smaller integer.
FOR	Begins FOR ... NEXT or FOR ... STEP structure.
FP	Returns fractional part of a number.

$$x \rightarrow e^x - 1$$

$$\pm m_{flag} \rightarrow T/F$$

$$\pm m_{flag} \rightarrow T/F$$

$$date \rightarrow n_{alarm}$$

$$n_{digits} \rightarrow$$

$$x \rightarrow \pm n$$

$$x_{start} \quad x_{stop} \rightarrow$$

$$x \rightarrow FP(x)$$

Name	Description	In	Out
FREE	Frees merged memory.	$\{name/LID \dots\} \quad n_{port} \rightarrow$	
FREEZE	Freezes up to three display areas.	$n \rightarrow$	
FS?	Tests if specified flag is set.	$\pm m_{flag} \rightarrow$	T/F
FS?C	Tests if flag is set, then clears it.	$\pm m_{flag} \rightarrow$	T/F
FUNCTION	Selects FUNCTION plot.		
GET	Gets element from array or list.	$[array] \quad index \rightarrow$	z
GETI	Gets element from array or list and increments index.	$[array] \quad index_1 \rightarrow$	$[array] \quad index_2 \quad z$
GOR	Superposes graphics object ($grob_2$) onto target graphics object with upper left corner at point.	$grob_1 \quad point \quad grob_2 \rightarrow$	$grob_3$

GRAD	Selects Grads mode.
GRAPH	Activates graphics environment.
→GROB	Converts object into graphics object.
GXOR	Uses exclusive OR to superpose graphics object ($grob_2$) into target graphics object with upper left corner at point.
*H	Multiplies vertical plot scale by factor.
HALT	Halts program execution.
HEX	Sets hexadecimal base.
HISTPLOT	Draws histogram of data in ΣDAT .
HISTOGRAM	Selects HISTOGRAM plot.
HMS+	Adds in hms format.

$$obj \quad n_{font} \rightarrow grob$$

$$grob_1 \quad point \quad grob_2 \rightarrow grob_3$$

$$x_{factor} \rightarrow$$

$$hms_1 \quad hms_2 \rightarrow hms_3$$

Name	Description	In	Out
HMS–	Subtracts in hms format.	hms_1	$hms_2 \rightarrow hms_3$
HMS→	Converts hms to decimal format.		$hms \rightarrow x$
→HMS	Converts decimal number to hms format.		$x \rightarrow hms$
HOME	Selects HOME directory.		
i	Symbolic constant i ($\sqrt{-1}$).		
IDN	Creates identity matrix of specific size.		$n \rightarrow [matrix]$
IF	Begins IF ... THEN ... END or IF ... THEN ... ELSE ... END structure.		
IFERR	Begins IFERR ... THEN ... END or IFERR ... THEN ... ELSE ... END structure.		

IFT	IF / THEN command.	T/F	obj_{true}	$\rightarrow$	
IFTE	IF / THEN / ELSE function.	T/F	obj_{true}	obj_{false}	$\rightarrow$
IM	Returns Imaginary part.		(x,y)	$\rightarrow$	y
INCR	Increments and returns value of specified variable.		$'name'$	$\rightarrow$	x
INDEP	Specifies independent variable in plot.		$'name'$	$\rightarrow$	
INPUT	Suspends program, displays message, and waits for entry.	$"prompt"$	$"command\ line"$	$\rightarrow$	$"result"$
INV	Reciprocal.		z	$\rightarrow$	$1/z$
IP	Integer part of real number.		x	$\rightarrow$	$IP(x)$
ISOL	Isolates variable on one side of equation.	$'symb_1'$	$'name'$	$\rightarrow$	$'name = symb_2'$
KERRM	Returns last Kermit error message.			$\rightarrow$	$"error\ message"$

