



# The HP 48 Pocket Book

James Donnelly



# The HP 48 Pocket Book

**James Donnelly**

Copyright © Armstrong Publishing Company 1993

All rights reserved. No part of this book may be reproduced, transmitted, or stored in a retrieval system in any form or by any process, electronic, mechanical, photocopying or means yet to be invented, without specific prior written permission of the author.

First Edition

First Printing, June 1993

ISBN 1-879828-05-7

Armstrong Publishing Company  
3135 NW Ashwood Drive  
Corvallis, OR 97330 USA

For information on additional HP 48 publications, like *The HP 48 Handbook*, contact your calculator dealer or write to Armstrong Publishing Company.

# Flags



Flag	Description	Clear	Set	Default
Symbolic Math				
−1	Principal Solution	General solutions	Principal solutions	Clear
−2	Symbolic Constants	Symbolic form	Numeric form	Clear
−3	Numeric Results	Symbolic results	Numeric results	Clear
−4	Not used			
Binary Integer Math				
−5	Binary integer wordsize $n + 1$ : $0 \leq n \leq 63$			64
−10	Flag −10 is the most significant bit			
	Base	−11	−12	DEC
−11	DEC	Clear	Clear	
and	BIN	Clear	Set	
−12	OCT	Set	Clear	
	HEX	Set	Set	
−13	Not used			
Finance				
−14	TVM Payment Mode	End of Period	Beginning of Period	End



Coordinate System		-15	-16	Rect.
-15 and -16	Rectangular Cylindrical Polar Spherical Polar	Clear Clear Set	Clear Set Set	
Trigonometric Mode		-17	-18	Degrees
-17 and -18	Degrees Radians Grads	Clear Set Clear	Clear Clear Set	
Math Exception				
-19 -20	Vector/complex Underflow Exception	Vector Return 0, set -23 or -24	Complex Error	Vector 0
-21	Overflow Exception	Return ±MAXR, set -25	Error	±MAXR
-22	Infinite Result	Error	Return ±MAXR, set -26	Error
-23	Pos. Underflow Ind.	No Exception	Exception	No
-24	Neg. Underflow Ind.	No Exception	Exception	No
-25	Overflow Indicator	No Exception	Exception	No
-26	Infinite Result Ind.	No Exception	Exception	No
-27	Symbolic Decompilation	'X+Y*i'→'(X,Y)'	'X+Y*i'→'X+Y*i'	'(X,Y)'

<b>Plotting and Graphics</b>				
-28	Plotting Multiple Functions	Plotted serially	Plotted simultaneously	Serial
-29	Trace mode	Trace off	Trace on	Off
-30	Not used			
-31	Curve Filling	Filling enabled	Filling disabled	Enabled
-32	Graphics Cursor	Visible light bkgnd	Visible dark bkgnd	Light
<b>I/O and Printing</b>				
-33	I/O Device	Wire	IR	Wire
-34	Printing Device	IR	Wire	IR
-35	I/O Data Format	ASCII	Binary	ASCII
-36	RECV Overwrite	New variable	Overwrite	New
-37	Double-spaced Print	Single	Double	Single
-38	Linefeed	Inserts LF	Suppresses LF	Inserts
-39	Kermit Messages	Msg displayed	Msg suppressed	Displayed

Time Management				
–40	Clock Display	TIME menu only	Always	TIME menu
–41	Clock Format	12-hour	24-hour	12 hour
–42	Date Format	MM/DD/YY	DD.MM.YY	MM/DD/YY
–43	Rpt. Alarm Resched.	Rescheduled	Not Rescheduled	Rescheduled
–44	Acknowledged Alarms	Deleted	Saved	Deleted
<b>Notes:</b> If flag –43 is set, unacknowledged repeat alarms are <i>not</i> rescheduled. If flag –44 is set, acknowledged alarms are saved in the alarm catalog.				
Display Format				
–45→	Set the number of digits in Fix, Scientific, and			0
–48	Engineering modes			
Number Display Format		–49	–50	STD
–49	STD	Clear	Clear	
and	FIX	Clear	Set	
–50	SCI	Set	Clear	
	ENG	Set	Set	
–51	Fraction Mark	Decimal	Comma	Decimal
–52	Single Line Display	Multi–line	Single–line	Multi
–53	Precedence	() suppressed	() displayed	Suppressed

Miscellaneous				
-54	Singular Value Underflow	Replaces “tiny” pivots with 0	No replacement	Replaces
-55	Last Arguments	Saved	Not Saved	Saved
-56	Beep	On	Off	On
-57	Alarm Beep	On	Off	On
-58	Verbose Messages	On	Off	On
-59	Fast Catalog Display	Off	On	Off
-60	Alpha Key Action	Twice to lock	Once to lock	Twice
-61	USR Key Action	Twice to lock	Once to lock	Twice
-62	User Mode	Not Active	Active	Clear
-63	Vectored Enter	Off	On	Off
-64	Set by GETI or PUTI when their element indices wrap around			
Equation Library				
60	Units Type	SI units	English units	SI
61	Units Usage	Units used	Units not used	Used
Multiple Equation Solver				
63	Variable State Change	 recalls variable	 toggles variable state	Recalls

# Operator Precedence

Operator precedence controls the order in which calculations take place within an algebraic expression. Functions with the highest precedence (1) are evaluated before those with the lowest precedence (11). The evaluation order is left-to-right for operators having the same precedence. For instance, in the expression '3+5\*7', the multiply operation takes precedence over the add, resulting in the answer 38.

Level	Operation
1	Expressions within parentheses
2	Functions
3	! (Factorial)
4	Power (^) and square root ( $\sqrt{\phantom{x}}$ )
5	Negate (−) multiply(*) divide (/)
6	Add (+) and subtract (-)
7	Relational operators (==, ≠, <, >, ≤, ≥)
8	AND and NOT
9	OR and XOR
10	Left argument for ! (where)
11	=

# Object Types

Type	Object	Example	Type	Object	Example
0	Real number	1.2345	14	XLIB name	XLIB 766 1
1	Complex number	(2.3,4.5)	15	Directory	DIR ... END
2	String	"ABC"	16	Library	Library 766: ...
3	Real array	[ 1 2 3 ]	17	Backup object	Backup HOMEDIR
4	Complex array	[ (1,2) (3,4) ]	18	Built-in function	SIN
5	List	{ "ABC" Var }	19	Built-in command	SWAP
6	Global name	X	20	Internal binary integer	<247d>
7	Local name	y	21	Extended real number	Long Real
8	Program	« A 2 + »	22	Extended complex no.	Long Complex
9	Algebraic	'Y=X^2'	23	Linked array	Linked Array
10	Binary Integer	# 247Ah	24	Character object	Character
11	Graphics object	Graphic 131 x 64	25	Code object	Code
12	Tagged object	Dist: 34.45	26	Library data	Library Data
13	Unit object	32_ft/s^2	27-30	External objects	External

# Reserved Variables

Name	Description	Name	Description
<i>ALRMDAT</i>	Current alarm editing data	<i>PPAR</i>	PLOT parameters
<i>CST</i>	Custom menu contents	<i>PRTPAR</i>	PRINT parameters (HOME directory)
<i>EQ</i>	Current equation for SOLVE and PLOT	<i>VPAR</i>	3D PLOT view volume parameters
<i>EXPR</i>	Current expression for symbolic operations	<i>ZPAR</i>	Stores copy of PPAR from prev. zoom
<i>IERR</i>	Uncertainty of integration	<i>der...</i>	User-defined derivatives begin with <i>der</i>
<i>IOPAR</i>	I/O parameters (HOME directory)	<i>n1, n2, ...</i>	Integers created by ISOL
<i>MHpar</i>	Saves the state of the Minehunt game	<i>s1, s2, ...</i>	Signs created by ISOL and QUAD
<i>Mpar</i>	Multiple Equation Solver equation set	<i>ΣDAT</i>	Current statistical matrix
<i>Nmines</i>	Specifies the number of Minehunt mines	<i>ΣPAR</i>	Statistics parameters
<i>PICT</i>	References the graphics display		

**NOTE:** I/O and print commands affect only the copies of reserved variables *IOPAR* and *PRTPAR* in the HOME directory.

