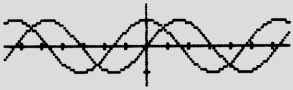
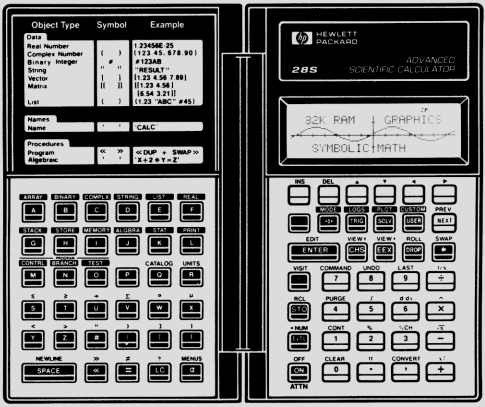
THE HP-28 POCKET BOOK



```
« 'SIN(X)=COS(X)'
STEQ RAD 'PPAR'
PURGE CLLCD DRAW
»
```



Copyright © James Donnelly 1988

HP PART NO. 92222G Printed in U.S.A. 10/88

PLOT AND SOLVE EXAMPLE

Step	Type	Press	Comments
1	'X^2-3*X-3'	■ PLOT STEQ	Establish the current equation
2	'PPAR'	■ PURGE	Purge existing plot parameters
3	(-3.5,-6)	PMIN	Set lower-left plot coordinates
4	(6.5,16)	PMAX DRAW	Set upper-right coordinates and draw graph
5	▷ ◁ △ ▽		Move the cursor to digitize the left root
6		INS	Put the cursor coordinates on the stack
7	▷ ◁ △ ▽	INS	Digitize the right root the same way
8		ATTN	Restore the stack display
9		SOLV	Activate the SOLVE menu
10		SOLVR	Activate the Solver variables menu
11		X	Enter the digitized guess for the right root
12		■ X	Solve for the right root ☞ 3.79128784747
13		■ SWAP	Exchange levels 1 and 2
14		X	Enter the digitized guess for the left root
15		■ X	Solve for the left root ☞ -.791287847477

DIFFERENTIATION EXAMPLE

Continuing from the previous example, find the minimum of the curve by differentiating and solving for the root.

Step	Type	Press	Comments
16	■ PLOT	RCEQ ENTER	Recall two copies of the equation to the stack
17	'X'	PURGE	Formalize X for a symbolic solution
18	'X'	ENTER ■ d/dx	Differentiate with respect to X ☞ '2*X-3'
19		STEQ	Establish the new current equation for plotting
20		DRAW	Plot the new function
21	▷ ◁ △ ▽	INS	Digitize the left side of root
22	▷ ◁ △ ▽	INS	Digitize the right side of root
23		ATTN	Restore the stack display
24	2	■ LIST →LIST	Combine the two digitized points into a list
25		SOLV SOLVR	Activate the Solver variables menu
26		X	Enter the two gueses for the root into X
27		■ X	Solve for the root ☞ X=1.5
28		■ SWAP	Move the original equation back to level 1
29		EVAL	Evaluate to find the minimum ☞ -5.25

INTEGRATION EXAMPLES

Symbolic Integration

3:	'X^3+4*X+8'	Integrand
2:	'X'	Variable of integration
1:	3	Degree of polynomial
		■ ∫ ☞
1:	'8*X+2*X^2+.25*X^4'	Integrated expression

Numerical Integration

3:	'X^3+4*X+8'	Integrand
2:	{ X 1 3 }	Variable, lower limit, upper limit
1:	.00001	Accuracy factor
		■ ∫ ☞
2:	52	Computed value of integral
1:	5.1992166102E-4	Error value

UNIT EXAMPLES

Simple Units

3:	80	Quantity of unit identified in level 2
2:	'°F'	"From" unit
1:	'°C'	"To" unit
		■ CONVERT ☞
2:	26.6666666667	Result quantity
1:	'°C'	Result units

Compound Units

3:	1	Quantity of unit identified in level 2
2:	"b*Mpc/s"	Barn-megaparsecs per second
1:	"gal/h"	Gallons per hour
		■ CONVERT ☞
2:	2.93453978138	
1:	"gal/h"	Gallons per hour

UNIT PREFIXES

HP-28 Symbol	Prefix	Number	Name
E	exa	+18	quintillion
P	peta	+15	quadrillion
T	tera	+12	trillion
G	giga	+9	billion
M	mega	+6	million
k,K	kilo	+3	thousand
h,H	hecto	+2	hundred
D	deka	+1	ten
d	deci	-1	tenth
c	centi	-2	hundredth
m	milli	-3	thousandth
μ	micro	-6	millionth
n	nano	-9	billionth
p	pico	-12	trillionth
f	femto	-15	quadrillionth
a	atto	-18	quintillionth

RESERVED NAMES

Name	Description
EQ	Current equation for root-finding and plotting
PPAR	Plot parameters
der...	User-defined derivatives begin with *der*
n1, n2, ...	Arbitrary integers created by ISOL
s1, s2, ...	Arbitrary signs created by ISOL and QUAD
ΣDAT	Current statistical array
ΣPAR	Statistics parameters

TRUTH TABLE

arg_1	arg_2	arg_1 AND arg_2	arg_1 OR arg_2	arg_1 XOR arg_2	NOT arg_1
1	1	1	1	0	0
1	0	0	1	1	0
0	1	0	1	1	1
0	0	0	0	0	1
"s_1"	"s_2"	"s_3"	"s_4"	"s_5"	"s_6"

NOTE: String arguments must have the same length.