Name	Description	In	Out
KEY	Returns number indicating last key pressed.		→ 0 or <i>rc.p</i> 1
KGET	Gets file from another device.	' <i>name</i> '	→
KILL	Aborts all suspended programs.		
LABEL	Labels axes.		
LAST	Same as LASTARG.		
LASTARG	Returns last arguments to stack.		→ <i>obj_{arg1} . . . obj_{arg-n}</i>
LCD→	Puts displayed image on stack.		→ <i>grob</i>
→LCD	Displays graphics object.	<i>grob</i>	→
LIBS	Lists libraries attached to current directory.		→ { <i>title LID port . . .</i> }

LINE	Draws line between two coordinates.	$point_1 \quad point_2 \rightarrow$
Σ LINE	Returns best-fit line for data in ΣDAT .	$\rightarrow 'symbol'$
LINFIT	Sets curve-fitting model to linear.	
$\rightarrow$ LIST	Combines objects into list.	$obj_1 \dots obj_n \quad n \rightarrow \{obj_1 \dots obj_n\}$
LIST $\rightarrow$	Separates list.	$\{obj_1 \dots obj_n\} \rightarrow obj_1 \dots obj_n \quad n$
LN	Natural (base e) logarithm.	$z \rightarrow \ln(z)$
LNP1	Natural logarithm of $(x + 1)$.	$x \rightarrow \ln(x + 1)$
LOG	Common (base 10) logarithm.	$z \rightarrow \log(z)$
LOGFIT	Sets curve-fitting model to logarithmic.	
LR	Calculates linear regression.	$\rightarrow x_{intercept} \quad x_{slope}$
MANT	Mantissa of number.	$x \rightarrow y$

Name	Description	In	Out
↑MATCH	Match-and-replace, beginning with subexpressions.	' <i>symb</i> ' { ' <i>pattern₁</i> ' ' <i>pattern₂</i> ' }	→ ' <i>result</i> ' <i>T/F</i>
↓MATCH	Match-and-replace, beginning with top-level expression.	' <i>symb</i> ' { ' <i>pattern₁</i> ' ' <i>pattern₂</i> ' }	→ ' <i>result</i> ' <i>T/F</i>
MAX	Maximum of two real numbers.	<i>x y</i>	→ max(<i>x,y</i>)
MAXR	Maximum real number (9.999999999999E499).		
MAXΣ	Finds maximum column values in Σ <i>DAT</i> statistics matrix.		→ [<i>x₁ ... x_n</i>]
MEAN	Calculates statistical mean of Σ <i>DAT</i> data.		→ [<i>x₁ ... x_n</i>]

MEM	Bytes of available memory.	$\rightarrow x$
MENU	Creates custom menu.	$\{obj \dots\} \rightarrow$
MERGE	Merges RAM with main memory.	$n_{port} \rightarrow$
MIN	Minimum of two real numbers.	$x \ y \rightarrow \min(x,y)$
MINR	Minimum machine-representable real number (1.E-499).	
MIN Σ	Finds minimum column values in ΣDAT statistics matrix.	$\rightarrow [x_1 \dots x_n]$
MOD	Modulo.	$x \ y \rightarrow x \bmod y$
N Σ	Returns number of rows in ΣDAT .	$\rightarrow n$
NEG	Negates level-1 argument.	$z \rightarrow -z$
NEWOBJ	Creates new copy of object.	$obj \rightarrow obj$

Name	Description	In	Out
NEXT	Ends FOR ... NEXT or START ... NEXT structure.		
NOT	Logical or binary NOT.	$\#n_1$	$\rightarrow$ NOT $\#n_1$
NUM	Returns character code of leading character.	"string"	$\rightarrow$ n
$\rightarrow$ NUM	Evaluates object in numerical mode.	obj	$\rightarrow$
OBJ $\rightarrow$	Returns object components to stack (combines ARRY $\rightarrow$, LIST $\rightarrow$ STR $\rightarrow$).	$\{obj_1 \dots obj_n\}$	$\rightarrow$ $obj_1 \dots obj_n$ n
OCT	Sets octal base.		
OFF	Turns calculator off.		
OLDPRT	Remaps printer character set to match HP 82240A printer.		