IOPAR <i>default:</i>	{ baud parity receive-pacing transmit-pacing checksum translate-code } { 9600 0 (No parity) 0 0 3 (CRC) 1 (LF to CR-LF) }
PPAR <i>default:</i>	{ ( $x_{\min}$ , $Y_{\min}$ ) ( $X_{\max}$ , $Y_{\max}$ ) indep res ( $X_{\text{axis}}$ , $Y_{\text{axis}}$ ) ptype depend } { (-6.5, -3.1) ( 6.5, 3.2) X 0 (0,0) FUNCTION Y }
PRTPAR <i>default:</i>	{ delay “remap” linelength “lineterm” } { 1.8 “” 80 “CR LF” }
VPAR <i>default:</i>	{ $X_{\text{left}}$ $X_{\text{right}}$ $Y_{\text{near}}$ $Y_{\text{far}}$ $Z_{\text{low}}$ $Z_{\text{high}}$ $X_{\min}$ $X_{\max}$ $X_{\text{eye}}$ $Y_{\text{eye}}$ $Z_{\text{eye}}$ $X_{\text{step}}$ $Y_{\text{step}}$ } { -1 1 -1 1 -1 1 -1 1 -1 1 0 -3 0 10 8 }
ZPAR <i>default:</i>	{ h-factor v-factor recenter-flag { last PPAR } } { 4 4 0 { } }
$\Sigma$ PAR <i>default:</i>	{ independent-column dependent-column intercept slope model } { 1 2 0 0 LINFIT }



# Symbolic Constants

The HP 48 has five constants which may be used in symbolic form or as approximate numerical values:

Name	Machine Value
$\pi$	3.14159265359
e	2.71828182846
i	(0,1)
MAXR	9.999999999999E499
MINR	1.E-499

System flags -2 and -3 control evaluation of symbolic constants:

Flag	Description	Clear	Set	Default
-2	Symbolic Constants	Symbolic form	Numeric form	Clear
-3	Numeric Results	Symbolic results	Numeric results	Clear

# Program Branch Structures

Conditional Structures			Error Traps	
<b>CASE</b> test <sub>1</sub> <b>THEN</b> action <sub>1</sub> <b>END</b> test <sub>2</sub> <b>THEN</b> action <sub>2</sub> <b>END</b> ... test <sub>n</sub> <b>THEN</b> action <sub>n</sub> <b>END</b> default-action (optional) <b>END</b>	<b>IF</b> test-clause <b>THEN</b> true-clause <b>END</b>	<b>IF</b> test-clause <b>THEN</b> true-clause <b>ELSE</b> false-clause <b>END</b>	<b>IFERR</b> trap-clause <b>THEN</b> error-clause <b>END</b>	<b>IFERR</b> trap-clause <b>THEN</b> error-clause <b>ELSE</b> normal-clause <b>END</b>

Definite Loops	Indefinite Loops
start finish <b>START</b> loop-clause <b>NEXT</b> start finish <b>START</b> loop-clause increment <b>STEP</b> start finish <b>FOR</b> counter loop-clause <b>NEXT</b> start finish <b>FOR</b> counter loop-clause increment <b>STEP</b>	<b>DO</b> loop-clause <b>UNTIL</b> test-clause <b>END</b> <b>WHILE</b> test-clause <b>REPEAT</b> loop-clause <b>END</b>

# System Operations



<b>[ON] [A] and [F]</b>	Erases all memory (including port 0 and merged memory) and sets the HP 48 to its default states. Merged memory remains merged.
<b>[ON] [B]</b>	Cancels the current selection if selected before all keys are released.
<b>[ON] [C]</b>	Often known as a system halt or a warmstart, this operation places the calculator in a known state without resetting user memory. The stack is cleared, the VAR directory is set to HOME, the MTH menu is displayed, User mode is cleared, <i>PICT</i> is cleared, and the system configuration is updated to recognize all libraries.
<b>[ON] [D]</b>	Starts the interactive self test.
<b>[ON] [E]</b>	Runs a continuous self test.
<b>[ON] [SPC]</b>	Coma mode: a deep-sleep shutdown which turns off the system timers (including the clock) and clears the system halt log.
<b>[ON] [I/O]</b>	Performs a graphics screen dump in HP 82240A/B graphics format (regardless of the I/O port selection).
<b>[ON] [+ ] or [- ]</b>	Adjusts the display contrast
<b>[ON] [TIME]</b>	Cancels the next repeating alarm.

# Unit Prefixes









































Name	Prefix	HP 48 Symbol	Number	Number	HP 48 Symbol	Prefix	Name
	yotta	Y	+24	−24	y	yocto	
	zetta	Z	+21	−21	z	zepto	
quintillion	exa	E	+18	−18	a	atto	quintillionth
quadrillion	peta	P	+15	−15	f	femto	quadrillionth
trillion	tera	T	+12	−12	p	pico	trillionth
billion	giga	G	+9	−9	n	nano	billionth
million	mega	M	+6	−6	μ	micro	millionth
thousand	kilo	k, K	+3	−3	m	milli	thousandth
hundred	hecto	h, H	+2	−2	c	centi	hundredth
ten	deka	D	+1	−1	d	deci	tenth

Prefix–unit combinations that match built-in units are: au, cd, cu, ft, flam, kph, mph, min, nmi, Pa, ph, pt, yd, and yr.

# Menu Numbers

#	Menu Name	#	Menu Name
0	LAST Menu	18	MTH BASE BYTE
1	CST	19	MTH FFT
2	VAR	20	MTH CMPL
3	MTH	21	MTH CONS
4	MTH VECTR	22	PRG
5	MTH MATR	23	PRG BRCH
6	MTH MATR MAKE	24	PRG IF
7	MTH MATR NORM	25	PRG CASE
8	MTH MATR FACTR	26	PRG START
9	MTH MATR COL	27	PRG FOR
10	MTH MATR ROW	28	 EDIT
11	MTH LIST	29	PRG DO
12	MTH HYP	30	 SOLVE ROOT SOLVR
13	MTH PROB	31	PRG WHILE
14	MTH REAL	32	PRG TEST
15	MTH BASE	33	PRG TYPE
16	MTH BASE LOGIC	34	PRG LIST
17	MTH BASE BIT	35	PRG LIST ELEM

36	PRG LIST PROC	57	☞ UNITS RAD
37	PRG GROB	58	☞ UNITS VISC
38	PRG PICT	59	☞ UNITS
39	PRG IN	60	PRG ERROR IFERR
40	PRG OUT	61	PRG ERROR
41	PRG RUN	62	☞ CHARS
42	☞ UNITS	63	☞ MODES
43	☞ UNITS LENG	64	☞ MODES FMT
44	☞ UNITS AREA	65	☞ MODES ANGL
45	☞ UNITS VOL	66	☞ MODES FLAG
46	☞ UNITS TIME	67	☞ MODES KEYS
47	☞ UNITS SPEED	68	☞ MODES MENU
48	☞ UNITS MASS	69	☞ MODES MISC
49	☞ UNITS FORCE	70	☞ MEMORY
50	☞ UNITS ENRG	71	☞ MEMORY DIR
51	☞ UNITS POWR	72	☞ MEMORY ARITH
52	☞ UNITS PRESS	73	☞ STACK
53	☞ UNITS TEMP	74	☞ SOLVE
54	☞ UNITS ELEC	75	☞ SOLVE ROOT
55	☞ UNITS ANGL	76	☞ SOLVE DIFFEQ
56	☞ UNITS LIGHT	77	☞ SOLVE POLY

78	 SOLVE SYS	98	 STAT $\Sigma$ PAR
79	 SOLVE TVM	99	 STAT $\Sigma$ PAR MODL
80	 SOLVE TVM SOLVR	100	 STAT IVAR
81	 PLOT	101	 STAT PLOT
82	 PLOT PTYPE	102	 STAT FIT
83	 PLOT PPAR	103	 STAT SUMS
84	 PLOT 3D	104	 IO
85	 PLOT 3D PTYPE	105	 IO SRVR
86	 PLOT 3D VPAR	106	 IO IOPAR
87	 PLOT STAT	107	 IO PRINT
88	 PLOT STAT PTYPE	108	 IO PRINT PRTPA
89	 PLOT STAT $\Sigma$ PAR	109	 IO SERIA
90	 PLOT STAT $\Sigma$ PAR MODL	110	 LIBRARY
91	 PLOT STAT DATA	111	 LIBRARY PORTS
92	 PLOT FLAG	112	 LIBRARY
93	 SYMBOLIC	113	 EQ LIB
94	 TIME	114	 EQ LIB EQLIB
95	 TIME ALARM	115	 EQ LIB COLIB
96	 STAT	116	 EQ LIB MES
97	 STAT DATA	117	 EQ LIB UTILS

# Messages

General Messages			
1	Insufficient Memory	292	LAST STACK Disabled
2	Directory Recursion	293	LAST CMD Disabled
3	Undefined Local Name	294	HALT Not Allowed
4	Undefined XLIB Name	295	Array
5	Memory Clear	296	Wrong Argument Count
6	Power Lost	297	Circular Reference
7	Warning:	298	Directory Not Allowed
8	Invalid Card Data	299	Non-Empty Directory
9	Object In Use	300	Invalid Definition
10	Port Not Available	301	Missing Library
11	No Room in Port	302	Invalid PPAR
12	Object Not in Port	303	Non-Real Result
13	Recovering Memory	304	Unable to Isolate
14	Try To Recover Memory?	Low Memory	
15	Replace RAM, Press ON	305	No Room to Show Stack
16	No Mem To Config All	306	Warning
257	No Room to Save Stack	307	Error:
258	Can't Edit Null Char.	308	Purge?
259	Invalid User Function	309	Out of Memory
260	No Current Equation	310	Stack
262	Invalid Syntax	311	Last Stack
		312	Last Commands