KEY STRINGS

Key	String	Key	String
△	"UP"	INS	"INS"
▽	"DOWN"	DEL	"DEL"
◁	"LEFT"	LC	"l"
▷	"RIGHT"	◁△▽▷	"CURSOR"
↵	"BACK"	SPACE	" "
×	"*"	■	"SHIFT"
÷	"/"		

KEY returns either a 0 flag to level 1 or a 1 flag to level 1 and a string representing the oldest key in the buffer to level 2. Keys not listed here return their keyboard nomenclature.

SYSTEM OPERATIONS

Name	Keystrokes
Attention	[ON]
Print Display (28S only)	[ON] [L]
Contrast Control	[ON] [+] or [ON] [−]
System Halt	[ON] [△]
Memory Reset	[ON] [INS] [▷]
Repeating Test	[ON] [◁]
Keyboard Test (28S only)	[ON] [NEXT]
Cancel System Operation	[ON] [DEL]

INTERACTIVE PLOT MODE

Key	Action
[INS]	Return cursor coordinates to stack.
[DEL]	Return cursor coordinates to stack (28C).
[DEL]	Return LCD→ string to stack (28S).
[◁] [▷] [△] [▽]	Move the cursor. (*Press the shift key first to move to the edge.*)
[◁◆▷]	Display cursor coordinates (28S only).
[ON] [L]	Print the display (28S only).
[ON]	End interactive plot mode.

SYSEVAL ENTRIES

Description	28C 1BB	28C 1CC	28S 2BB
ROM version	A	A	A
Clock	123E	1266	11CA
Command stack off	F3B8	F3CE	
Command stack on	F39A	F3B0	
LAST off	F3F4	F40A	
LAST on	F3D6	F3EC	
UNDO off	F44E	F464	
UNDO on	F412	F428	
Multi-line display off	F4B2	F4C8	
Multi-line display on	F48F	F4A5	
Command stack toggle			21E02
LAST toggle			21E2F
UNDO toggle			21E5C
Multi-line display toggle			21EA7

HP-28S MENUS

n	Standard Menu	*n*	Standard Menu
1	ARRAY	13	PRGM CONTRL
2	BINARY	14	PRGM BRANCH
3	COMPLEX	15	PRGM TEST
4	STRING	16	MODE
5	LIST	17	LOGS
6	REAL	18	PLOT
7	STACK	19	CUSTOM
8	STORE	20	Cursor
9	MEMORY	21	TRIG
10	ALGEBRA	22	SOLVE
11	STAT	23	USER
12	PRINT	24	Solver

OBJECT TYPES

Type	Object
0	Real number
1	Complex number
2	String
3	Real vector or matrix
4	Complex vector or matrix
5	List
6	Global name
7	Local name
8	Program
9	Algebraic
10	Binary integer

OPERATOR PRECEDENCE

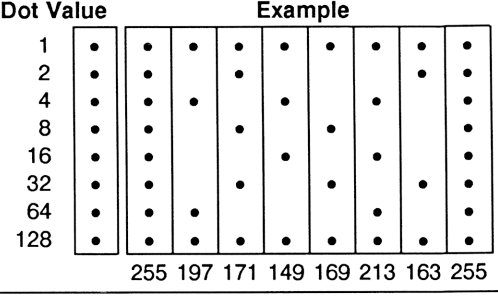
Level	Operation
1	Expressions within parentheses
2	Functions
3	Power (^) and root ($\sqrt{\ }$)
4	Negate (−), multiply (*), divide (/)
5	Add (+) and subtract (−)
6	Relational operators (==, ≠, <, >, ≤, ≥)
7	AND and NOT
8	OR and XOR
9	=

PRINTER CONTROL

Printer Mode	Control Codes*	
Carriage right	4	
Carriage return/LF	10	
Column graphics	27	n $C_1 \ldots C_n$ †
Underline off‡	27	250
Underline on	27	251
Single wide print‡	27	252
Double wide print	27	253
Self-test	27	254
Reset	27	255

*Decimal value † $1 \leq n \leq 166$ ‡Default mode

COLUMN GRAPHICS



28S LCD: <<255 197 171 149 169 213 163 255 "" 1 8 START SWAP CHR SWAP + NEXT CLLCD →LCD >>

PRINTER: <<27 8 255 197 171 149 169 213 163 255 "" 1 10 START SWAP CHR SWAP + NEXT PR1 >>

PPAR AND ΣPAR PARAMETERS

PPAR ☞ { (χ_{min}, ψ_{min}) (χ_{max}, ψ_{max}) indep-var resolution $(\chi_{axis}, \psi_{axis})$ }