OPENIO	Opens IR or wired port.
OR	Logical or binary OR.
ORDER	Rearranges directory.
OVER	Duplicates level 2 object.
PARAMETRIC	Selects PARAMETRIC plot.
PARITY	Sets parity.
PATH	Returns path to current directory.
PDIM	Changes size of <i>PICT</i> .
PERM	Permutations of n objects taken m at a time.
PGDIR	Purges specified directory.

$\#n_1 \ \#n_2 \rightarrow \#n_1 \text{ OR } \#n_2$

$\{name \dots\} \rightarrow$

$obj_1 \ obj_2 \rightarrow obj_1 \ obj_2 \ obj_1$

$n \rightarrow$

$\rightarrow \{\text{HOME } name \dots\}$

$\#n_{width} \ \#n_{height} \rightarrow$

$n \ m \rightarrow P_{n,m}$

$'global' \rightarrow$

Name	Description	In	Out
PICK	Copies obj_1 from level $n + 1$ to level 1.	$obj_1 \dots obj_n \quad n$	$\rightarrow \quad obj_1 \dots obj_n \quad obj_1$
PICT	Returns <i>PICT</i> to level 1.		$\rightarrow \quad PICT$
PIXOFF	Turns off pixel in <i>PICT</i> .	$point$	$\rightarrow$
PIXON	Turns on pixel in <i>PICT</i> .	$point$	$\rightarrow$
PIX?	Tests pixel in <i>PICT</i> .	$point$	$\rightarrow \quad T/F$
PKT	Sends Kermit packet.	"contents" "type"	$\rightarrow \quad$ "results"
PMAX	Sets upper right plot coordinates.	(x, y)	$\rightarrow$
PMIN	Sets lower left plot coordinates.	(x, y)	$\rightarrow$
POLAR	Selects POLAR plot.		
POS	Returns position of object in list or string.	$\{obj \dots \} \quad obj_{to-find}$	$\rightarrow \quad n_{position}$

PREDV	Predicted value.	$x \rightarrow y$
PREDX	Returns predicted independent variable value.	$y \rightarrow x$
PREDY	Returns predicted dependent variable value.	$x \rightarrow y$
PRLCD	Prints display.	
PROMPT	Displays prompt and halts program.	"prompt" $\rightarrow$
PRST	Prints all objects on stack.	
PRSTC	Prints stack in compact form.	
PRVAR	Prints name and contents of variables.	' name ' $\rightarrow$
PR1	Prints object in level 1.	<i>obj</i> $\rightarrow$ <i>obj</i>
PURGE	Purges one or more variables.	' name ' $\rightarrow$
PUT	Replaces element in array or list.	$[array_1] \text{ index } z \rightarrow [array_2]$
PUTI	Same as PUT, plus increments index.	$[array_1] \text{ index}_1 z \rightarrow [array_2] \text{ index}_2$

Name	Description	In	Out
PVARS	Returns list of port backup objects and libraries.		→ { <i>name/LID</i> ... }
PVIEW	Displays <i>P/CT</i> with specified pixel at upper left corner.	<i>point</i>	→
PWRFIT	Selects power curve-fitting model.		
PX→C	Pixel to user-unit coordinate conversion.	{# <i>n</i> # <i>m</i> }	→ (<i>x</i> , <i>y</i>)
→Q	Converts number to fractional equivalent.	<i>x</i>	→ ' <i>symb</i> '
QUAD	Finds roots of quadratic polynomial.	' <i>symb</i> ₁ '	' <i>global</i> ' → ' <i>symb</i> ₂ '
QUOTE	Returns argument expression unevaluated.	' <i>symb</i> '	→ ' <i>symb</i> '