Low Memory (cont.)		Array	
313	Key Assignments	1281	Invalid Dimension
314	Alarms	1282	Invalid Array Element
315	Last Arguments	1283	Deleting Row
316	Name Conflict	1284	Deleting Column
317	Command Line	1285	Inserting Row
Stack Operations		1286	Inserting Column
513	Too Few Arguments	Statistics	
514	Bad Argument Type	1537	Invalid $\Sigma$ Data
515	Bad Argument Value	1538	Nonexistent $\Sigma$ DAT
516	Undefined Name	1539	Insufficient $\Sigma$ Data
517	LASTARG Disabled	1540	Invalid $\Sigma$ PAR
EquationWriter		1541	Invalid $\Sigma$ Data LN(Neg)
518	Incomplete Subexpression	1542	Invalid $\Sigma$ Data LN(0)
519	Implicit (~) off	Plot, Solve, Stat	
520	Implicit (~) on	1543	Invalid EQ
Dec	Floating Point Errors	1544	Current equation:
769	Positive Underflow	1545	No current equation.
770	Negative Underflow	1546	Enter eqn, press NEW
771	Overflow	1547	Name the equation, press ENTER
772	Undefined Result	1548	Select plot type
773	Infinite Result	1549	Empty catalog

Plot, Solve, Stat (cont.)		1570	Enter value (zoom out if >1), press ENTER
1550	undefined	1571	Copied to stack
1551	No stat data to plot	1572	x axis zoom w/AUTO.
1552	Autoscaling	1573	x axis zoom.
1553	Solving for	1574	y axis zoom.
1554	No current data. Enter	1575	x and y-axis zoom.
1555	data point, press $\Sigma+$	1576	IR/wire:
1556	Select a model	1577	ASCII/binary:
Alarms		1578	baud:
1557	No alarms pending.	1579	parity:
1558	Press ALRM to create	1580	checksum type:
1559	Next alarm:	1581	translate code:
1560	Past due alarm:	1582	Enter matrix, then NEW
1561	Acknowledged	2561	Bad Guess(es)
1562	Enter alarm, press SET	2562	Constant?
1563	Select repeat interval	2563	Interrupted
I/O, Plot, Solve, Stat		2564	Root
1564	I/O setup menu	2565	Sign Reversal
1565	Plot type:	2566	Extremum
1566	" "	2567	Left
1567	(OFF SCREEN)	2568	Right
1568	Invalid PTYPE	2569	Expr
1569	Name the stat data, press ENTER		

<b>Unit Management</b>		3091	Invalid PRTPAR
2817	Invalid Unit	3092	Low Battery
2818	Inconsistent Units	3093	Empty Stack
<b>I/O and Printing</b>		3094	Row
3073	Bad Packet Block Check	3095	Invalid Name
3074	Timeout	<b>Time</b>	
3075	Receive Error	3329	Invalid Date
3076	Receive Buffer Overrun	3330	Invalid Time
3077	Parity Error	3331	Invalid Repeat
3078	Transfer Failed	3332	Nonexistent Alarm
3079	Protocol Error	<b>Polynomial Root Finder</b>	
3080	Invalid Server Cmd.	49153	Unable to find root
3081	Port Closed	<b>Multiple Equation Solver</b>	
3082	Connecting	58369	Invalid Mpar
3083	Retry #	58370	Single Equation
3084	Awaiting Server Cmd.	58371	EQ Invalid for MINIT
3085	Sending	58372	Too Many Unknowns
3086	Receiving	58373	All Variables Known
3087	Object Discarded	58374	Illegal During MROOT
3088	Packet #	58375	Solving for
3089	Processing Command	58376	Searching
3090	Invalid IOPAR		

# Character Codes

NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR
0	▪	32		64	è	96	`	128	€	160		192	À
1	▪	33	!	65	À	97	a	129	¥	161	í	193	Á
2	▪	34	"	66	B	98	b	130	¢	162	ê	194	Â
3	▪	35	#	67	C	99	c	131	£	163	ë	195	Ã
4	▪	36	\$	68	D	100	d	132	¢	164	ä	196	Ä
5	▪	37	%	69	E	101	e	133	Σ	165	¥	197	Å
6	▪	38	&	70	F	102	f	134	►	166	ı	198	Æ
7	▪	39	'	71	G	103	g	135	π	167	ş	199	Ç
8	▪	40	<	72	H	104	h	136	ð	168	¨	200	È
9	▪	41	>	73	I	105	i	137	≤	169	ø	201	É
10	▪	42	*	74	J	106	j	138	≥	170	à	202	Ê
11	▪	43	+	75	K	107	k	139	≠	171	«	203	Ë
12	▪	44	,	76	L	108	l	140	α	172	¬	204	Ì
13	▪	45	-	77	M	109	m	141	→	173	-	205	Í
14	▪	46	.	78	N	110	n	142	←	174	®	206	Î
15	▪	47	/	79	O	111	o	143	↓	175	™	207	Ï
												224	à
												225	á
												226	â
												227	ã
												228	ä
												229	å
												230	æ
												231	ç
												232	è
												233	é
												234	ê
												235	ë
												236	ì
												237	í
												238	î
												239	ï

NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR
16	▪	48	Ø	80	P	112	Ɔ	144	↑	176	▫	208	Đ
17	▪	49	1	81	Q	113	q	145	γ	177	±	209	Ñ
18	▪	50	2	82	R	114	r	146	δ	178	ℤ	210	ò
19	▪	51	3	83	S	115	s	147	€	179	Ʒ	211	ó
20	▪	52	4	84	T	116	t	148	η	180	´	212	ô
21	▪	53	5	85	U	117	u	149	θ	181	μ	213	õ
22	▪	54	6	86	V	118	v	150	λ	182	¶	214	ö
23	▪	55	7	87	W	119	w	151	ρ	183	•	215	×
24	▪	56	8	88	X	120	x	152	σ	184	˘	216	ø
25	▪	57	9	89	Y	121	y	153	τ	185	ı	217	ù
26	▪	58	:	90	Z	122	z	154	ω	186	◊	218	ú
27	▪	59	;	91	[	123	{	155	△	187	⌘	219	û
28	▪	60	<	92	\	124		156	Π	188	¼	220	ü
29	▪	61	=	93	]	125	}	157	Ω	189	½	221	ý
30	▪	62	>	94	^	126	~	158	▪	190	¾	222	þ
31	...	63	?	95	_	127	⌘	159	∞	191	¿	223	ÿ

# Command Reference

This command reference summarizes the stack operations in the HP 48. Each stack diagram is representative of the usage of the command, but does not illustrate all of the possible combinations of arguments that can be used. The following table lists the terms used in the stack diagrams.

Note that system modes may affect the interpretation of input parameters or the results of some functions.

Term	Description
obj	Any object
x or y	Real number
a b c d	Real number
(x,y)	Complex number or user-unit graphics coordinates
z	Real or complex number
m or n	Positive integer real number (rounded if non-integer)
#n or #m	Binary integer
x_unit	Real number with units
“string”	Character string
{ list }	List of objects
grob	Graphics object
{ #x #y }	Pixel coordinates

Term	Description (cont.)
hms	Real number in HH.MMSS format
time	Time in HH.MMSS format
date	Date in current MM.DDYYYY or DD.MMYYYY format (flag -42)
T/F	Test result: 0 (false) or non-zero (true)
'symb'	Expression or name treated as an algebraic
[vector]	Real or complex vector
[[matrix]]	Real or complex matrix
{row col}	Coordinates of an element in a matrix
position	Real number specifying an element in a list, vector, or matrix. May be a list containing two real numbers specifying an element in a matrix.
'name'	Global or local name
'global'	Global name
rc or rc.p	Key location: row-col or row-col.plane
mm.pp	Menu specified as menu.page
d.o.f.	Degrees of freedom (positive integer)
port	Port number: 0 - 33 or & (wildcard)
backup	Backup object
library	Library object
LID	Library identifier (port:library number)

Command Reference Example						
NAME	Description	3:	2:	1:	3:	2: 1:
OVER	Copies the level 2 object into level 1		obj <sub>1</sub>	obj <sub>2</sub> →	obj <sub>1</sub>	obj <sub>2</sub> obj <sub>1</sub>

Note that most commands which do not accept lists as one of the required arguments have been extended to accept arguments in a list, execute once for each argument (or set of arguments), and return the results in a list. For instance, the program `« { 3 5 } SF »` sets user flags 3 and 5.

26

The program

```
« { 1 2 6 } { 3 7 8 } * »
```

returns the list

```
{ 3 14 48 }
```

A command like `LINE` which can accept coordinates in list form (pixel coordinates) cannot take lists of coordinates.