Parameter	Affected By Commands	Description
(χ_{min}, ψ_{min})	PMIN CENTR *H *W SCLΣ	Lower-left pixel coordinates
(χ_{max}, ψ_{max})	PMAX CENTR *H *W SCLΣ	Upper-right pixel coordinates
indep-var	INDEP	Global name for horizontal axis
resolution	RES	Real positive integer for point spacing
$(\chi_{axis}, \psi_{axis})$	AXES	Axes intersection coordinates

ΣPAR ☞ { col_1 col_2 intercept slope }

Parameter	Affected By Command	Description
col_1	COLΣ	Independent data column
col_2	COLΣ	Dependent data column
intercept	LR	Intercept of most recent LR calculation
slope	LR	Slope of most recent LR calculation

ΣDAT STATISTICS MATRIX

x Σ+ ☞ Append one data point with one coordinate value

$[\ x_{11}\ x_{12}\ \ldots\ x_{1m}\ ]$ Σ+ ☞ Append one data point with *m* coordinate values

$[[\ x_{21}\ x_{22}\ \ldots\ x_{2m}\ ]$
$\ldots$
$[\ x_{n1}\ x_{n2}\ \ldots\ x_{nm}\ ]]$ Σ+ ☞ Append *n* data points with *m* coordinate values

Σ- ☞ Reverses the effect of the last Σ+

Data Point	Coordinate Number					
	1	2	3	4	...	m
1	x_{11}	x_{12}	x_{13}	x_{14}	...	x_{1m}
2	x_{21}	x_{22}	x_{23}	x_{24}	...	x_{2m}
...	...	...	...	...	...	...
n	x_{n1}	x_{n2}	x_{n3}	x_{n4}	...	x_{nm}

CHARACTER CODES

NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR
0	·	32		64	@	96	`	128		160	¿	192	à	224	Á
1	·	33	!	65	A	97	a	129	÷	161	A	193	è	225	Ã
2	·	34	"	66	B	98	b	130	×	162	A	194	ò	226	ã
3	·	35	#	67	C	99	c	131	∫	163	É	195	ù	227	Đ
4	·	36	$	68	D	100	d	132	∫	164	É	196	ä	228	đ
5	·	37	%	69	E	101	e	133	Σ	165	É	197	ë	229	Ĩ
6	·	38	&	70	F	102	f	134	▶	166	Ĩ	198	ö	230	ĩ
7	·	39	'	71	G	103	g	135	π	167	Ĩ	199	ü	231	Ó
8	·	40	(	72	H	104	h	136	∂	168	'	200	à	232	Ò
9	·	41	)	73	I	105	i	137	≤	169	`	201	è	233	Ô
10	·	42	*	74	J	106	j	138	≥	170	^	202	ò	234	Õ
11	·	43	+	75	K	107	k	139	≠	171	¨	203	ù	235	Ś
12	·	44	,	76	L	108	l	140	α	172	~	204	ä	236	Ŷ
13	·	45	-	77	M	109	m	141	→	173	Ù	205	ë	237	Ù
14	·	46	.	78	N	110	n	142	←	174	Ù	206	ö	238	Ỳ
15	·	47	/	79	O	111	o	143	μ	175	£	207	ü	239	ỳ

NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR	NUM	CHR
16	•	48	0	80	P	112	p	144	┘	176	─	208	Ã	240	Þ
17	•	49	1	81	Q	113	q	145	°	177	ÿ	209	ì	241	Ð
18	•	50	2	82	R	114	r	146	«	178	ÿ	210	Ø	242	•
19	•	51	3	83	S	115	s	147	»	179	■	211	Ë	243	µ
20	•	52	4	84	T	116	t	148	├	180	Ç	212	ª	244	¶
21	•	53	5	85	U	117	u	149	│	181	ç	213	í	245	¼
22	•	54	6	86	V	118	v	150	ª	182	Ñ	214	º	246	¯
23	•	55	7	87	W	119	w	151	º	183	ñ	215	¾	247	¼
24	•	56	8	88	X	120	x	152	ª	184	¡	216	Ã	248	½
25	•	57	9	89	Y	121	y	153	¡	185	¿	217	ì	249	¾
26	•	58	:	90	Z	122	z	154	¡	186	®	218	Ò	250	º
27	•	59	;	91	[	123	{	155	■	187	ƒ	219	Ù	251	«
28	•	60	<	92	\	124	|	156	¡	188	¥	220	É	252	■
29	•	61	=	93	]	125	}	157	¡	189	§	221	ì	253	»
30	•	62	>	94	^	126	~	158	k	190	ƒ	222	ß	254	±
31	•	63	?	95	_	127	▒	159	n	191	¢	223	Ô	255	

NOTE: 148 & 160 were blank on early printers.