$\rightarrow Q\pi$	Computes and compares quotients of number and number/ π .	$x \rightarrow 'symb'$
RAD	Selects Radians mode.	
RAND	Returns random number $0 < x < 1$.	$\rightarrow x$
RATIO	Prefix form of / used by EquationWriter application.	$z_1 \ z_2 \rightarrow z_1/z_2$
RCEQ	Returns object in EQ to level 1.	$\rightarrow obj$
RCL	Recalls contents of variable to stack.	$'name' \rightarrow obj$
RCLALARM	Recalls alarm from system alarm list.	$n_{alarm} \rightarrow \{alarm\}$
RCLF	Returns state of system and user flags.	$\rightarrow \{\#system \ #user\}$
RCLKEYS	Recalls user-key assignments.	$\rightarrow \{obj_1 \ rc.p_1 \dots obj_n \ rc.p_n\}$
RCLMENU	Returns number and page of active menu.	$\rightarrow x$

Name	Description	In	Out
RCWS	Recalls binary integer wordsize.		$\rightarrow n$
RCL Σ	Recalls current statistical matrix in ΣDAT .		$\rightarrow [matrix]$
RDM	Redimensions an array.	$[array_1] \{dim\}$	$\rightarrow [array_2]$
RDZ	Sets random number seed.	x	$\rightarrow$
RE	Returns real part of complex number or array.	(x,y)	$\rightarrow x$
RECN	Receives file from remote Kermit, saved in a stack specified variable.	$'name'$	$\rightarrow$
RECV	Receives file from remote Kermit, saved in sender specified variable.		

REPEAT Part of WHILE ... REPEAT ... END.

REPL Replaces portion of list, string,
graphics object, or PICT in level 3
with like object.

RESTORE Replaces HOME directory with
archived version.

RL Rotates left one bit.

RLB Rotates left one byte.

RND Rounds real or complex number or
array as specified in level 1.

RNRM Calculates row norm of array.

ROLL Moves obj_1 from level $n + 1$ to level 1.

ROLLD Moves object in level 2 to level n .

$obj_1 \quad obj_{position} \quad obj_2 \rightarrow obj_3$

$:n:name \rightarrow$

$\#n_1 \rightarrow \#n_2$

$\#n_1 \rightarrow \#n_2$

$x_1 \pm n \rightarrow x_2$

$[array] \rightarrow x_{row-norm}$

$obj_1 \dots obj_n \quad n \rightarrow obj_1 \dots obj_n \quad obj_1$

$obj_1 \dots obj_n \quad n \rightarrow obj_n \quad obj_1 \dots obj_{n-1}$

Name	Description	In	Out
ROOT	Solves for unknown variable in equation.	' symb ' ' global ' z_{guess}	$\rightarrow x_{root}$
ROT	Rotates level 3 object into level 1.	obj_1 obj_2 obj_3	$\rightarrow obj_2$ obj_3 obj_1
RR	Rotates right one bit.		$\#n_1 \rightarrow \#n_2$
RRB	Rotates right one byte.		$\#n_1 \rightarrow \#n_2$
RSD	Computes correction to solution of system of equations.	[array B] [matrix A] [array Z]	$\rightarrow [array B - AZ]$
R→B	Real-to-binary conversion.		$n \rightarrow \#n$
R→C	Real-to-complex conversion.	x y	$\rightarrow (x,y)$
R→D	Radians-to-degrees conversion.	$x_{radians}$	$\rightarrow x_{degrees}$
SAME	Tests two objects for equality.	obj_1 obj_2	$\rightarrow T/F$
SBRK	Sends serial break.		