ABS	Absolute value	$z \rightarrow  z $
ACK	Acknowledges displayed past due alarm	$\rightarrow$
ACKALL	Acknowledges all past due alarms	$\rightarrow$
ACOS	Arc cosine	$z \rightarrow \text{acos } z$
ACOSH	Inverse hyperbolic cosine	$z \rightarrow \text{acosh } z$
ADD	Same as +, but does element-wise addition of objects in lists	$\{ \text{list}_1 \} \{ \text{list}_2 \} \rightarrow \{ \text{list}_3 \}$
ALOG	Antilogarithm	$z \rightarrow 10^z$
AMORT	Calculates amortization from TVM variables	$\text{pmts} \rightarrow \text{prin int bal}$
AND	Logical or binary AND	$\#n_1 \#n_2 \rightarrow \#n_3$
ANIMATE	Animates a series of grobs	$\text{grob}_1 \dots \text{grob}_n \text{ n} \rightarrow \text{grob}_1 \dots \text{grob}_n \text{ n}$
APPLY	Returns evaluated expression as argument to an unevaluated local name	$\{ \text{symb} \dots \} \text{'name'} \rightarrow \text{'name(symb...)'}$
ARC	Draws an arc in <i>PICT</i> centered at (x,y), radius $r$ , counterclockwise from $\theta_1$ to $\theta_2$	$(x,y) \text{ r } \theta_1 \theta_2 \rightarrow$
ARCHIVE	Makes backup copy of HOME directory	$\text{:IO:name} \rightarrow$
ARG	Argument	$z \rightarrow \theta$
ARRY $\rightarrow$	Separates array into individual elements	$[\text{array}] \rightarrow z_1 \dots z_n \{ \text{dim} \}$
$\rightarrow$ ARRY	Combines numbers into an array	$z_1 \dots z_n \{ \text{dim} \} \rightarrow [\text{array}]$
ASIN	Arc sine	$z \rightarrow \text{asin } z$

ASINH	Inverse hyperbolic sine	$z \rightarrow$	$\operatorname{asinh} z$
ASN	Make a single user-key assignment	object rc.p $\rightarrow$	
ASR	Arithmetic shift right (preserves MSB)	$\#n_1 \rightarrow$	$\#n_2$
ATAN	Arc tangent	$z \rightarrow$	$\operatorname{atan} z$
ATANH	Inverse hyperbolic tangent	$z \rightarrow$	$\operatorname{atanh} z$
ATICK	Specifies the tick spacing on plot axes	$\{ x \ y \} \rightarrow$	
ATTACH	Attaches library to current directory	LID $\rightarrow$	
AUTO	Scales y-axis	$\rightarrow$	
AXES	Sets the intersection of axes	$(x,y) \rightarrow$	
BAR	Selects bar plot	$\rightarrow$	
BARPLOT	Draws a bar plot of the data in $\Sigma DAT$	$\rightarrow$	
BAUD	Sets the serial baud rate	$n \rightarrow$	
BEEP	Sounds a beep	Hz secs $\rightarrow$	
BESTFIT	Selects the statistics model that yields largest correlation coefficient and executes LR	$\rightarrow$	
BIN	Sets binary base	$\rightarrow$	
BINS	Sorts $\Sigma DAT$ data into bins using the independent variable column as the key	$X_{\min} \text{ width } N \rightarrow$	$[[b_1] \dots [b_n]] [b_L \ b_R]$
BLANK	Creates a blank graphics object	$\#width \ \#height \rightarrow$	
BOX	Draws a box in <i>PICT</i>	$\{ \#x_1 \ \#y_1 \} \ \{ \#x_2 \ \#y_2 \} \rightarrow$	

BUFLEN	Returns the number of characters in the serial buffer and 1 if no error occurred	→ n	T/F
BYTES	Returns the checksum and size of an object	object → #cksum	size
B→R	Binary to real conversion	#n →	n
CASE	Begins CASE...THEN...END...END	→	
CEIL	Next greatest integer	x →	n
CENTR	Sets center of plot display	(x,y) →	
CF	Clears a system or user flag	±n →	
CHOOSE	Displays a choose box with highlight positioned at the specified element	“title” { list } start →	obj T/F
CHR	Makes a one character string	n →	“string”
CKSM	Selects the checksum scheme	n →	
CLEAR	Clears the stack	objects →	
CLKADJ	Adds clock ticks (8192/sec) to the system time	±n →	
CLLCD	Clears the stack display	→	
CLOSEIO	Closes the serial port, clears input buffer	→	
CLTEACH	Purges the examples directory	→	
CLUSR	Purges all user variables in the current directory	→	
CLVAR	Purges the statistics matrix $\Sigma DAT$	→	
CNRM	Computes the column norm of an array	[array] →	column-norm

COL+	Inserts a column vector into an array	$[[\text{matrix}]] \text{ [vector] } n \rightarrow [[\text{matrix}]]'$
COL-	Deletes a column from a matrix	$[[\text{matrix}]] \text{ } n \rightarrow [[\text{matrix}]]'$
COL→	Transforms column vectors into a matrix	$[\text{col}_1] \dots [\text{col}_n] \text{ } n \rightarrow [[\text{matrix}]]$
→COL	Transforms a matrix into column vectors	$[[\text{matrix}]] \rightarrow [\text{col}_1] \dots [\text{col}_n] \text{ } n$
COLCT	Collects like terms	$'\text{symb}_1' \rightarrow '\text{symb}_2'$
COLΣ	Specifies independent and dependent statistics columns	$\text{indep } \text{depend} \rightarrow$
COMB	Combinations of $n$ object taken $m$ at a time	$n \text{ } m \rightarrow C_{n,m}$
CON	Creates constant array	$\{ \text{rows cols} \} \text{ } z \rightarrow [[\text{matrix}]]$
COND	Estimates the column norm condition number of a square matrix	$[[\text{matrix}]] \rightarrow \text{condition-number}$
CONIC	Selects conic plot	$\rightarrow$
CONJ	Complex conjugate	$(x,y) \rightarrow (x,-y)$
CONLIB	Displays the Constants Library catalog	$\rightarrow$
CONST	Returns the value of a constant	$'\text{constname}' \rightarrow \text{constant}$
CONT	Continues a halted program	$\rightarrow$
CONVERT	Performs a unit conversion	$x_{\text{old}} \text{ } y_{\text{new}} \rightarrow x_{\text{new}}'$
CORR	Correlation coefficient of ΣDAT data	$\rightarrow \text{correlation}$
COS	Cosine	$z \rightarrow \cos z$
COSH	Hyperbolic cosine	$z \rightarrow \cosh z$

COV	Sample covariance of $\Sigma DAT$ data	→	covariance
CR	Prints a carriage-right	→	
CRDIR	Creates a directory	'name' →	
CROSS	Cross product	[ A ] [ B ] →	[ A x B ]
CSWP	Column swap	[[matrix]] n m →	[[matrix]]'
CYLIN	Sets polar/cylindrical mode	→	
C→PX	User-unit to pixel coordinate conversion	(x,y) →	{ #x #y }
C→R	Complex-to-real conversion	(x,y) →	x y
DARCY	Calculates Darcy friction factor	e/D Re →	d
DATE	Returns the system date	→	date
→DATE	Sets the system date	date →	
DATE+	Adds a number of days to a date	date ±days →	date'
DDAYS	Number of days between two dates	date <sub>1</sub> date <sub>2</sub> →	Δdays
DEC	Sets decimal base	→	
DECR	Decrements and returns value of variable	'name' →	x
DEFINE	Creates user-defined function	'name=expression' →	
DEG	Sets Degrees mode	→	
DELALARM	Deletes alarm from alarm list	n →	
DELAY	Sets delay between printed lines	secs →	
DELKEYS	Clears user-key assignments	rc.p →	

DEPND	Specifies plot dependent variable	'name' →	
DEPTH	Counts the objects on the stack	objects →	objects n
DET	Determinant of a square matrix	[[matrix]] →	determinant
DETACH	Detaches library from current directory	library-number →	
→DIAG	Returns vector of major diagonal elements	[[matrix]] →	[diagonals]
DIAG→	Creates matrix with specified diagonal elements	[diagonals] n →	[[nXn matrix]]
DIFFEQ	Selects differential equation plot	→	
DISP	Displays object in specified display line	object n →	
DO	Begins DO ... UNTIL ... END structure	→	
DOERR	Generates system or user-defined error	n →	
DOLIST	Applies $n$ arguments to an object	{list <sub>1</sub> }...{list <sub>n</sub> } n obj →	{results}
DOSUBS	Executes object using arguments in list	{list} n object →	{ list }'
DOT	Dot product of two vectors	[vector A] [vector B] →	x
DRAW	Draws a plot	→	
DRAX	Draws axes	→	
DROP	Drops one object off the stack	object →	
DROPN	Drops $n$ objects from the stack	obj <sub>n</sub> ... obj <sub>1</sub> n →	
DROP2	Drops two objects from the stack	obj <sub>2</sub> obj <sub>1</sub> →	
DTAG	Removes all tags from an object	:tag:obj →	obj

DUP	Duplicates one object on the stack	obj	→	obj obj
DUPN	Duplicates $n$ objects on the stack	obj <sub>n</sub> ... obj <sub>1</sub> n	→	obj <sub>n</sub> ...obj <sub>1</sub> obj <sub>n</sub> ...obj <sub>1</sub>
DUP2	Duplicates two objects on the stack	obj <sub>1</sub> obj <sub>2</sub>	→	obj <sub>1</sub> obj <sub>2</sub> obj <sub>1</sub> obj <sub>2</sub>
D→R	Degrees-to-radians conversion	x	→	( $\pi/180$ )x
e	Symbolic constant $e$		→	2.71828182846
EGV	Computes eigenvalue and right eigenvectors of a square matrix	[[matrix]	→	[eigenvectors] [eigenvalues]
EGVL	Computes eigenvalues of a square matrix	[[matrix]]	→	[eigenvalues]
ELSE	Begins false clause in IF...THEN...ELSE...END structure		→	
END	Ends program structures		→	
ENDSUB	Returns the number of frames in argument list for DOSUBS		→	n
ENG	Sets Engineering display mode	n	→	
EQ→	Separates equation into left and right sides	'symp <sub>1</sub> =symp <sub>2</sub> '	→	'symp <sub>1</sub> ' 'symp <sub>2</sub> '
EQNLIB	Displays the Equation Library catalog		→	
ERASE	Erases <i>PICT</i>		→	
ERRM	Returns the last error message		→	"string"
ERRN	Returns the last error number		→	#n
ERR0	Clears the last error number		→	
EVAL	Evaluates an object	obj	→	