ERRORS BY NUMBER

Hex	Decimal	Message
General Operation Errors		
001	001	Insufficient Memory
003	003	Undefined Local Name
005	005	Memory Lost
006	006	Power Lost (28S)
101	257	No Room for UNDO
102	258	Can't Edit CHR(0)
103	259	Improper User Function
104	260	No Current Equation
105	261	No Room to ENTER
106	262	Syntax Error
11E	286	Invalid PPAR
11F	287	Non-real Result
120	288	Unable to Isolate
121	289	HALT Not Allowed
124	292	UNDO Disabled
125	293	Command Stack Disabled
128	296	Wrong Argument Count
129	297	Circular Reference
12A	298	Directory Not Allowed (28S)
12B	299	Non-Empty Directory (28S)

ERRORS BY NUMBER

Hex	Decimal	Message
Stack Operation Errors		
201	513	Too Few Arguments
202	514	Bad Argument Type
203	515	Bad Argument Value
204	516	Undefined Name
205	517	LAST Disabled
Real Number Errors		
301	769	Positive Underflow
302	770	Negative Underflow
303	771	Overflow
304	772	Undefined Result
305	773	Infinite Result
Array Operation Errors		
501	1281	Invalid Dimension
Statistics Operation Errors		
601	1537	Invalid ΣDAT
602	1538	Nonexistent ΣDAT
603	1539	Insufficient Σ Data
604	1540	Invalid ΣPAR
Root-finder Errors		
A01	2561	Bad Guess(es)
A02	2562	Constant?
Unit Conversion Errors		
B01	2817	Invalid Unit String
B02	2818	Inconsistent Units

ERRORS BY MESSAGE

Message	Hex	Decimal
Bad Argument Type	202	514
Bad Argument Value	203	515
Bad Guess(es)	A01	2561
Can't Edit CHR(0)	102	258
Circular Reference	129	297
Command Stack Disabled	125	293
Constant?	A02	2562
Constant Equation	Status	
Directory Not Allowed (28S)	12A	298
Extremum	Status	
HALT Not Allowed	121	289
Improper User Function	103	259
Inconsistent Units	B02	2818
Infinite Result	305	773
Insufficient Memory	001	001
Insufficient Σ Data	603	1539
Interrupted	Status	
Invalid Dimension	501	1281
Invalid PPAR	11E	286
Invalid Unit String	B01	2817
Invalid ΣDAT	601	1537
Invalid ΣPAR	604	1540
LAST Disabled	205	517
Low Memory!	Status	
Memory Lost	005	005

ERRORS BY MESSAGE

Message	Hex	Decimal
Negative Underflow	302	770
No Current Equation	104	260
Nonexistent ΣDAT	602	1538
Non-Empty Directory (28S)	12B	299
Non-real Result	11F	287
No Room for UNDO	101	257
No Room to ENTER	105	261
No Room to Show Stack	Status	
name Not in Equation	Status	
Out of Memory		
Overflow	303	771
Positive Underflow	301	769
Power Lost (28S)	006	006
Sign Reversal	Status	
Syntax Error	106	262
Too Few Arguments	201	513
Unable to Isolate	120	288
Undefined Local Name	003	003
Undefined Name	204	516
Undefined Result	304	772
UNDO Disabled	124	292
Using *name*	Status	
Wrong Argument Count	128	296
Zero	Status	

FLAGS

Flag	Description	Set	Clear	Default
1-30	User flags			Clear
31	LAST arguments mode	LAST arguments on	LAST arguments off	On
32	Printer trace mode	Printer trace on	Printer trace off	Off
33	Auto CR mode	Auto CR off	Auto CR on	On
34	Solution mode	Principal value	General solution	Gen soln
35	Constants mode	Symbolic	Numerical	Symbolic
36	Results mode	Symbolic	Numerical	Symbolic
37-42	Binary integer wordsize $n+1$: $0 \leq n \leq 63$, flag 42 is most significant bit			64
43,44	**Binary Integer Base**	**43**	**44**	DEC
	DEC	Clear	Clear	
	BIN	Clear	Set	
	OCT	Set	Clear	
	HEX	Set	Set	
45	Multi-line display	On	Off	On
46	Index wrap indicator	Wrap occurred	No wrap occurred	No wrap

Flag	Description	Set	Clear	Default
47	Double-space print	On	Off	Off
48	Radix mark	'MOD(X.Y)' 1234,56	'MOD(X,Y)' 1234.56	1234.56
49,50	**Number Format**	**49**	**50**	STD
	STD	Clear	Clear	
	SCI	Clear	Set	
	FIX	Set	Clear	
	ENG	Set	Set	
51	Beeper mode	Off	On	On
52	Print speed	Faster (AC)	Normal (Battery)	Normal
53-56	Number of decimal digits n: $0 \leq n \leq 11$, flag 56 is most significant bit			0
57	Underflow action	Error	Return 0, set 61 or 62	Clear
58	Overflow action	Error	Return ±MAXR, set 63	Clear
59	Infinite result action	Error	Return ±MAXR, set 64	Error
60	Angle mode	Radians	Degrees	Degrees
61	Underflow− indicator	Exception	No Exception	Clear
62	Underflow+ indicator	Exception	No Exception	Clear
63	Overflow indicator	Exception	No Exception	Clear
64	Infinite result indicator	Exception	No Exception	Clear