SCALE	Sets scale of PLOT axes.	$x_{x-scale}$ $y_{y-scale}$	→
SCATRLOT	Draws scatter plot of ΣDAT data.		
SCATTER	Selects SCATTER plot.		
SCI	Selects Scientific display mode.	n_{digits}	→
SCONJ	Conjugates contents of variable.	'name'	→
SDEV	Calculates standard deviation of ΣDAT .		→ $[x_1 \dots x_n]$
SEND	Sends object to another Kermit device.	'name'	→
SERVER	Selects Kermit Server mode.		
SF	Sets specified flag.	$\pm m$	→
SHOW	Makes name references explicit.	'symb ₁ ' 'global'	→ 'symb ₂ '

Name	Description	In	Out
SIGN	Returns sign of number.	z_1	$\rightarrow z_2$
SIN	Sine.	z	$\rightarrow \sin(z)$
SINH	Hyperbolic sine.	z	$\rightarrow \sinh(z)$
SINV	Replaces variable with its inverse.	'name'	$\rightarrow$
SIZE	Returns size of object.	obj	$\rightarrow n$ or {dim}
SL	Shifts left one bit.	$\#n_1$	$\rightarrow \#n_2$
SLB	Shifts left one byte.	$\#n_1$	$\rightarrow \#n_2$
SNEG	Negates contents of variable.	'name'	$\rightarrow$

SQ	Squares object in level 1.	$x \rightarrow x^2$
SR	Shifts right one bit.	$\#n_1 \rightarrow \#n_2$
SRB	Shifts right one byte.	$\#n_1 \rightarrow \#n_2$
SRECV	Reads n characters from I/O port.	$n \rightarrow \text{"string"} \quad T/F$
START	Begins START ... NEXT or START ... STEP structure.	$x_{start} \quad y_{stop} \rightarrow$
STD	Selects Standard display mode.	
STEP	Ends definite loop.	$x_{step} \rightarrow$
STEQ	Stores equation in EQ .	$'symb' \rightarrow$
STIME	Sets serial transmit/receive timeout period.	$x \rightarrow$
STO	Stores object in variable.	$obj \quad 'name' \rightarrow$

Name	Description	In	Out
STOALARM	Stores alarm in system alarm list.	$time$	$\rightarrow n_{alarm}$
STOF	Stores system and user flags.	$\{ \#system \#user \}$	$\rightarrow$
STOKEYS	Makes multiple user-key assignments.	$\{ obj_1 \ rc.p_1 \dots obj_n \ rc.p_n \}$	$\rightarrow$
STO+	Storage arithmetic — add.	$z \ 'name'$	$\rightarrow$
STO−	Storage arithmetic — subtract.	$z \ 'name'$	$\rightarrow$
STO*	Storage arithmetic — multiply.	$z \ 'name'$	$\rightarrow$
STO/	Storage arithmetic — divide.	$z \ 'name'$	$\rightarrow$
STO Σ	Stores statistics matrix in ΣDAT .	$[matrix]$	$\rightarrow$
STR→	Executes objects defined by string.	"string"	$\rightarrow$

→STR	Converts object into string.	$obj \rightarrow "obj"$
STWS	Sets binary integer wordsize.	$n \rightarrow$
SUB	Extracts portion of list, string, graphics object, or <i>PICT</i> .	$obj_1 \ n_{start} \ n_{end} \rightarrow obj_2$
SWAP	Exchanges objects in levels 1 and 2.	$obj_1 \ obj_2 \rightarrow obj_2 \ obj_1$
SYSEVAL	Evaluates system object.	$\#n_{address} \rightarrow$
%T	Returns percent of total.	$x_{total} \ y_{part} \rightarrow x\%T$
→TAG	Combines levels 1 and 2 objects into tagged object.	$obj \ "tag" \rightarrow :tag:obj$
TAN	Tangent.	$z \rightarrow \tan(z)$
TANH	Hyperbolic tangent.	$z \rightarrow \tanh(z)$
TAYLR	Calculates Taylor polynomial.	$'symb_1' \ 'global' \ n_{order} \rightarrow 'symb_2'$