EXP	Natural exponential	$z \rightarrow \exp z$
EXPAN	Expands an algebraic	'symb <sub>1</sub> ' $\rightarrow$ 'symb <sub>2</sub> '
EXPFIT	Selects exponential curve-fitting model	$\rightarrow$
EXPM	Natural exponential minus 1	$x \rightarrow \exp(x)-1$
EYEPT	Specifies eyepoint coordinates for 3D plot	$x_{eye} \ y_{eye} \ z_{eye} \rightarrow$
F0 $\lambda$	Calculates fraction of black-body emissive power at temperature T between wavelengths 0 and $\lambda$	$\lambda \ T \rightarrow$ fraction
FACT	Factorial or gamma function	$n \rightarrow n!$
FANNING	Calculates Fanning friction factor	$e/D \ Re \rightarrow f$
FC?	Tests a system or user flag	$\pm n \rightarrow T/F$
FC?C	Tests and clears a system or user flag	$\pm n \rightarrow T/F$
FFT	Discrete Fourier transform	[array] $\rightarrow$ [array]'
FINDALARM	Returns alarm index $n$ or 0 if no alarm is found	date $\rightarrow n$
FINISH	Terminates Kermit server mode	$\rightarrow$
FIX	Sets Fix display mode	$n \rightarrow$
FLOOR	Next smallest integer	$x \rightarrow n$
FOR	Begins FOR ... NEXT or FOR ... STEP structure	start end $\rightarrow$
FP	Fractional part	$x \rightarrow y$
FREE	Frees (makes independent) merged memory (port 1 only)	{name/LID} port $\rightarrow$



FREE1	Frees (makes independent) merged memory in port 1	{name/LID} →	
FREEZE	Freezes up to three display areas specified by lower three bits	n →	
FS?	Tests a system or user flag	±n →	T/F
FS?C	Tests and clears a system or user flag	±n →	T/F
FUNCTION	Selects function plot	→	
GET	Gets an element from a list or array	{list} index →	object
GETI	Gets an element from a list or array	{list} index →	{list} index' obj
GOR	Superimposes grob' onto grob at the specified coordinates	grob { #x #y } grob' →	grob''
GRAD	Selects Grads mode	→	
GRAPH	Enters the Graphics environment	→	
GRIDMAP	Selects the gridmap plot type	→	
→GROB	Converts object into graphics object	object size →	grob
GXOR	Superimposes and inverts grob' onto grob at the specified coordinates	grob { #x #y } grob' →	grob''
*H	Multiplies the vertical plot scale by specified factor	x →	
HALT	Suspends program execution until either CONT or KILL are executed	→	
HEAD	Returns the first object in a list	{ obj <sub>1</sub> ... obj <sub>n</sub> } →	obj <sub>1</sub>

HEX	Sets hexadecimal base		→
HISTOGRAM	Selects histogram plot		→
HISTPLOT	Draws a histogram of the data in $\Sigma DAT$		→
HMS+	Adds in HH.MMSSs format	$hms_1 \ hms_2$	→ $hms_1+hms_2$
HMS-	Subtracts in HH.MMSSs format	$hms_1 \ hms_2$	→ $hms_1-hms_2$
HMS→	Converts a number from HH.MMSSs format	$hms$	→ $x$
→HMS	Converts a number to HH.MMSSs format	$x$	→ $hms$
HOME	Selects the HOME directory		→
i	Symbolic constant $i$		→ (0,1)
IDN	Creates an identity matrix	$n$	→ $[[n \times n \text{ matrix}]]$
IF	Begins IF ... THEN ... END or IF ... THEN ... ELSE ... END structure		→
IFERR	Begins IFERR ... THEN ... END or IFERR ... THEN ... ELSE ... END structure		→
IFFT	Inverse discrete Fourier transform	$[array]$	→ $[array]'$
IFT	IF ... THEN ... END test. Executes <i>obj</i> if T/F is true.	T/F $obj$	→
IFTE	IF ... THEN ... ELSE ... END test	T/F true-obj false-obj	→
IM	Returns imaginary part of a number	$(x,y)$	→ $y$
INCR	Increments and returns value of variable	'name'	→ $x$
INDEP	Specifies plot independent variable	'name'	→

INFORM	Displays input form	“title” {labels} fmt {defaults} {initial values} →	{data} T/F
INPUT	Suspends program, displays message, and waits for data	“message” “prompt” →	object
INV	Inverse (reciprocal)	z →	1/z
IP	Integer part	x →	n
ISOL	Isolates a variable in an equation	‘symb <sub>1</sub> ’ ‘global’ →	‘symb <sub>2</sub> ’
KERRM	Returns the last Kermit error message	→	“message”
KEY	Returns number indicating last key pressed	→	rc 1 <i>or</i> 0
KGET	Gets named data from remote device	‘name’ →	
KILL	Cancels all suspended programs	→	
LABEL	Labels axes	→	
LAST	Returns last arguments	→	object(s)
LASTARG	Returns last arguments	→	object(s)
LCD→	Returns LCD as 131x64 pixel graphics object	→	grob
→LCD	Displays graphics object in the upper-left corner of the stack display	grob →	
LIBEVAL	Executes library object. Upper three digits are library number, lower three digits are function number.	#n →	

LIBS	Lists library objects attached to the current directory	→	{ title LID port ... }
LINE	Draws a line in <i>PICT</i>	{ #x <sub>1</sub> #y <sub>1</sub> } { #x <sub>2</sub> #y <sub>2</sub> } →	
ΣLINE	Returns best-fit line for ΣDAT data	→	'symb'
LINFIT	Selects linear curve-fitting model	→	
LININ	Determines if an equation is linear in a variable	'symb' 'name' →	T/F
LIST→	Separates a list into individual elements	{ obj <sub>1</sub> ... obj <sub>n</sub> } →	obj <sub>1</sub> ... obj <sub>n</sub> n
→LIST	Combines <i>n</i> objects into a list	obj <sub>1</sub> ... obj <sub>n</sub> n →	{ obj <sub>1</sub> ... obj <sub>n</sub> }
ΔLIST	Computes first differences of objects in a list	{ obj <sub>1</sub> obj <sub>2</sub> ... obj <sub>n</sub> } →	{ obj <sub>2-1</sub> obj <sub>3-2</sub> ... }
ΣLIST	Sums the elements in a list	{ obj <sub>1</sub> ... obj <sub>n</sub> } →	obj <sub>1</sub> +...+obj <sub>n</sub>
ΠLIST	Returns the product of the elements in a list	{ obj <sub>1</sub> ... obj <sub>n</sub> } →	obj <sub>1</sub> *...*obj <sub>n</sub>
LN	Natural logarithm	z →	ln z
LNP1	Natural logarithm of (argument+1)	x →	ln(x+1)
LOG	Common (base 10) logarithm	z →	log z
LOGFIT	Selects logarithmic curve-fitting model	→	
LQ	Returns LQ factorization of a matrix	[[A]] →	[[L]] [[Q]] [[P]]
LR	Computes linear regression of ΣDAT data	→	intercept slope
LSQ	Returns minimum norm least squares solution to system or linear equations	[B] [[A]] →	[X]
LU	Returns Crout LU decomposition of a square matrix	[[A]] →	[[L]] [[U]] [[P]]
MANT	Returns the mantissa of a number	x →	y

↑MATCH	Match-and-replace, beginning with subexpressions	'symb' { 'pattern' 'replacement' } →	'result' T/F
↓MATCH	Match-and-replace, beginning with top level expression	'symb' { 'pattern' 'repl' } →	'result' T/F
MAX	Returns the maximum of two numbers	x y →	max(x,y)
MAXR	Symbolic constant – maximum real number	→	'MAXR'
MAXΣ	Finds the maximum column values of data in $\Sigma DAT$	→	$[x_1 \dots x_m]$
MCALC	Sets Multiple Equation Solver variable to <i>not</i> user-defined state	'name' →	
MEAN	Computes means of the data in $\Sigma DAT$	→	$[\bar{x}_1 \dots \bar{x}_m]$
MEM	Returns available memory	→	bytes
MENU	Selects a built-in menu or creates a custom menu	mm.pp →	
MERGE	Merges RAM card with main memory	port →	
MERGE1	Merges RAM card in port 1 with main memory	→	
MIN	Returns the minimum of two numbers	x y →	min(x,y)
MINEHUNT	Starts the Minehunt game	→	
MINIT	Establishes <i>Mpar</i> from <i>EQ</i>	→	
MINR	Symbolic constant – minimum real number	→	'MINR'
MINΣ	Finds the minimum column values of data in $\Sigma DAT$	→	$[x_1 \dots x_m]$
MITM	Changes the title and variable menu in <i>Mpar</i>	"title" { names } →	