MEMORY USAGE

Object/Structure	Size (bytes)
Real integers -9 to +9	2.5
π, *e*, *i*, MINR, MAXR	2.5
Real number	10.5
Complex number	18.5
Binary integer	13
Object in variable	4.5 + *1 byte per character of name* + *object size*
String	5 + *number of characters*
Real vector	12.5 + 8 × *no. of elements*
Complex vector	12.5 + 16 × *no. of elements*
Real matrix	15.5 + 8 × *no. of elements*
Complex matrix	15.5 + 16 × *no. of elements*
List, algebraic	5 + *included objects*
Unquoted name	3.5 + *number of characters*
Quoted name	8.5 + *number of characters*
Command	2.5
Program	12.5 + *included objects*
`IF ... THEN ... END`	17.5
`IFERR ... THEN ... END`	17.5
`IF ... THEN ... ELSE ... END`	25
`IFERR ... THEN ... ELSE ... END`	25
`DO ... UNTIL ... END`	7.5
`WHILE ... REPEAT ... END`	12.5
`START/FOR ... NEXT/STEP`	5
→ ... << ... >>	7.5
→ ... ' ... '	7.5

COMMAND REFERENCE

Term	Description	Term	Description
obj	Any object	"string"	Character string
x or y	Real number	[array]	Real or complex vector or matrix
z	Real or complex number		
m or n	Positive integer real number	[vector]	Real or complex vector
hms	Real number in hours-minutes-seconds format	[matrix]	Real or complex matrix
		{list}	List of objects
flag	Real number, 0 (false) or non-zero (true)	index	Real number specifying an element in a list or array
(x,y)	Complex number in rectangular form		
(r,θ)	Complex number in polar form	{dim}	List of 1 or 2 array dimensions
#n	Binary integer	'name'	Global name or local name
'symb'	Expression, equation, or name treated as an algebraic	'global'	Global name

EXAMPLE

Name	Description	Level 2	Level 1 ☞	Level 2	Level 1
SWAP	Swaps the objects in levels 1 and 2	obj_1	obj_2	obj_2	obj_1

Name	Description	In ☞	Out
ABORT	Aborts program execution		
ABS	Absolute value	z	|z|
ACOS	Arc cosine	z	acos z
ACOSH	Inverse hyperbolic cosine	z	acosh z
ALOG	Antilogarithm	z	10^z
AND	Logical or binary AND	$\#n_1$ $\#n_2$	$\#n_3$
ARG	Argument	z	θ
ARRY→	Separate array into individual elements	[vector]	$x_1 \ldots x_n$ {n}
		[matrix]	$x_{11} \ldots x_{nm}$ {n m}
ASIN	Arc sine	z	asin z
ASINH	Inverse hyperbolic sine	z	asinh z
ASR	Arithmetic shift right	$\#n_1$	$\#n_2$
ATAN	Arc tangent	z	atan z
ATANH	Inverse hyperbolic tangent	z	atanh z
AXES	Sets intersection of axes	(x,y)	
BEEP	Sounds a beep	*Hz secs*	
BIN	Sets binary base		
B→R	Binary-to-real conversion	#n	n
CEIL	Next greater integer	x	n

Name	Description	In ☞	Out
CENTR	Sets center of plot display	(x,y)	
CF	Clears a user flag	n	
CHR	Makes a one-character string	n	"string"
CLEAR	Clears the stack	n	
CLLCD	Clears the display		
CLMF	Clears the system message flag		
CLUSR	Clears all user variables		
CLΣ	Purges the statistics matrix		
CNRM	Computes a column norm	[array]	*column-norm*
COLCT	Collects like terms	'symb$_1$'	'symb$_2$'
COLΣ	Selects statistics matrix columns	n_1 n_2	
COMB	Combinations (28S)	n m	$C_{n,m}$
CON	Creates a constant matrix	{dim} z	[array]
CONJ	Complex conjugate	[array]	[array]
CONVERT	Performs a unit conversion	x "*old*" "*new*"	y "*new*"
CORR	Correlation coefficient		*correlation*
COS	Cosine	z	cos z
COSH	Hyperbolic cosine	z	cosh z

Name	Description	In ☞	Out
COV	Covariance		*covariance*
CR	Prints a carriage-right		
CRDIR	Creates a directory (28S)	'name'	
CROSS	Cross product	[**A**] [**B**]	[**A** × **B**]
C→R	Complex-to-real conversion	(x,y)	x y
DEC	Sets decimal base		
DEG	Sets degrees mode		
DEPTH	Counts the objects on the stack		n
DET	Determinant of a matrix	[matrix]	determinant
DGTIZ	Activates interactive plot mode (28S)		
DISP	Displays an object	obj n	
DO	Begins DO ... UNTIL ... END	**DO** *loop-clause* **UNTIL** *test-clause* **END**	
DOT	Dot product of two vectors	[**A**] [**B**]	x
DRAW	Creates a mathematical function plot		
DRAX	Draws axes		
DROP	Drops one object from the stack	obj	
DROPN	Drops $n+1$ objects from the stack (excluding n)	obj_n ... obj_1 n	
DROP2	Drops two objects from the stack	obj_2 ... obj_1	