Name	Description	In	Out
TEXT	Selects stack display.		
THEN	Begins THEN clause.		
TICKS	Returns time as binary integer in clock tick units.		→ <i>#ticks</i>
TIME	Returns current time as real number.		→ <i>time</i>
→TIME	Sets specified time.	<i>time</i>	→
TLINE	Toggles pixels on straight line.	<i>point</i> ₁ <i>point</i> ₂	→
TMENU	Creates temporary custom menu.	{ <i>obj</i> ₁ ... <i>obj</i> _{<i>n</i>} }	→
TOT	Sums each column of matrix in ΣDAT .		→ [<i>x</i> ₁ ... <i>x</i> _{<i>n</i>}]
TRANSIO	Sets character-translation mode.	<i>n</i>	→
TRN	Transposes and conjugates matrix.	[<i>matrix</i> ₁]	→ [<i>matrix</i> ₂]

TRNC	Truncates level 2 object as specified in level 1.	$x_1 \pm n \rightarrow x_2$
TRUTH	Selects TRUTH plot.	
TSTR	Converts date/time to string form.	<i>date time</i> $\rightarrow$ "date time"
TVAR\$	Lists variables containing specified object type.	$n_{type} \rightarrow \{global \dots\}$
TYPE	Returns type-number of object.	<i>obj</i> $\rightarrow n_{type}$
UBASE	Converts unit object to SI base units.	$x_units \rightarrow y_base-units$
UFACT	Factors compound unit.	$x_1_units_1 \ x_2_units_2 \rightarrow x_3_units_2 * units_3$
$\rightarrow$ UNIT	Combines levels 1 and 2 objects to create unit object.	$x \ y_units \rightarrow x_units$
UNTIL	Begins UNTIL clause.	
UPDIR	Makes parent directory current directory.	

Name	Description	In	Out
UTPC	Upper-tail chi-square distribution.	$n \ x$	$\rightarrow \text{utpc}(n, x)$
UTPF	Upper-tail Snedecor's F distribution.	$n_1 \ n_2 \ x$	$\rightarrow \text{utpf}(n_1, n_2, x)$
UTPN	Upper-tail normal distribution.	$x_{mean} \ x_{var} \ x$	$\rightarrow \text{utpn}(x_{mean}, x_{var}, x)$
UTPT	Upper-tail Student's t distribution.	$n \ x$	$\rightarrow \text{utpt}(n, x)$
UVAL	Returns number portion of unit object.	x_unit	$\rightarrow x$
VARS	Returns list of variables in current directory.		$\rightarrow \{global \dots\}$
VTYPER	Returns type number of stored object.	$'name'$	$\rightarrow n$
→V2	Combines two real numbers into 2-D vector or complex number.	$x_1 \ x_2$	$\rightarrow [x_1 \ x_2]$
→V3	Combines three real numbers into 3-D vector.	$x_1 \ x_2 \ x_3$	$\rightarrow [x_1 \ x_2 \ x_3]$

V→	Separates vector into components.	$[x_1 \ x_2] \rightarrow x_1 \ x_2$
*W	Multiplies horizontal plot scale by factor.	$x_{factor} \rightarrow$
WAIT	Pauses program execution for n seconds.	$n \rightarrow$
WHILE	Begins WHILE ... REPEAT ... END structure.	
ΣX	Sum of data in independent ΣDAT column.	$\rightarrow \Sigma x_i$
ΣX^2	Sum of squares of data in independent ΣDAT column.	$\rightarrow \Sigma x_i^2$
XCOL	Specifies ΣDAT column as independent variable.	$n_{indep} \rightarrow$
XMIT	Sends string without Kermit protocol.	"string" $\rightarrow$ 1 or "unsent" 0