MOD	Modulo	$x \ y \rightarrow$	$x \bmod y$
MROOT	Solves for single ('name') or all ("ALL") variables using the Multiple Equation Solver	'name' $\rightarrow$	value
MSGBOX	Displays a message, waits for keystroke	"message" $\rightarrow$	
MSOLVR	Displays the Multiple Equation Solver menu	$\rightarrow$	
MUSER	Sets Multiple Equation Solver variable to user-defined state	'name' $\rightarrow$	
NDIST	Normal probability density	mean variance $x \rightarrow$	$\text{ndist}(\text{mean}, \text{variance}, x)$
NEG	Negates an argument	$z \rightarrow$	$-z$
NEWOBJ	Separates object from list or backup object	object $\rightarrow$	object
NEXT	Ends FOR ... NEXT or START ... NEXT	$\rightarrow$	
NOT	Logical or binary NOT	$\#n_1 \rightarrow$	$\#n_2$
NOVAL	Placeholder for unspecified values in INFORM argument list	$\rightarrow$	NOVAL
NSUB	Returns the current frame number for DOSUBS	$\rightarrow$	$n$
NUM	Returns character code of a string's first character	"string" $\rightarrow$	$n$
$\rightarrow$ NUM	Evaluates an object to yield a numeric result	object $\rightarrow$	$z$
NUMX	Specifies number of plot increments in X for 3D plot types	$n \rightarrow$	
NUMY	Specifies number of plot increments in Y for 3D plot types	$n \rightarrow$	
$N\Sigma$	Returns the number of data points in $\Sigma DAT$	$\rightarrow$	$n$
OBJ $\rightarrow$	Decomposes an object onto the stack	$\{ \text{obj}_1 \dots \text{obj}_n \} \rightarrow$	$\text{obj}_1 \dots \text{obj}_n \ n$

OCT	Sets octal base	→	
OFF	Turns the calculator off	→	
OLDPRT	Remaps printer output to the HP 82240A character set	→	
OPENIO	Opens IR or wired port	→	
OR	Logical or binary OR	#n <sub>1</sub> #n <sub>2</sub> →	#n <sub>3</sub>
ORDER	Rearranges the VAR menu	{ names } →	
OVER	Copies the level 2 object into level 1	obj <sub>1</sub> obj <sub>2</sub> →	obj <sub>1</sub> obj <sub>2</sub> obj <sub>1</sub>
PARAMETRIC	Selects parametric plot	→	
PARITY	Sets parity	n →	
PARSURFACE	Selects parametric surface plot	→	
PATH	Returns a list showing the current path	→	{ HOME names }
PCOEF	Computes coefficients of a polynomial with specified roots	[roots] →	[coefficients]
PCONTOUR	Selects pseudo-contour plot	→	
PCOV	Population covariance of $\Sigma DAT$ data	→	
PDIM	Changes the size of <i>PICT</i>	#horizontal #vertical →	
PERM	Permutations of $n$ objects taken $m$ at a time	n m →	P <sub>n,m</sub>
PEVAL	Evaluates polynomial with specified coefficients at $x$	[coefficients] x →	y
PGDIR	Purges specified directory and its contents	'name' →	
PICK	Copies $n$ th object into level 1	obj <sub>n</sub> ... obj <sub>1</sub> n →	obj <sub>n</sub> ... obj <sub>1</sub> obj <sub>n</sub>

PICT	Returns the name <i>PICT</i> to level 1	→	<i>PICT</i>
PICTURE	Enters the Graphics environment	→	
PINIT	Initializes RAM card in port 2	→	
PIXOFF	Turns off a pixel in <i>PICT</i>	{ #x #y } →	
PIXON	Turns on a pixel in <i>PICT</i>	{ #x #y } →	
PIX?	Tests a pixel in <i>PICT</i>	{ #x #y } →	T/F
PKT	Sends commands to server	“contents” “type” →	“response”
PMAX	Sets the upper-right plot coordinates	(x,y) →	
PMIN	Sets the lower-left plot coordinates	(x,y) →	
POLAR	Selects polar plot	→	
POS	Finds substring in string or object in list	{ list } object →	n
PREDV	Predicted dependent variable value	x →	predicted-value
PREDX	Predicted independent variable value	y →	predicted-value
PREDY	Predicted dependent variable value	x →	predicted-value
PRLCD	Prints an image of the display	→	
PROMPT	Displays a prompt and halts the program	“prompt” →	
PROOT	Computes polynomial roots	[ z <sub>1</sub> ... z <sub>n</sub> ] →	[ roots ]
PRST	Prints the stack	→	
PRSTC	Prints the stack in compact format	→	
PRVAR	Prints the name and contents of a variable	‘name’ →	
PR1	Prints the level 1 object	object →	object



PSDEV	Computes population standard deviation of data in $\Sigma DAT$	$\rightarrow [x_1 \dots x_m]$
PURGE	Purges one or more variables	'name' $\rightarrow$
PUT	Replaces an element in an array or list	{ list } index obj $\rightarrow$ { list }'
PUTI	Replaces an element in an array or list and increments the position	{ list } index obj $\rightarrow$ { list }' index'
PVAR	Computes population variances of $\Sigma DAT$ data in columns specified by $COL\Sigma$	$\rightarrow [x_1 \dots x_m]$
PVARS	Returns list of objects in a port	port $\rightarrow$ { list } type/bytes
PVIEW	Displays <i>PICT</i> with specified coordinate in the upper-left corner	{ #x #y } $\rightarrow$
PWRFIT	Selects power curve-fitting model	$\rightarrow$
PX $\rightarrow$ C	Pixel to user-unit coordinate conversion	{ #x #y } $\rightarrow$ (x,y)
$\rightarrow$ Q	Converts numbers to fractional equivalent	x $\rightarrow$ 'a/b'
$\rightarrow$ Q $\pi$	$\rightarrow$ Q after factoring out $\pi$	x $\rightarrow$ 'a/b* $\pi$ '
QR	Computes QR factorization of a matrix	[[A]] $\rightarrow$ [[Q]] [[R]] [[P]]
QUAD	Solves a quadratic polynomial	'symp <sub>1</sub> ' 'global' $\rightarrow$ 'symp <sub>2</sub> '
QUOTE	Returns argument expression unevaluated	'symp' $\rightarrow$ 'symp'
RAD	Sets Radians mode	$\rightarrow$
RAND	Returns a random number between 0 and 1	$\rightarrow$ x

RANK	Estimates the rank of a rectangular matrix	[[matrix]]	→	rank
RANM	Replaces or creates a matrix with random integers from -9 to 9	[[matrix]]	→	[[matrix]]'
RCEQ	Recalls the contents of the variable <i>EQ</i>		→	object
RCI	Multiplies elements in a row of a matrix by a factor	[[matrix]] factor row	→	[[matrix]]'
RCIJ	Multiplies elements in row <sub>i</sub> of a matrix by a factor and adds the result to elements in row <sub>j</sub>	[[matrix]] factor i j	→	[[matrix]]'
RCL	Recalls the contents of a variable	'name'	→	object
RCLALARM	Recalls alarm from alarm list	n	→	{ alarm }
RCLF	Returns a list containing two binary integers representing the system and user flags		→	{ #system #user }
RCLKEYS	Lists user-key assignments		→	{ obj <sub>1</sub> rc.p <sub>1</sub> ... }
RCLMENU	Recalls number and page of active menu		→	mm.pp
RCL $\Sigma$	Recalls the contents of $\Sigma$ DATA		→	object
RCWS	Recalls the binary integer wordsize		→	n
RDM	Redimensions a matrix	[[matrix <sub>1</sub> ]] {dim}	→	[[matrix <sub>2</sub> ]]
RDZ	Sets the random number seed	x	→	
RE	Returns the real part of complex number or array	(x,y)	→	x
REC�	Receives file from remote Kermit and saves in named object	'name'	→	

RECV	Receives file from remote Kermit and saves in sender-named object	→
REPEAT	Begins loop clause in WHILE ... REPEAT ... END structure	T/F →
REPL	Replaces level 1 object onto level 3 object at position $n$	$\{ \text{list} \} \ n \ \text{object} \rightarrow \{ \text{list} \}'$
RES	Sets the plot resolution	$\#n \rightarrow$
RESTORE	Replaces HOME directory with backup	backup →
REVLIST	Reverses the order of objects in a list	$\{ \text{obj}_1 \dots \text{obj}_n \} \rightarrow \{ \text{obj}_n \dots \text{obj}_1 \}$
RKF	Computes solution of initial value problem using RKF method	$\{ t \ y_0 \ f(t,y) \} \ \text{tol} \ t_f \rightarrow \{ t_0 \ y_0 \ f(t,y) \} \ \text{tol}$
RKFERR	Computes change in solution and error estimate using RKF method	$\{ t \ y \ f(t,y) \} \ \text{stepsize} \rightarrow \{ t \ y \ f(t,y) \} \ \text{stepsize} \ \Delta y \ \text{error}$
RKFSTEP	Computes next solution step of initial value problem using RKF method	$\{ t \ y \ f(t,y) \} \ \text{tol} \ \text{stepsize} \rightarrow \{ t \ y \ f(t,y) \} \ \text{tol} \ \text{next-step}$
RL	Rotates left by one bit	$\#n_1 \rightarrow \#n_2$
RLB	Rotates left by one byte	$\#n_1 \rightarrow \#n_2$
RND	Rounds fractional part of number to $n$ places	$z_1 \ n \rightarrow z_2$
RNRM	Computes maximum value of the sums of the absolute values of all elements over all rows	$[\text{array}] \rightarrow \text{row-norm}$
ROLL	Moves the level $n+1$ object to level 1	$\text{obj}_n \dots \text{obj}_1 \ n \rightarrow \text{obj}_{n-1} \dots \text{obj}_1 \ \text{obj}_n$