Name	Description	In ☞	Out
DRWΣ	Creates a statistics scatter plot		
DUP	Duplicates one object on the stack	obj	obj obj
DUPN	DUPs *n* objects on the stack (excluding *n*)	obj_n ... obj_1 n	obj_n ... obj_1 obj_n ... obj_1
DUP2	Duplicates two objects on the stack	obj_1 obj_2	obj_1 obj_2 obj_1 obj_2
D→R	Degrees-to-radians conversion	x	$(\pi/180)x$
e	Symbolic constant *e*		2.71828182846
ELSE	Begins ELSE clause		
END	Ends program structures		
ENG	Sets engineering display format	n	
ERRM	Returns the last error message		"*error message*"
ERRN	Returns the last error number		#n
EVAL	Evaluates an object	obj	
EXGET	Gets a subexpression	'$symb_1$' n	'$symb_2$'
EXP	Natural exponential	z	exp z
EXPAN	Expands an algebraic	'$symb_1$'	'$symb_2$'
EXPM	Natural exponential minus 1	x	$\exp(x)-1$
EXSUB	Substitutes a subexpression	'$symb_1$' n '$symb_2$'	'$symb_3$'
FACT	Factorial or gamma function	n *or* x	n! *or* $\Gamma(x+1)$

Name	Description	In ☞	Out
FC?	Tests a user flag	n	flag
FC?C	Tests and clears a user flag	n	flag
FIX	Sets FIX display format	n	
FLOOR	Next smaller integer	x	n
FOR	Begins FOR ... NEXT	*start finish* **FOR** *name loop-clause* **NEXT**	
	or FOR ... STEP	*start finish* **FOR** *name loop-clause increment* **STEP**	
FORM	Changes the form of an algebraic	'symb$_1$'	'symb$_2$'
FP	Fractional part	x	y
FS?	Tests a user flag	n	flag
FS?C	Tests and clears a user flag	n	flag
GET	Gets an element from an object	[array] index	z
GETI	Gets an element from an object and increments the index	[array] index$_1$	[array] index$_2$ z
HALT	Suspends program execution		
HEX	Sets hexadecimal base		
HMS+	Adds in HMS format	hms$_1$ hms$_2$	hms$_1$ + hms$_2$
HMS−	Subtracts in HMS format	hms$_1$ hms$_2$	hms$_1$ − hms$_2$
HMS→	Converts from HMS format	hms	x
HOME	Selects the HOME directory (28S)		

Name	Description	In ☞	Out
i	Symbolic constant *i*		(0,1)
IDN	Creates an identity matrix	n	[*real-identity-matrix*]
IF	Begins IF ... THEN ... END	**IF** *test-clause* **THEN** *true-clause* **END**	
	or IF ... THEN ... ELSE ... END	**IF** *test-clause* **THEN** *true-clause* **ELSE** *false-clause* **END**	
IFERR	Begins IFERR ... THEN ... END	**IFERR** *test-clause* **THEN** *true-clause* **END**	
	or IFERR ... THEN ... ELSE ... END	**IFERR** *test-clause* **THEN** *true-clause* **ELSE** *false-clause* **END**	
IFT	IF ... THEN ... END cmd	flag obj	
IFTE	IF ... THEN ... ELSE ... END cmd	flag *true-obj* *false-obj*	
IM	Returns imaginary part of a number	(x,y)	y
	or array	[$array_1$]	[$array_2$]
INDEP	Selects plot independent variable	'global'	
INV	Inverse (reciprocal)	z	1/z
IP	Integer part	x	n
ISOL	Isolates a variable in an equation	'$symb_1$ 'global'	$symb_2$
KEY	Returns 0 if no key in buffer		0
	otherwise returns 1 and key		"string" 1
KILL	Aborts all suspended programs		
LAST	Returns LAST arguments (if saved)		*LAST-argument(s)*

Name	Description	In ☞	Out
LCD→	Returns LCD graphics string (28S)		"string"
LIST→	Separates list into individual objects	$\{obj_1 \ldots obj_n\}$	$obj_1 \ldots obj_n$ n
LN	Natural logarithm	z	ln z
LNP1	Natural logarithm of (arg + 1)	x	ln(1+x)
LOG	Common (base 10) logarithm	z	log z
LR	Computes a linear regression		*intercept slope*
MANT	Returns the mantissa of a number	x	y
MAX	Returns the maximum of two numbers	x y	max(x,y)
MAXR	Symbolic constant maximum real		9.99999999999E499
MAXΣ	Finds the maximum coordinate		x
	values in the statistics matrix		$[x_1 \ldots x_m]$
MEAN	Computes statistical means		x
MEM	Returns available memory		x
MENU	Selects a numbered menu (28S)	n	
	Creates a custom menu (28S)	{ *names and cmds* }	
	Creates a custom input menu (28S)	{ STO *names* }	
MIN	Returns the minimum of two numbers	x y	min(x,y)
MINR	Symbolic constant minimum real		1.E-499