Name	Description	In	Out
XOR	Logical or binary exclusive OR.	$\#n_1 \quad \#n_2$	$\rightarrow \#n_1 \text{ XOR } \#n_2$
XPON	Returns exponent of number.	x	$\rightarrow n$
XRNG	Specifies x-axis plotting range.	$x_{\min} \quad x_{\max}$	$\rightarrow$
XROOT	Returns x th root of y .	$y \quad x$	$\rightarrow \sqrt[x]{y}$
$\Sigma X*Y$	Sums products of corresponding data in ΣDAT columns.		$\rightarrow \Sigma x_i y_i$
ΣY	Sums data in dependent ΣDAT column.		$\rightarrow \Sigma y_i$
ΣY^2	Sum of squares of data in dependent ΣDAT column.		$\rightarrow \Sigma y_i^2$
YCOL	Specifies a ΣDAT column as the dependent variable.	n	$\rightarrow$

YRNG	Specifies y-axis plotting range.	$y_{\min}$ $y_{\max}$ $\rightarrow$
+	Adds two objects.	z_1 z_2 $\rightarrow$ $z_1 + z_2$
-	Subtracts two objects.	z_1 z_2 $\rightarrow$ $z_1 - z_2$
*	Multiplies two objects.	z_1 z_2 $\rightarrow$ $z_1 z_2$
/	Divides two objects.	z_1 z_2 $\rightarrow$ z_1/z_2
%	Percent.	x_{whole} y_{part} $\rightarrow$ $x\%$
^	Raises number to specified power.	z_1 z_2 $\rightarrow$ $z_1^{z_2}$
√	Square root.	z $\rightarrow$ $\sqrt{z}$
!	Factorial.	n or x $\rightarrow$ $n!$ or $\Gamma(x+1)$
∂	Derivative.	' symb_1 ' 'global' $\rightarrow$ ' symb_2 '

Name	Description	In		Out	
$\int$	Integral.	x_{lower}	x_{upper}	'integrand'	'name' $\rightarrow x_{integral}$
<	"Less-than" comparison.			x	$y \rightarrow T/F$
$\leq$	"Less-than-or-equal to" comparison.			x	$y \rightarrow T/F$
=	Equals.			z_1	$z_2 \rightarrow 'z_1 = z_2'$
= =	Equality comparison.			x	$y \rightarrow T/F$
$\neq$	"Not-equal" comparison.			x	$y \rightarrow T/F$
>	"Greater-than" comparison.			x	$y \rightarrow T/F$
$\geq$	"Greater-than-or-equal-to" comparison.			x	$y \rightarrow T/F$

π	Symbolic constant pi.		
Σ	Summation.	'name' $x_{lower} x_{upper}$ 'symb' $\rightarrow$ x_{sum}	
$\Sigma+$	Adds data point to ΣDAT matrix.	$[x_1 \dots x_n]$ $\rightarrow$	
$\Sigma-$	Removes data point from ΣDAT matrix.	$\rightarrow [x_1 \dots x_n]$	
$\rightarrow$	Creates local variables.		
	Appends local name and value to expression.	'symb ₁ ' {name 'symb' ... } $\rightarrow$ 'symb ₂ '	

Notice

This booklet and any examples contained herein are provided “as is” and are subject to change without notice. **Hewlett-Packard Company makes no warranty of any kind with regard to this booklet, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.** Hewlett-Packard Co. shall not be liable for any errors or for incidental or consequential damages in connection with the furnishing, performance, or use of this booklet or the examples contained herein.

**Corvallis Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.**

Alpha Mode

Left- and right- shifted keys

a	α	b	β	c	Δ	d	δ	e	ϵ	f	θ
g	τ	h	η	i	∞	j		k	$\uparrow$	l	λ
m	'	n	μ	o	Ω	p	$\leftarrow$	q	$\downarrow$	r	ρ
s	σ	t	τ	u	%	v	$\sim$	w	ω	x	$\bar{x}$
&		@	y	$\pm$	z	π	!	i	?	ι	
LC	INS	\	/	^	~	..	etc. ()	#			
	\$	¢	£	¥	¤	□	[]	-			
	=	≠	<	>	≤	≥	<<>>	■ ■			
CONT	OFF	=	→	,	↙	π	∠	{ }	::		



HEWLETT
PACKARD