ROLLD	Moves the level 2 object to level $n$	$\text{obj}_n \dots \text{obj}_1 \ n \rightarrow \text{obj}_1 \ \text{obj}_n \dots \text{obj}_2$
ROOT	Finds a numerical root	'symb' 'name' guess $\rightarrow$ root
ROT	Moves the level 3 object to level 1	$\text{obj}_3 \ \text{obj}_2 \ \text{obj}_1 \rightarrow \text{obj}_2 \ \text{obj}_1 \ \text{obj}_3$
ROW+	Inserts a row vector into a matrix	$[[\text{matrix}]] \ [\text{vector}] \ n \rightarrow [[\text{matrix}]]'$
ROW-	Deletes a row from a matrix	$[[\text{matrix}]] \ n \rightarrow [[\text{matrix}]]'$
ROW $\rightarrow$	Transforms a series of row vectors into a matrix	$[\text{vector}_1] \dots [\text{vector}_n] \ n \rightarrow [[\text{matrix}]]$
$\rightarrow$ ROW	Transforms a matrix into a series of row vectors	$[[\text{matrix}]] \rightarrow [\text{vector}_1] \dots [\text{vector}_n] \ n$
RR	Rotates right by one bit	$\#n_1 \rightarrow \#n_2$
RRB	Rotates right by one byte	$\#n_1 \rightarrow \#n_2$
RREF	Computes reduced row echelon form of a rectangular matrix	$[[\text{matrix}]] \rightarrow [[\text{matrix}]]'$
RRK	Computes solution of initial value problem using Rosenbrock and RKF methods	$\{ t \ y_0 \ f(t,y) \} \ \text{tol} \ t_f \rightarrow \{ t_0 \ y_0 \ f(t,y) \} \ \text{tol}$
RRKSTEP	Computes next solution step of initial value problem using Rosenbrock and RKF methods	$\{ t \ y \ f(t,y) \ \partial f/\partial T \ \partial f/\partial y \} \ \text{tol} \ \text{stepsize} \rightarrow \{ t \ y \ f(t,y) \ \partial f/\partial T \ \partial f/\partial y \} \ \text{tol} \ \text{next-step}$
RSBERR	Computes change in solution and error estimate using Rosenbrock and RKF methods	$\{ t \ y \ f(t,y) \ \partial f/\partial T \ \partial f/\partial y \} \ \text{stepsize} \rightarrow \{ t \ y \ f(t,y) \ \partial f/\partial T \ \partial f/\partial y \} \ \text{stepsize} \ \Delta y \ \text{error}$
RSD	Computes a correction to the solution of a system of equations	$[B] \ [[A]] \ [Z] \rightarrow [B-AZ]$
RSWP	Row swap	$[[\text{matrix}]] \ \text{index}_1 \ \text{index}_2 \rightarrow [[\text{matrix}]]'$

R→B	Real-to-binary conversion	n → #n
R→C	Real-to-complex conversion	x y → (x,y)
R→D	Radians-to-degrees conversion	x → (180/π)x
SAME	Tests two objects for equality	obj <sub>1</sub> obj <sub>2</sub> → T/F
SBRK	Sends serial break	→
SCALE	Specifies x and y scale in units per 10 pixels	x y →
SCATRLOT	Draws a scatter plot of the data in $\Sigma DAT$	→
SCATTER	Selects scatter plot	→
SCHUR	Computes the Schur decomposition of a square matrix	[[A]] → [[Q]] [[T]]
SCI	Sets Scientific display mode	n →
SCONJ	Conjugates the contents of a variable	'name' →
SDEV	Computes sample standard deviations of the data in $\Sigma DAT$	→ [ x <sub>1</sub> ... x <sub>m</sub> ]
SEND	Sends object to another Kermit device	'name' →
SEQ	Generates a list of results from repeated execution of an object	obj 'name' start end step → { list }
SERVER	Selects Kermit server mode	→
SF	Sets a system or user flag	±n →
SHOW	Resolves all name references	'symb <sub>1</sub> ' 'name' → 'symb <sub>2</sub> '
SIDENS	Intrinsic density of silicon as a function of temperature	T → density

SIGN	Sign of a number	$z_1 \rightarrow z_2$
SIN	Sine	$z \rightarrow \sin z$
SINH	Hyperbolic sine	$z \rightarrow \sinh z$
SINV	Inverts the contents of a variable	'name' $\rightarrow$
SIZE	Finds the dimensions of an object	object $\rightarrow n \text{ or } \{ \text{dim} \}$
SL	Shifts left by one bit	$\#n_1 \rightarrow \#n_2$
SLB	Shifts left by one byte	$\#n_1 \rightarrow \#n_2$
SLOPEFIELD	Selects slopefield plot	$\rightarrow$
SNEG	Negates the contents of a variable	'name' $\rightarrow$
SNRM	Computes the spectral norm of an array	[[matrix]] $\rightarrow$ spectral_norm
SOLVEQN	Places Equation Library equation(s) in solver and places corresponding picture in <i>PICT</i> if <i>PICT-option</i> is non-zero	subject title PICT-option $\rightarrow$
SORT	Sorts the elements of a list in ascending order	{ list } $\rightarrow$ { list }'
SPHERE	Sets polar/spherical mode	$\rightarrow$
SQ	Squares a number or matrix	$z \rightarrow z^2$
SR	Shifts right by one bit	$\#n_1 \rightarrow \#n_2$
SRAD	Computes spectral radius of a square matrix	[[matrix]] $\rightarrow$ spectral_radius
SRB	Shifts right by one byte	$\#n_1 \rightarrow \#n_2$
SRECV	Reads $n$ characters from the I/O port	$n \rightarrow$
START	Begins START ... NEXT or START ... STEP	start end $\rightarrow$

STD	Sets Standard display mode	→	
STEP	Ends FOR ... STEP or START ... STEP	increment →	
STEQ	Stores into reserved variable <i>EQ</i>	object →	
STIME	Sets serial timeout	seconds →	
STO	Stores an object into a variable	object 'name' →	
STOALARM	Stores alarm in system alarm list	time →	alarm_number
STOF	Sets the system and user flags	{ #system #user } →	
STOKEYS	Makes multiple key assignments	{ obj <sub>1</sub> rc.p <sub>1</sub> ... } →	
STO+	Storage addition	object 'name' →	
STO-	Storage subtraction	object 'name' →	
STO*	Storage multiplication	object 'name' →	
STO/	Storage division	object 'name' →	
STO $\Sigma$	Stores into reserved variable $\Sigma DAT$	object →	
STR→	Evaluates the commands defined by a string	"string" →	
→STR	Converts an object to a string	object →	"string"
STREAM	Executes <i>object</i> using first two objects in <i>list</i> , then executes <i>object</i> using next object in <i>list</i> and previous result until list is exhausted	{ list } object →	object
STWS	Sets the binary integer wordsize	n →	
SUB	Extracts a portion of a list, string, or grob	{ list } start end →	{ list }'

SVD	Computes the singular value decomposition of a matrix	[[A]] →	[[U]] [[V]] [S]
SVL	Computes the singular values of a matrix	[[matrix A]] →	[vector S]
SWAP	Swaps the objects in levels 1 and 2	obj <sub>2</sub> obj <sub>1</sub> →	obj <sub>1</sub> obj <sub>2</sub>
SYSEVAL	Executes a system object specified by address	#n →	
→TAG	Tags an object with another object	object "tag" →	:tag:object
TAIL	Returns a list less its first object	{ list } →	{ list }'
TAN	Tangent	z →	tan z
TANH	Hyperbolic tangent	z →	tanh z
TAYLR	Computes a Taylor series approximation	'symb <sub>1</sub> ' 'name' degree →	'symb <sub>2</sub> '
TDELTA	Calculates temperature difference	T <sub>1</sub> T <sub>2</sub> →	difference
TEACH	Creates example directory in the VAR menu	→	
TEXT	Selects the stack display	→	
THEN	Begins true-clause of IF, CASE, or IFERR structures	T/F →	
TICKS	Returns time in binary integer clock ticks (8192/sec)	→	#n
TIME	Returns current time using 24 hour format	→	HH.MMSS
→TIME	Sets the system time using 24 hour format	HH.MMSS →	
TINC	Adds temperature increment	T <sub>1</sub> increment →	T <sub>2</sub>
TLINE	Toggles pixels on a straight line	{ #x <sub>1</sub> #y <sub>1</sub> } { #x <sub>2</sub> #y <sub>2</sub> } →	
TMENU	Displays temporary menu	mm.pp →	
TOT	Sums the columns in $\Sigma DAT$	→	[ x <sub>1</sub> ... x <sub>m</sub> ]