Name	Description	In ☞	Out
MINΣ	Finds the minimum coordinate values in the statistics matrix		x $[x_1 \ldots x_m]$
MOD	Modulo	x y	x mod y
NEG	Negates an argument	z	−z
NEXT	Ends FOR ... NEXT or START ... NEXT		
NOT	Logical or binary NOT	$\#n_1$	$\#n_2$
NUM	Returns character code	"string"	n
NΣ	Returns the number of data points in the statistics matrix		n
OBGET	Extracts an object from an algebraic	'symb' n	{ obj }
OBSUB	Substitutes an object into an algebraic	$'symb_1'$ n { obj }	$'symb_2'$
OCT	Sets octal base		
OR	Logical or binary OR	$\#n_1$ $\#n_2$	$\#n_3$
ORDER	Rearranges the user menu	{ *names* }	
OVER	Copies the object in level 2 into level 1	obj_2 obj_1	obj_2 obj_1 obj_2
PATH	Returns a list showing the current path (28S)		{ HOME *dir-names* }
PERM	Permutations (28S)	n m	$P_{n,m}$
PICK	Copies *n*th object into level 1 (excluding *n*)	$obj_n \ldots obj_1$ n	$obj_n \ldots obj_1$ obj_n

Name	Description	In ☞	Out
PIXEL	Turns on a display pixel	(x,y)	
PMAX	Sets the upper-right plot coordinates	(x,y)	
PMIN	Sets the lower-left plot coordinates	(x,y)	
POS	Finds a substring in a string or	"$string_1$" "$string_2$"	n
	finds an object in a list	{list} obj	n
PREDV	Predicted value	x	*predicted-value*
PRLCD	Prints an image of the display		
PRMD	Prints and displays current modes		
PRST	Prints the stack		
PRSTC	Prints the stack in compact format		
PRUSR	Prints a list of variables in the current directory		
PRVAR	Prints the name and contents of one	'global'	
	or more variables	{ $global_1$ $global_2$ ... }	
PR1	Prints the level 1 object	obj	obj
PURGE	Purges one	'global'	
	or more variables	{ $global_1$ $global_2$ ... }	
PUT	Replaces an element in an array or list	[$array_1$] index x	[$array_2$]

Name	Description	In ☞	Out
PUTI	Replaces an element in an array or list and increments the index	$[array_1]$ index x	$[array_2]$ $index_2$
P→R	Polar-to-rectangular conversion	(r,θ)	(x,y)
QUAD	Solves a quadratic polynomial	$'symb_1'$ 'global'	$'symb_2'$
RAD	Sets radians mode		
RAND	Returns a random number		x
RCEQ	Recalls the current equation		obj
RCL	Recalls the contents of a variable	'name'	obj
RCLF	Returns a binary integer representing the user flags		#n
RCLΣ	Recalls the current statistics matrix		obj
RCWS	Recalls the binary integer wordsize		n
RDM	Redimensions an array	$[array_1]$ { dim }	$[array_2]$
RDZ	Sets the random number seed	x	
RE	Returns the real part of a complex number or array	(x,y)	x
REPEAT	Part of WHILE ... REPEAT ... END	*flag*	
RES	Sets the plot resolution	n	

Name	Description	In ☞	Out
RL	Rotates left by one bit	$\#n_1$	$\#n_2$
RLB	Rotates left by one byte	$\#n_1$	$\#n_2$
RND	Rounds using real number mode	z_1	z_2
RNRM	Computes row norm of an array	[array]	*row-norm*
ROLL	Moves level $n+1$ object to level 1	obj_n ... obj_1 n	obj_{n-1} ... obj_1 obj_n
ROLLD	Moves the level 2 object to level n	obj_1 ... obj_n n	obj_n obj_1 ... obj_{n-1}
ROOT	Finds a numerical root	'symb' 'global' guess	root
ROT	Moves the level 3 object to level 1	obj_1 obj_2 obj_3	obj_2 obj_3 obj_1
RR	Rotates right by one bit	$\#n_1$	$\#n_2$
RRB	Rotates right by one byte	$\#n_1$	$\#n_2$
RSD	Computes a correction to the solution of a system of equations	[array **B**] [matrix **A**] [array **Z**]	[array **B-AZ**]
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	$(180/\pi)x$
R→P	Rectangular-to-polar conversion	(x,y)	(r,θ)
SAME	Tests two objects for equality	obj_1 obj_2	flag
SCI	Sets scientific display format	n	

Name	Description	In ☞	Out
SCLΣ	Auto-scales the plot parameters according to the statistical data		
SCONJ	Conjugates the contents of a variable	'global'	
SDEV	Computes standard deviation		x
SF	Sets a user flag	n	
SHOW	Resolves all references to a name implicit in an algebraic	'symb$_1$' 'global'	'symb$_2$'
SIGN	Sign of a number	z_1	z_2
SIN	Sine	z	sin z
SINH	Hyperbolic sine	z	sinh z
SINV	Inverts the contents of a variable	'global'	
SIZE	Finds the dimensions of a list, array, string, or algebraic	obj	n
SL	Shifts left by one bit	$\#n_1$	$\#n_2$
SLB	Shifts left by one byte	$\#n_1$	$\#n_2$
SNEG	Negates the contents of a variable	'global'	
SQ	Squares a number or matrix	z	z^2
SR	Shifts right by one bit	$\#n_1$	$\#n_2$

Name	Description	In ☞	Out
SRB	Shifts right by one byte	$\#n_1$	$\#n_2$
START	Begins START ... NEXT	*start finish* **START** *loop-clause* **NEXT**	
	or START ... STEP	*start finish* **START** *loop-clause increment* **STEP**	
STD	Sets standard display format		
STEP	Ends FOR ... STEP or START ... STEP	increment	
STEQ	Stores into reserved variable EQ	obj	
STO	Stores an object in a variable	obj 'name'	
STOF	Sets all user flags according to the value of a binary integer	#n	
STO*	Storage arithmetic multiply	z 'global'	
STO+	Storage arithmetic add	z 'global'	
STO−	Storage arithmetic subtract	z 'global'	
STO/	Storage arithmetic divide	z 'global'	
STOΣ	Stores into reserved variable ΣDAT	obj	
STR→	Parses and evaluates the commands defined by a string	"string"	
STWS	Sets the binary integer wordsize	n	
SUB	Extracts a portion of a list or string	"$string_1$" *start end*	"$string_2$"

Name	Description	In ☞	Out
SWAP	Swaps the objects in levels 1 and 2	obj_1 obj_2	obj_2 obj_1
SYSEVAL	Executes a system object	#n	
TAN	Tangent	z	tan z
TANH	Hyperbolic tangent	z	tanh z
TAYLR	Computes a Taylor series approx.	'$symb_1$' 'global' n	'$symb_2$'
THEN	Begins THEN clause	*flag*	
TOT	Sums the coordinate values in the statistics matrix		x
TRN	Transposes a matrix	[$matrix_1$]	[$matrix_2$]
TYPE	Returns the type of an object	obj	type
UNTIL	Part of DO ... UNTIL ... END		
UTPC	Upper-tail Chi-Square distribuion	n x	utpc(n,x)
UTPF	Upper-tail F-distribution	n_1 n_2 x	utpf(n_1,n_2,x)
UTPN	Upper-tail normal distribution	m v x	utpn(m, v, x)
UTPT	Upper-tail t-distribution	n x	utpt(n,x)
VAR	Computes statistical variances		x or [x ... x_n]
VARS	Returns a list of variables in the current directory (28S)		{ names }

Name	Description	In ☞	Out
WAIT	Pauses program execution	*secs*	
WHILE	Begins WHILE ... REPEAT ... END	**WHILE** *test-clause* **REPEAT** *loop-clause* **END**	
XOR	Logical or binary XOR	$\#n_1$ $\#n_2$	$\#n_3$
XPON	Returns the exponent of a number	x	n
*H	Adjusts the height of a plot	factor	
*W	Adjusts the width of a plot	factor	
$\sqrt{\ }$	Square root	z	sqrt z
$\int$	Numeric integral	'symb' { global a b } *accuracy*	*integral error*
$\int$	Symbolic integral	'symb' 'global' degree	*'integral'*
∂	Derivative	'symb$_1$' 'global'	'symb$_2$'
π	Symbolic constant π		π
$\Sigma+$	Appends data point to the stat matrix	x	
$\Sigma-$	Deletes data point from the stat matrix		x
→	Assigns local variable(s)	n	
→ARRY	Combines numbers into an array	x_1 ... x_n n	[vector]
		x_1 ... x_n y_1 ... y_n { dim }	[matrix]
→HMS	Converts a number to HMS format	x	hms
→LCD	Displays graphics string (28S)	"string"	

Name	Description	In ☞	Out
→LIST	Combines objects into a list	obj_1 ... obj_n n	{obj_1 ... obj_n}
→NUM	Evaluates an object to yield a numeric result	obj	z
→STR	Converts an object to a string	obj	"string"
<	Less-than comparison	x y	x<y (flag)
>	Greater-than comparison	x y	x>y (flag)
≤	Less-than-or-equal comparison	x y	x≤y (flag)
≥	Greater-than-or-equal comparison	x y	x≥y (flag)
≠	Not-equal comparison	x y	x≠y (flag)
+	Adds two objects	z_1 z_2	$z_1 + z_2$
−	Subtracts two objects	z_1 z_2	$z_1 - z_2$
*	Multiplies two objects	z_1 z_2	$z_1 * z_2$
/	Divides two objects	z_1 z_2	z_1/z_2
^	Raises a number to a power	z_1 z_2	$z_1^{z_2}$
%	Percent	x y	xy/100
%CH	Percent change	x y	100(y−x)/x
%T	Percent total	x y	100y/x

THE HP-28 POCKET BOOK

Plot and Solve Example	2	Object Types	10
Differentiation Example	3	Operator Precedence	10
Integration Examples	4	Printer Control	11
Unit Examples	5	Column Graphics	11
Unit Prefixes	6	PPAR and ΣPAR Parameters	12
Reserved Names	6	ΣDAT Statistics Matrix	13
Truth Table	7	Character Codes	14
Key Strings	7	Errors by Number	16
System Operations	8	Errors by Message	18
Interactive Plot Mode	8	Flags	20
SYSEVAL Entries	9	Memory Usage	22
HP-28S Menus	9	Command Reference	23

To order additional copies, send $6.95 each plus $1.00 shipping and handling in U.S. funds (checks payable through a U.S. bank) to:

James Donnelly, 3135 N.W. Ashwood Drive, Corvallis, OR 97330 USA