TRACE	Computes the trace (sum of diagonal elements) of a matrix	[[matrix]]	→	trace
TRANSIO	Selects character translation mode	n	→	
TRN	Transposes a matrix	[[matrix <sub>1</sub> ]]	→	[[matrix <sub>2</sub> ]]
TRNC	Truncates a number to $n$ digits	$z_1$ n	→	$z_2$
TRUTH	Selects truth plot		→	
TSTR	Converts date and time numbers to string form	date time	→	“string”
TVARS	Lists the variables of specified type in the current directory	type	→	{ names }
TVM	Displays the TVM menu		→	
TVMBEG	Sets Begin payment mode		→	
TVMEND	Sets End payment mode		→	
TVMROOT	Solves for TVM variable	‘name’	→	value
TYPE	Returns the type of an object (see <i>Object Types</i> )	object	→	type
UBASE	Converts unit object to SI base units	x_units	→	y_base-units
UFACT	Factors specified compound unit	$x\_units_1$ $y\_units_2$	→	$x\_units_2 * units_3$
→UNIT	Combines a number and unit object to create a new unit object	x y_units	→	x_units
UNTIL	Begins test-clause of DO ... UNTIL ... END		→	
UPDIR	Makes parent directory the current directory		→	
UTPC	Upper-tail Chi-Square distribution	d.o.f. x	→	utpc(d.o.f.,x)

UTPF	Upper-tail F-distribution	d.o.f. <sub>1</sub> d.o.f. <sub>2</sub> x →	utpf(d.o.f. <sub>1</sub> ,d.o.f. <sub>2</sub> ,x)
UTPN	Upper-tail normal distribution	mean variance x →	uptn(mean,variance,x)
UTPT	Upper-tail Student's t-distribution	d.o.f. x →	utpt(d.o.f.,x)
UVAL	Returns scalar portion of unit object	x_unit →	x
→V2	Combines two real numbers into 2D vector or complex number	x y →	[ x y ]
→V3	Combines three real numbers into 3D vector	x y z →	[ x y z ]
V→	Separates a 2 or 3 element vector	[ x y z ] →	x y z
VAR	Sample variances of $\Sigma DAT$ data	→	[ x <sub>1</sub> ... x <sub>n</sub> ]
VARS	Returns list of variables in the current directory	→	{ names }
VERSION	Returns operating system version and copyright msg	→	"version" "copyright"
VTYP	Returns the type of an object in named variable	'name' →	type
*W	Multiplies the horizontal plot scale by specified factor (alter <i>PPAR</i> )	x →	
WAIT	Pauses program execution for <i>n</i> seconds or waits for a key ( <i>n</i> =0)	n →	rc.p ( <i>n</i> =0)
WHILE	Begins WHILE ... REPEAT ... END structure	→	
WIREFRAME	Selects wireframe plot type	→	
WSLOG	Returns four strings indicating time, date, and source of four most recent system halts	→	"string <sub>1</sub> " ... "string <sub>4</sub> "
XCOL	Specifies $\Sigma DAT$ column as the independent variable	x-column →	

XMIT	Sends string through I/O port without Kermit	“string” →	1 or “unsent” 0
XOR	Logical or binary XOR	#n <sub>1</sub> #n <sub>2</sub> →	#n <sub>3</sub>
XPON	Returns the exponent of a number	x →	n
XRECV	Receive an object using Xmodem protocol	‘name’ →	
XRNG	Specifies x-axis plotting range	x <sub>min</sub> x <sub>max</sub> →	
XROOT	Returns x <sup>th</sup> root of y	y x →	$\sqrt[x]{y}$
XSEND	Sends an object using Xmodem protocol	‘name’ →	
XVOL	Sets the width of the 3D plotting volume	x <sub>left</sub> x <sub>right</sub> →	
XXRNG	Sets width of 3D target mapping range for gridmap and parsurface plots	x <sub>min</sub> x <sub>max</sub> →	
YCOL	Specifies a $\Sigma DAT$ column as the dependent variable	y-column →	
YRNG	Specifies y-axis plotting range	y <sub>min</sub> y <sub>max</sub> →	
YSLICE	Selects yslice plot type	→	
YVOL	Sets the depth of the 3D plotting volume	y <sub>near</sub> y <sub>far</sub> →	
YYRNG	Sets the depth of 3D target mapping range for gridmap and parsurface plots	y <sub>min</sub> y <sub>max</sub> →	
ZFACTOR	Calculates gas compressibility factor Z	T <sub>r</sub> P <sub>r</sub> →	Z
ZVOL	Sets the height of the 3D plotting volume	z <sub>low</sub> z <sub>high</sub> →	
$\sqrt{\phantom{x}}$	Square root	z →	$\sqrt{z}$

$\int$	Integral	lower-limit upper-limit 'integrand' 'name' $\rightarrow$	integral
$\partial$	Derivative	'symp <sub>1</sub> ' 'name' $\rightarrow$	'symp <sub>2</sub> '
$\pi$	Symbolic constant $\pi$	$\rightarrow$	' $\pi$ '
$\Sigma$	Summation	'index' initial-value final-value 'summand' $\rightarrow$	sum
$\Sigma X$	Sum of data in independent $\Sigma DAT$ column	$\rightarrow$	$\Sigma X_i$
$\Sigma X^2$	Sum of squares of data in independent $\Sigma DAT$ column	$\rightarrow$	$\Sigma X_i^2$
$\Sigma Y$	Sum of data in dependent $\Sigma DAT$ column	$\rightarrow$	$\Sigma Y_i$
$\Sigma Y^2$	Sum of squares of data in independent $\Sigma DAT$ column	$\rightarrow$	$\Sigma Y_i^2$
$\Sigma X*Y$	Sum of products of data in independent and dependent $\Sigma DAT$ columns	$\rightarrow$	$\Sigma X_i Y_i$
$\Sigma +$	Appends one or more data points to $\Sigma DAT$	x $\rightarrow$	
$\Sigma -$	Deletes last row from $\Sigma DAT$	$\rightarrow$	x
$<$	Less-than comparison	x y $\rightarrow$	T/F
$>$	Greater-than comparison	x y $\rightarrow$	T/F
$\leq$	Less-than-or-equal comparison	x y $\rightarrow$	T/F
$\geq$	Greater-than-or-equal	x y $\rightarrow$	T/F
$\neq$	Not-equal comparison	x y $\rightarrow$	T/F
$=$	Equal comparison	$z_1 \ z_2 \rightarrow$	' $z_1=z_2$ '
$==$	Logical equality comparison	x y $\rightarrow$	T/F

→	Assigns local variable(s)	$\text{obj}_1 \dots \text{obj}_n \rightarrow$
+	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	$z_1 \ z_2 \rightarrow z_1/z_2$
^	Raises a number to a power	$z_1 \ z_2 \rightarrow z_1^{z_2}$
!	Factorial or gamma function	$n \rightarrow n!$
	Substitutes symbolics for names in a symbolic expression	$\text{'symb}_{\text{old}}' \ \{ \text{name}_1 \ \text{symb}_1 \dots \} \rightarrow \text{'symb}_{\text{new}}'$
%	Percent	$x \ y \rightarrow xy/100$
%CH	Percent change	$x \ y \rightarrow 100(y-x)/x$
%T	Percent total	$x \ y \rightarrow 100y/x$

## Going Further

56 Do you want to know more about your HP 48? *The HP 48 Handbook* is the ultimate reference for people who want to get the most from the HP 48 calculator. The *Handbook* contains chapters that cover all aspects of programming, user interface design, graphics, data transfer, memory management, custom menus, and much more. Additional reference tables cover the equation library, constants, key location codes, units, and much more. Many example programs are included to demonstrate various programming and graphics techniques. The command reference lists *every* stack diagram, keystroke access, characteristics, and related flags for every command. Do you want to place graphics in the stack display? Perhaps work with extended precision real numbers? The System Programming chapter introduces the magical world of internal unnamed objects accessible through the SYSEVAL command.

*The HP 48 Handbook* is the HP 48 reference book you can't live without!

To order a copy of *The HP 48 Handbook*, contact your calculator dealer or send a check (or postal money order) for \$20 in U.S. dollars drawn on a U.S. bank (overseas bank drafts are not accepted) to:

Armstrong Publishing Co.  
1050 Springhill Drive  
Albany OR 97321 USA



# The HP 48 Pocket Book

Flags	2
Operator Precedence	7
Object Types	8
Reserved Variables	9
Symbolic Constants	11
Program Branch Structures	12
System Operations	13
Unit Prefixes	14
Menu Numbers	15
Messages	18
Character Codes	22
Command Reference	24
Going Further	56

*The HP 48 Pocket Book* goes where you go – providing a concise collection of handy tables covering many aspects of HP 48 operations and a 32 page command reference listing the syntax for all commands.

ISBN 1 - 879828-05-7